1-2009

Maximizing Throughput of Bucket Brigades on Discrete Work Stations

Yun Fong LIM
Singapore Management University, yflim@smu.edu.sg

Kum Khiong YANG
Singapore Management University, kkyang@smu.edu.sg

Maximizing Throughput of Bucket Brigades on Discrete Work Stations

Yun Fong Lim • Kum Khiong Yang

Lee Kong Chian School of Business, Singapore Management University,
50 Stamford Road, Singapore 178899, Singapore
Phone: (65) 6828-0774 Fax: (65) 6828-0777
yflim@smu.edu.sg • kkyang@smu.edu.sg

Maximizing Throughput of Bucket Brigades on Discrete Work Stations

Abstract

One way to coordinate workers along an assembly line that has fewer workers than work stations is to form a bucket brigade. The throughput of a bucket brigade on discrete work stations may be compromised due to blocking even if workers are sequenced from slowest to fastest. For a given work distribution on the stations we find policies that maximize the throughput of the line. When workers have very different production rates, fully cross-training the workers and sequencing them from slowest to fastest is almost always the best policy. This policy outperforms other policies for most work distributions except for some cases in which limiting the work zones of workers produces higher throughput. In environments where the work can be adjusted across stations, we identify conditions for a line to prevent blocking.

Key words: bucket brigades; assembly lines; work stations; cross-training; work sharing

Submissions and Acceptance: Received September 2006; revisions received July 2007, April 2008; accepted April 2008 by Panos Kouvelis.

1 Introduction

To increase the production efficiency of an assembly line that has fewer workers than work stations, one needs to effectively coordinate the workers to minimize their idle time. One way is to coordinate the workers as a bucket brigade. When a bucket brigade is formed each worker simultaneously assembles a single item (an instance of the product) along the line. The worker carries the item from work station to work station until either he hands off his item to a downstream co-worker or he completes the work for his item. The worker then walks back to get another item, either from his co-worker upstream or from a buffer at the beginning of the line. The most notable application of bucket brigades is to coordinate workers to pick products for customer orders in distribution centers, as reported in Bartholdi and Eisenstein (1996b) and Bartholdi et al. (2001). Bucket brigades have also been used in the production of garments, the packaging of cellular phones, and the assembly of tractors, large-screen televisions, and automotive electrical harnesses (see Bartholdi and Eisenstein (1996a,b), Bartholdi and Eisenstein (2005), and Villalobos et al. (1999a,b)).

Bartholdi and Eisenstein (1996a) are the first who studied bucket brigade assembly lines analytically. They assumed that work content of the product is deterministic. Workers are not allowed to pass one another so that their sequence along the line is preserved. Each worker proceeds forward with a deterministic, finite *work* velocity reflecting the worker's familiarity with the work content. Furthermore, the time for each worker to walk back to get more work is assumed to be negligible (that is, workers walk back with an infinite velocity). They proved that when workers are sequenced from slowest to fastest (according to their work velocities) in the direction of production flow, workers will eventually hand off items to their co-worker downstream at fixed locations. Every worker will eventually repeat their respective portion of work content on each item produced. The system is said to converge to a *fixed point* (see, for example, Alligood et al. (1996)).

Bartholdi and Eisenstein (1996a) also examined the throughput (number of items produced per unit time) of bucket brigade assembly lines. For a line with discrete work stations they observed that the throughput of the slowest-to-fastest ordering of workers is always within a factor of n of the best achievable, where n is the number of workers on the line. In contrast, the throughput of other orderings can be arbitrarily bad. They also studied a special case in which the work content is continuously and evenly distributed along the line. They proved that the slowest-to-fastest ordering attains the maximum possible throughput (sum of work velocities of all workers) in this special case.

Convergence to the fixed point is desirable because it creates several positive effects, such as the skills of workers are reinforced by repetition, the effort of each worker contributes directly to the output, and the output is regular, which simplifies the coordination of downstream processes. All these effects are created without the intervention of management or engineering.

Some extended work on bucket brigades has been done based on the assumption that the work content of the product is distributed continuously and evenly along the assembly line. Bartholdi et al. (1999) have described all possible dynamics of two- and three-worker bucket brigades. They analyzed the system with workers not necessarily sequenced from slowest to fastest. Armbruster and Gel (2006) studied a two-worker bucket brigade in which the work velocities of workers do not dominate each other along the entire line. Bartholdi and Eisenstein (2005) and Bratcu and Dolgui (2005) assumed workers spend significant time to walk back for more work. Bartholdi et al. (2006) extended the ideas of bucket brigades to a network of subassembly lines so that all subassembly lines are synchronized to produce at the same rate and items are completed at regular, predictable intervals.

However, for many assembly lines the work content is neither continuously nor evenly distributed, but is clumped in various proportions into discrete work stations. Workers in such lines may block each other as each station can accommodate only one worker at a time (for example, due to limited equipment). The chance of blocking increases as the number of workers approaches the number of stations. In such situations a model based on discrete work stations becomes more accurate. Furthermore, all the extended work on bucket brigades mentioned above assumes that workers are *fully cross-trained* — they are trained to work at any part of the assembly line. In many environments in which the product's life cycle is short, such as the fashion industry reported by Bischak (1996), it can be costly to train workers to use all machines. Instead, in such environments workers are usually *partially cross-trained* — each worker can only work on a subset of stations on the line.

Our goal in this paper is to identify conditions that maximize the throughput of bucket brigades on discrete work stations. Although Bartholdi and Eisenstein (1996a) have examined the throughput of a fully cross-trained team on discrete work stations in terms of worst-case bounds, some questions remain unanswered:

1. Is it always necessary to fully cross-train the workers in order to maximize the throughput? Does limiting the work zone of workers to a subset of stations always make the line less productive? How should the workers be sequenced (for example, slowest-to-fastest or fastest-to-slowest)? In short, let a *policy* be a combination of the span of cross-training and the ordering of workers along the line. For a given work-content distribution on the stations what is the policy that produces the highest throughput?

2. For a given policy how should we adjust the work content across stations, if the tasks on the stations can be reassigned, so that the throughput is maximized?

We first answer the above questions for a small system by studying its dynamical behavior. We then extend some of our observations to larger systems. Specifically, we first determine the asymptotic behavior and average throughput of a line with three stations and two workers, who are only partially cross-trained. We examine both slowest-to-fastest and fastest-to-slowest orderings of workers. For each ordering, we consider a general distribution of work content on the stations. We then repeat the same analysis for a line with fully cross-trained workers. By studying both partially and fully cross-trained teams, with respect to the two orderings of workers, we find a policy that produces the highest throughput for a given work-content distribution on the stations.

Our main result is summarized as follows. When workers have very different work velocities, fully cross-training and sequencing them from slowest to fastest *almost always* produces the highest throughput. This policy outperforms other policies for most work-content distributions except for some cases in which the system is more productive if the work zone of workers is limited to a subset of stations. This observation is especially useful when there is variability in work content such that the work-content distribution changes from one batch of items to another. In such situations, fully cross-training and sequencing the workers from slowest to fastest is almost always the best policy.

For assembly lines with work content that is adjustable across stations we find work-content distributions that maximize the throughput for a given policy. We also identify conditions for larger bucket brigades to prevent blocking so that the line can fully use its production capacity.

2 Bucket brigades on discrete work stations

Consider an assembly line in which each instance of a product is progressively assembled on the same sequence of m work stations. We assume that the work content of the product on work station j is deterministic and is denoted as s_j. We normalize the total work content to 1 so that $\sum_{j=1}^{m} s_j = 1$. The assembly line can be conceptually represented by a real line with length 1 such as shown in Figure 1. The work on each item is cumulative along the line so that point 0 represents the beginning of the work

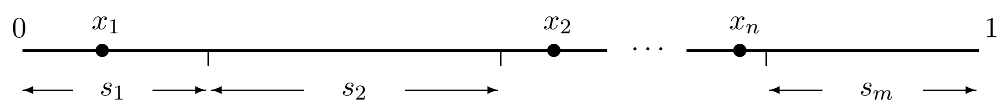


Figure 1: The total work content of the product is conceptually represented by a line segment, which is partitioned into intervals corresponding to the work stations. The location of worker i is denoted as x_i, which represents the cumulative fraction of work content completed on his item.

on an item and point 1 represents the completion of the item.

Workers are indexed from 1 to n and they remain in this sequence along the assembly line in the direction of production flow. Workers $i-1$ and $i+1$ are the *predecessor* and the *successor*, respectively, of worker i. We consider assembly lines in which each worker i is cross-trained to work on *zone* Z_i — a set of contiguous stations on the line. A worker i is fully cross-trained if Z_i contains all stations on the line. Otherwise, he is partially cross-trained. We assume that $\bigcup_{i=1}^{n} Z_i = \{1, 2, \ldots, m\}$.

Each worker i carries a single item and continues to assemble his item from station to station until he hands off his item to his successor or, if worker i is the last worker on the line, he completes the work on his item at the end of the line. Note that a worker $i < n$ will be *blocked* if he finishes his work on station j while his successor is still working on the next station $j + 1 \in Z_i$. The blocked worker remains idle until the next station becomes available. A worker $i < n$ will be *halted* if he finishes his work on all stations in Z_i before he can hand off his item to his successor. The halted worker remains idle until his successor takes over his item.

After relinquishing an item worker i walks back upstream to take over work from his predecessor or, if worker i is the first worker on the line, he initiates a new item. Note that a worker $i > 1$ will be *starved* if he walks back and reaches the beginning of his zone before his predecessor can hand off an item to him. The starved worker remains idle until he takes over an item. When an item is handed off we assume that the work content is *preemptible* without any loss of work.

When the last worker (worker n) finishes the work on his item the line *resets* itself: worker n walks back to get work from worker $n-1$, who in turn walks back to get work from worker $n-2$, and so on until worker 1 initiates a new item. Unlike the model studied by Bartholdi and Eisenstein (1996a), a reset in our model may not be instantaneous because a worker $i > 1$, who walks back to get more work, could be starved at the start of zone Z_i. If this is the case, the predecessor of worker i will not start walking back until he reaches the start of zone Z_i and passes his item to worker i. Table 1 summarizes the rules followed by the workers.

Table 1: Each worker independently follows these extended Bucket Brigade Rules.

Forward Rule – Work forward with your item until one of the following events occurs:

1. Your item is taken by your successor;
2. you are halted, in this case wait till you pass your item to your successor;
3. you complete your item at the end of the line;

then follow the Backward Rule.

Backward Rule – Walk back until one of the following events occurs:

1. You encounter your predecessor, in this case take over his item;
2. you are starved, in this case wait till you receive an item from your predecessor;
3. you reach the start of the line, in this case begin a new item;

then follow the Forward Rule.

Each worker i works forward with a constant work velocity v_i within zone Z_i except when they are blocked. This velocity represents the worker's dexterity and motivation. We consider assembly lines in which the time to assemble an item is significantly longer than the time to walk the entire line. Therefore, we assume that each worker spends negligible time to walk back to his predecessor or to the beginning of his zone.

Note that workers will not be halted or starved if they are fully cross-trained. By analyzing the dynamics of a line with two workers on three stations we study how blocking, halting, and starvation arise, and how these different forms of worker idleness deteriorate the throughput of the line.

3 A three-station, two-worker line

Consider an assembly line with $m = 3$ stations and $n = 2$ workers. We analyze two cases:

1. Each worker is partially cross-trained with $Z_1 = \{1, 2\}$ and $Z_2 = \{2, 3\}$.

2. All workers are fully cross-trained. This corresponds to the system considered by Bartholdi and Eisenstein (1996a).

We will present the results for partially and fully cross-trained teams in Subsections 3.1 and 3.2 respectively, but first we introduce the methodology that we use for both cases.

Let x^k be the location of worker 1 along the line (such as shown in Figure 1) immediately before the kth reset. x^k represents the cumulative fraction of work content completed on worker 1's item immediately before worker 2 finishes the work on item k at the end of the line. Since the last station (station 3) can only be occupied by worker 2, $x^k \in [0, s_1 + s_2]$.

If $x^k \in [s_1, s_1 + s_2]$ then the next hand-off will occur at point x^k. However, if $x^k \in [0, s_1)$ then the next hand-off location depends on whether worker 2 can work on station 1. If workers are partially cross-trained worker 2, immediately after relinquishing item k, will be starved at point s_1 until worker 1 hands off item $k+1$ to him. In contrast, if workers are fully cross-trained the next hand-off will occur at point x^k.

A *map* is a function f that determines the next iterate x^{k+1} based on the current iterate x^k, that is $x^{k+1} = f(x^k)$. The sequence of iterates $x^1, x^2, x^3, \ldots$ is called the *orbit* of an initial iterate x^0 under f. By constructing the map of the line we can obtain the orbit of any initial iterate x^0. The asymptotic behavior of the line can be predicted

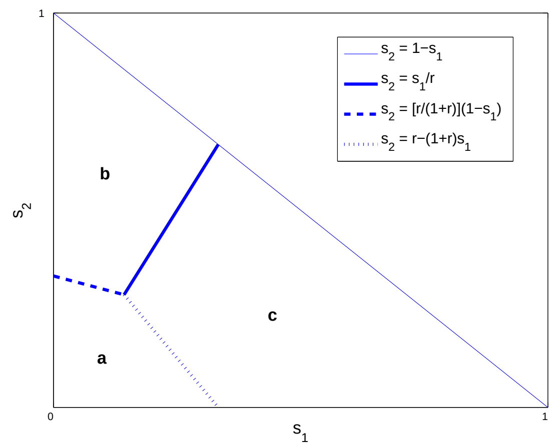


Figure 2: The feasible work-content region ($s_2 < 1 - s_1$) is partitioned into three mutually exclusive regions. The dynamics in each region are described by a distinct map.

by analyzing the properties of its map. Furthermore, define $r = v_1/v_2$. The feasible work-content region ($s_2 < 1 - s_1$) of the line can be partitioned into three mutually exclusive regions as depicted in Figure 2. The dynamics in each region are described by a distinct map given in Lemma 1.

Lemma 1. *The map $x^{k+1} = f(x^k)$ for the line with $m = 3$ stations and $n = 2$ workers, who are all partially or all fully cross-trained, is given as follows:*

Region a, *where $s_2 < r/(1+r) - [r/(1+r)]s_1$ and $s_2 < r - (1+r)s_1$,*

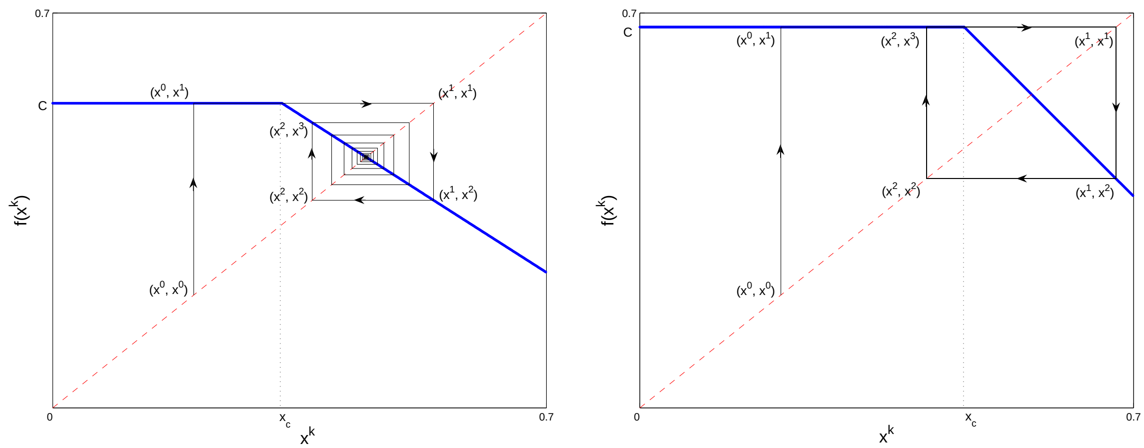
$$f(x^k) = \begin{cases} s_1 + s_2, & \text{if } x^k < 1 - (s_1 + s_2)/r; \\ r - rx^k, & \text{otherwise.} \end{cases}$$

Region b, *where $s_2 \geq r/(1+r) - [r/(1+r)]s_1$, $s_2 > s_1/r$, and $s_2 < 1 - s_1$,*

$$f(x^k) = \begin{cases} s_1 + r(1 - s_1 - s_2), & \text{if } x^k < (1 - 1/r)s_1 + s_2; \\ r - rx^k, & \text{otherwise.} \end{cases}$$

Region c, *where $s_2 \geq r - (1+r)s_1$, $s_2 \leq s_1/r$, and $s_2 < 1 - s_1$,*

$$f(x^k) = \begin{cases} r(1 - s_1), & \text{if } x^k \leq s_1; \\ r - rx^k, & \text{otherwise.} \end{cases}$$



(a) (b)

Figure 3: (a) The map of an assembly line with three stations and two workers is plotted with $s_1 = 0.3, s_2 = 0.4$, and $r = 0.80$. The system converges to a fixed point. (b) The map of the same assembly line is plotted with the positions of workers swapped. Thus, $r = 1/0.8 = 1.25$. The system converges to a period-2 orbit.

Proof. See Appendix A.1. □

For each of the three cases of Lemma 1, the function f has one of two forms: Either it is equal to a constant (which we denote as C) whenever x^k is less than a threshold value (which we denote as x_c), or it is linear with slope $-r$ whenever x^k exceeds the threshold value. Each case has a different constant C and a different threshold value x_c. For example, $C = s_1 + s_2$ and $x_c = 1 - (s_1 + s_2)/r$ in Region a, whereas $C = r(1 - s_1)$ and $x_c = s_1$ in Region c.

Figure 3 gives two examples of the map of an assembly line with $s_1 = 0.3$ and $s_2 = 0.4$. Graph (a) shows the map of the line with $r = 0.80 < 1$. In this case the system falls in Region b of Figure 2. The thick solid line represents the function f. The dashed line is the diagonal. The thin solid line with arrows traces the orbit of the initial iterate $x^0 = 0.2$ under f. It links the points $(x^k, x^k), (x^k, x^{k+1})$, and (x^{k+1}, x^{k+1}), for $k = 0, 1, 2, \ldots$. (See, for example, Alligood et al. (1996) for an interpretation of such a graph.)

The function f intersects with the diagonal at point $x^* = r/(1+r)$, which is a fixed point because $f(x^*) = x^*$. The function f has an upper bound of $C = s_1 + r(1-s_1-s_2)$. Since $f(C) > x_c = (1-1/r)s_1 + s_2$ the orbit of all initial iterates x^0 will be bounded between x_c and C. Any hand-off location between x_c and C can be expressed as $x^k = x^* + \eta^k$, where η^k represents the deviation of x^k from x^*. Is the orbit of x^k attracted to or repelled from x^* under f? Consider the next iterate

$$\begin{aligned} x^{k+1} &= f(x^k); \\ x^* + \eta^{k+1} &= r - r(x^* + \eta^k); \\ \eta^{k+1} &= -r\eta^k. \end{aligned} \tag{1}$$

Since $\eta^1 = -r\eta^0$ and $\eta^2 = -r\eta^1 = (-r)^2\eta^0$, in general $\eta^k = (-r)^k\eta^0$. If $r < 1$ (such as the map shown in Figure 3(a)) $\eta^k \to 0$ as $k \to \infty$. This implies that the orbit of any point x^k between x_c and C will be attracted to the point x^*. Thus, when $r < 1$ the orbit of all initial iterates $x^0 \in [0, s_1 + s_2]$ will converge to the fixed point x^*. In Figure 3(a) the thin solid line shows how the orbit of the initial iterate $x^0 = 0.2$ converges to x^*. After the system converges to the fixed point worker 2 always takes over work from worker 1 at x^*. The average throughput of the system is $\rho = [(1-x^*)/v_2]^{-1} = v_1 + v_2$.

If we swap the positions of workers along the line in the previous example then $r = 1/0.80 = 1.25$. In this case the system also falls in Region b of Figure 2. The map is shown in Figure 3(b). According to Equation (1), if $r > 1$ the orbit of any point x^k between x_c and C will be repelled from x^*. The orbit will eventually converge to a *period-2 orbit*: $C, f(C), C, f(C), \ldots$, where $C = s_1 + r(1-s_1-s_2)$ and $f(C) = r - rC$. Thus, when $r > 1$ the orbit of all initial iterates $x^0 \in [0, s_1 + s_2]$ will converge to the same period-2 orbit. In Figure 3(b) the thin solid line with arrows shows how the orbit of the initial iterate $x^0 = 0.2$ converges to this period-2 orbit. Since both C and $f(C)$ are greater than s_1, after the system converges to the period-2 orbit, worker 2 takes over work from worker 1 at point C and point $f(C)$ alternatively. The average throughput of the system is $\rho = 2/[(1-C)/v_2 + (1-f(C))/v_2] = 2v_2/[2-r+(r-1)C]$.

Following the same method as described in the previous examples we analyze the dynamics and determine the average throughput of the system for both partially and fully cross-trained workers. In each case we consider both slowest-to-fastest and the reverse orderings with respect to a general distribution of work content on the stations.

3.1 Results for partially cross-trained workers

Sequencing workers from slowest to fastest ($r \leq 1$)

The *phase diagram* in Figure 4(a) summarizes the asymptotic behavior of the system with partially cross-trained workers when $r < 1$. The horizontal and vertical axes of the phase diagram correspond to s_1 and s_2 respectively. Each point on the phase diagram represents a distribution of work content on the work stations. The feasible region is $s_2 < 1 - s_1$, which can be partitioned into four mutually exclusive regions. The system converges to a distinct fixed point in each region. However, the line does not fully use its production capacity in all regions. We briefly summarize the asymptotic behavior in each region below. More details can be found in Appendix A.2.

Region 1 (Halting): Worker 1 is always halted at point $s_1 + s_2$ (in front of station 3) before the line resets, because the work content on the first two stations is so small that he can finish his work on these stations before worker 2 takes over work from him.

Region 2 (Full capacity): The fixed point is located on station 2. There will be no idleness on the fixed point at which hand-offs repeatedly occur. The average throughput is $\rho = v_1 + v_2$, the maximum possible. This is the only region in which workers are fully utilized.

Region 3 (Blocking): Worker 1 is always blocked at point s_1 (in front of station

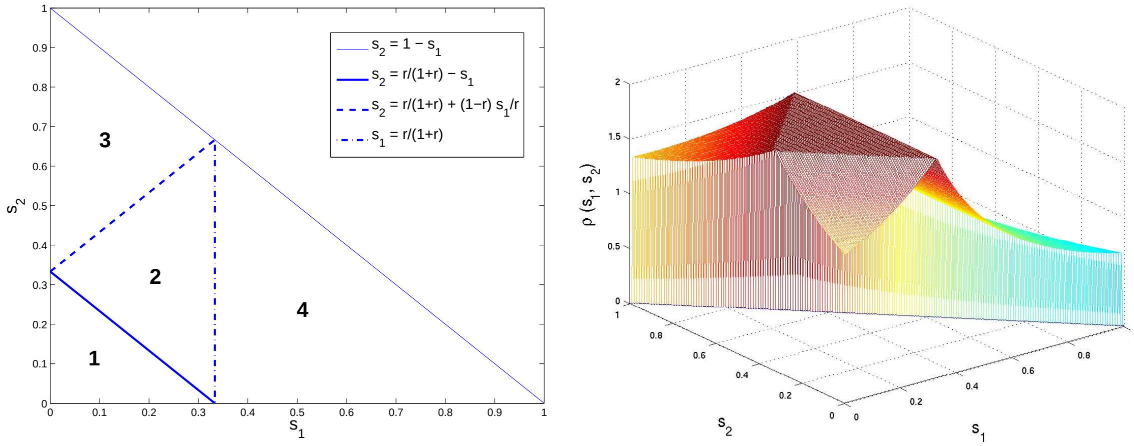


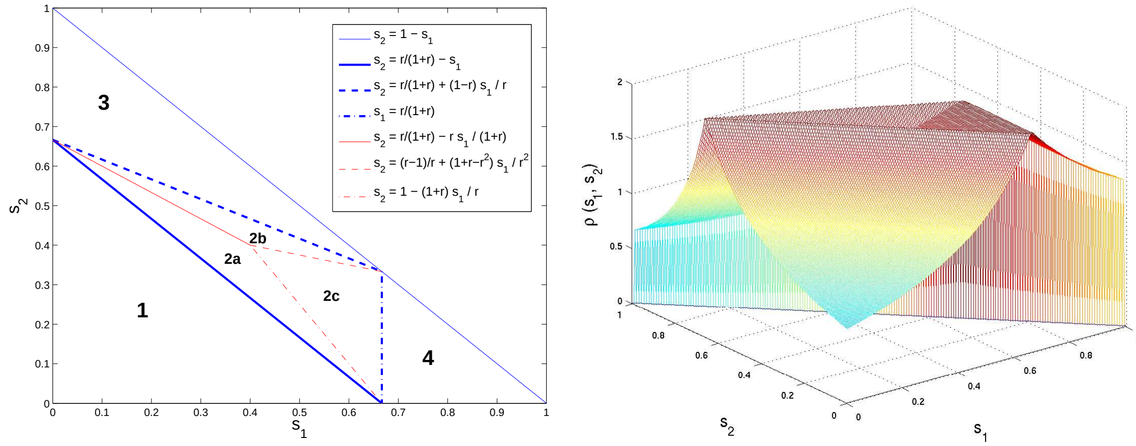
Figure 4: (a) The phase diagram of a line with three stations and two partially cross-trained workers, with $r = 1/2 < 1$, can be divided into four regions. Each region is characterized by a distinct asymptotic behavior. (b) The average throughput of the system is plotted as a function of s_1 and s_2 (in this case $v_1 = 2/3$ and $v_2 = 4/3$). The average throughput is the highest in Region 2.

2) before station 2 becomes available because the work content on station 1 is relatively small.

Region 4 (Starvation): The work content on station 1 is so large that the fixed point is located on station 1. Thus, worker 2 is always starved at point s_1 before worker 1 can hand off work to him.

The average throughput of the system is shown in Figure 4(b). The average throughput is the highest in Region 2. It decreases monotonically as we move into other regions due to halting (Region 1), blocking (Region 3), or starvation (Region 4).

When workers have the same work velocity ($r = 1$) the phase diagram of the system is similar to Figure 4. The asymptotic behavior and average throughput in each region remain unchanged except for Region 2, in which the system converges to a period-2 orbit with average throughput $\rho = 2v_1 = 2v_2$.



(a) (b)

Figure 5: (a) The phase diagram of the same assembly line is plotted with the positions of workers swapped ($r = 2 > 1$). The asymptotic behavior and average throughput in each region remain unchanged from that of Figure 4 except Region 2, which is now divided into three sub-regions. (b) The average throughput of the system is plotted as a function of s_1 and s_2 (in this case $v_1 = 4/3$ and $v_2 = 2/3$).

Sequencing workers from fastest to slowest ($r > 1$)

Figure 5 shows the phase diagram and average throughput of the system when workers are sequenced from fastest to slowest. The asymptotic behavior and the expression of average throughput in each region remain unchanged from that of the slowest-to-fastest ordering except Region 2, which is now divided into three sub-regions 2a, 2b, and 2c. In all these sub-regions the system converges to a period-2 orbit. Details can be found in Appendix A.2.

3.2 Results for fully cross-trained workers

When workers are fully cross-trained the line is identical to that considered by Bartholdi and Eisenstein (1996a). There is no halting or starvation. Therefore, blocking is the only source of idleness. Our contribution here is to find the asymptotic behavior and average throughput of the line for both slowest-to-fastest and the reverse orderings of workers.

The phase diagram of an assembly line with three stations and two fully cross-trained workers is similar to that of a partially cross-trained team (see Figures 4(a) and 5(a)). If $r < 1$ the asymptotic behavior (fixed point) in all regions remains unchanged (this is consistent with Theorem 2 of Bartholdi and Eisenstein (1996a)). In contrast, if $r > 1$ Region 2 is partitioned into three *new* sub-regions. In all these sub-regions the system converges to a period-2 orbit. Details on the asymptotic behavior and average throughput of the line can be found in Appendix A.3.

The phase diagrams capture all possible asymptotic behaviors of a three-station, two-worker line. They are useful for designing such a line when the work content is adjustable across stations. If the span of cross-training and the ordering of workers are fixed, the phase diagrams can be used to select an appropriate distribution of work content such that the throughput is maximized. For example, if the work velocities of two partially cross-trained workers are 2/3 and 4/3, then Figures 4 and 5 can be used to select the appropriate work-content distributions to maximize the throughput when workers are sequenced from slowest to fastest and from fastest to slowest respectively.

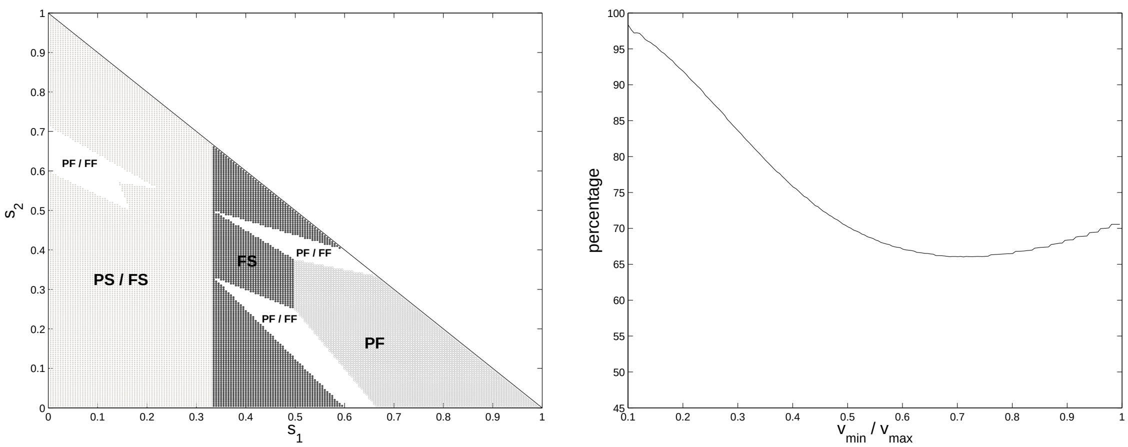
3.3 The best policies

Table 2 summarizes all policies discussed in this paper. Given the work-content distribution on the three work stations, which policy should we use so that the throughput of a two-worker bucket brigade is maximized?

Table 2: A policy is a combination of the span of cross-training and the ordering of workers along the assembly line.

	Partially cross-trained	**Fully cross-trained**
Slowest-to-fastest	PS	FS
Fastest-to-slowest	PF	FF

We compare these policies based on a two-worker line with work velocities 2/3 and



(a) (b)

Figure 6: (a) Different policies achieve the highest average throughput for different work-content distributions. The FS and PF policies dominate all other policies. [PS: partially cross-trained, slowest-to-fastest; PF: partially cross-trained, fastest-to-slowest; FS: fully cross-trained, slowest-to-fastest; FF: fully cross-trained, fastest-to-slowest.] (b) The FS policy is almost always the best policy when the work velocities are very different.

4/3. Figure 6(a) shows the policies that give the highest average throughput for different work-content distributions on the stations. The FS policy is the most productive among all policies in the heavily shaded area in the middle of the feasible region. The lightly shaded area on the left covers the distributions for which both PS and FS policies give the highest average throughput. The PF policy beats all other policies in the lightly shaded area on the right. Finally, the unshaded areas favor both PF and FF policies.

Figure 6(a) suggests that there is a substantial area in the feasible work-content region in which the PS policy performs equally well as the FS policy. If the work-content distribution on stations lies in this area we only need to partially cross-train the workers as the throughput of the system is equivalent to that of a fully cross-trained team. This is appealing because in practice it could be expensive or impossible to fully cross-train the workers.

The feasible region is dominated by the FS and PF policies — either FS or PF is the best policy for any work-content distribution. Note that the PF policy outperforms the FS policy when s_1 is relatively large compared to s_2. This is because under the PF policy the faster worker works on station 1 (with large work content) without being interrupted by the slower worker. Figure 6(a) is useful when the work content on the stations is fixed and nonadjustable (for example, for stations with discrete pieces of equipment). The graph presents the best policy for a given work-content distribution. For example, consider two workers with work velocities 2/3 and 4/3. Figure 6(a) shows that if stations 1 and 2 constitute 0.7 and 0.2, respectively, of the total work content, then limiting the work shared by the workers to station 2 and sequencing the workers from fastest to slowest produces the highest throughput.

Define $v_{min} = min\{v_1, v_2\}$ and $v_{max} = max\{v_1, v_2\}$. Figure 6(b) shows the percentage of the feasible region in which the FS policy outperforms the PF policy (and other policies) as a function of the velocity ratio v_{min}/v_{max}. Note that the FS policy is the best among all policies for about 80% of the feasible region when $v_{min}/v_{max} = 1/3$ (the minimum ratio observed in apparel manufacturing by Bartholdi and Eisenstein (1996a) and Bartholdi et al. (1999)). This percentage reaches about 98% when v_{min} is ten times smaller than v_{max}. This suggests that, if the work velocities are very different, fully cross-training the workers and sequencing them from slowest to fastest is almost always the best policy to use. This observation is useful when there is variability in the work content. For example, the distribution of work content on the stations may change from one batch of items to another. This causes (s_1, s_2) to shift from one point to another so that it distributes, for example, uniformly over the entire feasible work-content region. Our result suggests that the FS policy warrants the highest production rate most of the time in such situations.

4 Results for larger systems

The complexity of the analysis of a line with m stations and n workers increases rapidly with m and n, making the search for the policies that maximize the throughput of larger systems intractable. Bartholdi and Eisenstein (1996a) showed that there exists a fixed point for a line with n fully cross-trained workers on m discrete work stations, for $m > n \geq 2$ (Theorem 1 of Bartholdi and Eisenstein (1996a)). Furthermore, if workers are sequenced from slowest to fastest then the fixed point is unique (Lemma 1 of Bartholdi and Eisenstein (1996a)) and the line will converge to the fixed point (Theorem 2 of Bartholdi and Eisenstein (1996a)). However, convergence to the fixed point does not guarantee that the line will attain the maximum possible throughput ($\sum_{i=1}^{n} v_i$) because the fixed point may have blocking for a line with discrete work stations.

For a three-station, two-worker line, if workers are sequenced from slowest to fastest we can make the line converge to a fixed point with no blocking by adjusting the work-content distribution such that the system falls in Region 2 of its phase diagram (see Figure 4). Fortunately, the conditions for Region 2 (conditions to achieve full production capacity) can be generalized to a line with m stations and n workers, for $m > n \geq 2$.

The locations of workers along the line immediately before the kth reset can be expressed as $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_n^k)$, where x_i^k represents the fraction of work content completed on the item carried by worker i immediately before the kth reset. Note that $x_n^k = 1$ for all k. Let $\mathbf{F}$ be the function, defined implicitly by the bucket brigade protocol, that maps $\mathbf{x}^k$ to $\mathbf{x}^{k+1}$ so that $\mathbf{x}^{k+1} = \mathbf{F}(\mathbf{x}^k)$. We say $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is a fixed point if $\mathbf{x}^* = \mathbf{F}(\mathbf{x}^*)$. For convenience we define $x_0^* = 0$. Furthermore, define $j^*(i)$ as the smallest index such that $\sum_{j=1}^{j^*(i)} s_j \geq x_i^*$, for $i = 1, \ldots, n$. $j^*(i)$ is the index of the station in which the point x_i^* falls within. For example, $x_n^* = 1$ and so $j^*(n) = m$.

Lemma 2. *For a line with m stations and n fully cross-trained workers, for $m > n \geq 2$,* *no blocking will occur at the fixed point* $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$*, where*

$$x_i^* = \frac{\sum_{j=1}^{i} v_j}{\sum_{j=1}^{n} v_j}, \quad i = 1, \ldots, n;$$

if the following conditions are satisfied:

$$1 < j^*(1) < j^*(2) < \ldots < j^*(n-1) < m; \tag{2}$$

$$\frac{\sum_{j=1}^{j^*(i)} s_j - x_i^*}{v_{i+1}} < \frac{\sum_{j=1}^{j^*(i)-1} s_j - x_{i-1}^*}{v_i}, \quad i = 1, \ldots, n-1. \tag{3}$$

Proof. Suppose $\mathbf{x}^k = \mathbf{x}^*$ we will show that there will be no blocking after the kth reset and the hand-off locations will remain at $\mathbf{x}^*$. Condition (2) ensures the following:

1. No two neighboring hand-off locations x_i^* and x_{i+1}^* can be on the same station. Otherwise, more than one worker will occupy a single station at a time.
2. No hand-offs occur on station 1. Otherwise, worker 1 will be blocked at point 0.
3. No hand-offs occur on station m. This is necessary because only the last worker (worker n) can work on station m.

Condition (3) ensures that no blocking will occur while worker i enters station $j^*(i)$. Since workers will not be blocked and

$$\frac{x_i^* - x_{i-1}^*}{v_i} = \frac{x_{i+1}^* - x_i^*}{v_{i+1}},$$

for $i = 1, \ldots, n-1$, $\mathbf{x}^{k+1} = \mathbf{x}^*$. □

Lemma 2 lists the conditions to prevent blocking. It provides guidelines for designing a line where work content on the stations can be adjusted. If workers are sequenced from slowest to fastest then the line will converge to the fixed point.

Lemma 3. *For a line with m stations and n fully cross-trained workers, for $m > n \geq 2$, if workers are sequenced from slowest to fastest and the conditions in Lemma 2 hold, then the line will converge to the fixed point in Lemma 2 and the throughput at the fixed point is $\sum_{i=1}^{n} v_i$, the maximum possible.*

Proof. Theorem 2 of Bartholdi and Eisenstein (1996a) ensures the convergence to the unique fixed point. Lemma 2 ensures no blocking occurs at the fixed point. The throughput at the fixed point is $[(1 - x_{n-1}^*)/v_n]^{-1} = \sum_{i=1}^{n} v_i$. □

To make a bucket brigade converge to a fixed point that produces the maximum possible throughput on discrete work stations, one can use the following procedure: First, fully cross-train the workers. Second, sequence them from slowest to fastest. Last, adjust the work content on the stations so that the conditions in Lemma 2 are satisfied. Of course, to assess if the above procedure is justifiable in practice, one needs to compare the total cost of training the workers and adjusting the work content with the benefits of maximizing throughput.

5 Managerial implications and conclusions

Ideally, we want a bucket brigade on discrete work stations to converge to a fixed point so that each worker concentrates on his own interval of work content along the line and the line fully uses its production capacity. To achieve this goal for a system with arbitrary size, we require the workers to be flexible to work on any part of the assembly line and the work content to be adjustable across stations. In practice, however, it can be expensive to train workers to work on all stations and the tasks on the stations may not be adjustable. In these more restrictive environments the throughput of a bucket brigade assembly line may be compromised because of worker idleness due to blocking, halting, or starvation. In this paper, we provide guidelines on how to maximize the throughput of a bucket brigade on discrete work stations in such environments.

Our analysis on three-station, two-worker lines shows that when workers have very different work velocities, fully cross-training the workers and sequencing them from slowest to fastest almost always produces the highest throughput. For example, when the work velocities are three times different (which is observed in apparel manufacturing), 80% of feasible work-content distributions favor the FS policy. When the work velocities are ten times different, the FS policy dominates in 98% of feasible work-content distributions. This observation is especially useful when there is variability in work content across stations. For example, the work-content distribution may vary from a batch of items to another such that it distributes uniformly over the entire feasible work-content region. Our result suggests that the FS policy produces the highest throughput most of the time in such situations.

Only for a small minority of work-content distributions the FS policy does not dominate. This happens primarily when the work content on station 1 is relatively large compared to that of station 2. For these work-content distributions the best policy is to limit the share work to station 2 and sequence the workers from fastest to slowest. This is because under this policy the large work content on station 1 is covered by the faster worker and will not be preempted by the slower worker. Note that although this policy produces the highest throughput for these work-content distributions, the system does not converge to a fixed point. Instead, hand-offs occur at two alternate points on the line. This may dilute the learning effect of the workers because they do not experience a fixed assignment of work. Fortunately, as the difference in work velocities gets larger there will be less work-content distributions that favor this policy.

The analysis seems intractable for larger systems essentially because there are too many parameters involved. However, for the case in which workers are fully cross-trained and work content can be adjusted across stations, we have identified conditions to prevent blocking and thus attain the maximum possible throughput. The idea is to sequence the workers from slowest to fastest and to appropriately assign the work content to stations so that workers are never blocked when they enter a station.

Most of our results depend on the assumption that each worker has a constant work velocity over his work zone. Of course, this assumption may not hold in some environments. Relaxing this assumption becomes an interesting topic for future research.

Acknowledgments

The authors thank two anonymous referees and the senior editor for their constructive comments and suggestions, which led to a substantially improved version of this paper. The authors are supported by the Lee Kong Chian School of Business at the Singapore Management University.

References

Alligood, K.T., T.D. Sauer, J.A. Yorke. 1996. *Chaos: An Introduction to Dynamical Systems*. Springer.

Armbruster, D., E.S. Gel. 2006. Bucket brigades revisited: Are they always effective? *European Journal of Operational Research* **172**(1) 213–229.

Bartholdi, J.J. III, L.A. Bunimovich, D.D. Eisenstein. 1999. Dynamics of two- and three-worker "bucket brigade" production lines. *Operations Research* **47**(3) 488–491.

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

Bartholdi, J.J. III, D.D. Eisenstein. 1996b. The bucket brigade web page. URL `http://www.BucketBrigades.com`.

Bartholdi, J.J. III, D.D. Eisenstein. 2005. Using bucket brigades to migrate from craft manufacturing to assembly lines. *Manufacturing & Service Operations Management* **7**(2) 121–129.

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

Bartholdi, J.J. III, D.D. Eisenstein, Y.F. Lim. 2006. Bucket brigades on in-tree assembly networks. *European Journal of Operational Research* **168**(3) 870–879.

Bratcu, A.I., A. Dolgui. 2005. A survey of the self-balancing production lines ("bucket brigades"). *Journal of Intelligent Manufacturing* **16**(2) 139–158.

Bischak, D.P. 1996. Performance of a manufacturing module with moving workers. *IIE Transactions* **28**(9) 723–733.

Villalobos, J.R., L.F. Muñoz, L. Mar. 1999a. Assembly line designs that reduce the impact of personnel turnover. *Proceedings of IIE Solutions Conference*, Phoenix, AZ.

Villalobos, J.R., F. Estrada, L.F. Muñoz, L. Mar. 1999b. Bucket brigade: A new way to boost production. *Twin Plant News* **14**(12) 57–61.

A Technical details

A.1 Proof of Lemma 1

Proof. There are two mutually exclusive cases: (I) $s_2 > s_1/r$, and (II) $s_2 \leq s_1/r$.

(I) $s_2 > s_1/r$

When worker 2 completes item k worker 1 is either on station 1 ($x^k \leq s_1$) or station 2 ($s_1 < x^k \leq s_1 + s_2$). In either case worker 2 will walk back to take over work from worker 1, who in turn walks back and initiates a new item at the beginning of the line. There are two cases that lead to different behaviors: (A) $s_2 < r/(1+r) - [r/(1+r)]s_1$, and (B) $s_2 \geq r/(1+r) - [r/(1+r)]s_1$.

In case (A), $s_2 < r/(1+r) - [r/(1+r)]s_1 \Rightarrow s_2/v_1 < [1 - (s_1 + s_2)]/v_2$.

- If $x^k \leq s_1$ then worker 1 will be blocked at point s_1 after initiating a new item because $s_2 > s_1/r \Rightarrow s_2/v_2 > s_1/v_1$. Since $s_2/v_1 < [1 - (s_1 + s_2)]/v_2$, worker 1 will wait at point $s_1 + s_2$ before worker 2 walks back to take over work from him. Thus, $x^{k+1} = s_1 + s_2$.

- If $s_1 < x^k \leq s_1 + s_2$ then there are two different cases:

  – If $s_1/v_1 < (s_1 + s_2 - x^k)/v_2 \Rightarrow x^k < (1 - 1/r)s_1 + s_2$ then worker 1 will be blocked at point s_1 after initiating a new item. Since $s_2/v_1 < [1-(s_1+s_2)]/v_2$, worker 1 will wait at point $s_1 + s_2$ before worker 2 walks back to take over work from him. Thus, $x^{k+1} = s_1 + s_2$.

  – If $s_1/v_1 \geq (s_1 + s_2 - x^k)/v_2 \Rightarrow x^k \geq (1 - 1/r)s_1 + s_2$ then worker 1 will not be blocked when he reaches point s_1 after initiating a new item. If $(s_1 + s_2)/v_1 < (1 - x^k)/v_2 \Rightarrow x^k < 1 - (s_1 + s_2)/r$ then worker 1 will wait at point $s_1 + s_2$ before worker 2 walks back to take over work from him. Thus, $x^{k+1} = s_1 + s_2$. On the other hand, if $(s_1 + s_2)/v_1 \geq (1 - x^k)/v_2 \Rightarrow x^k \geq 1 - (s_1 + s_2)/r$ then worker 2 will reach the end of the line before worker 1 reaches point $s_1 + s_2$. Thus, $x^{k+1} = v_1[(1 - x^k)/v_2] = r - rx^k$.

Thus, in case (A), $x^{k+1} = s_1 + s_2$ if $x^k < 1 - (s_1 + s_2)/r$; and $x^{k+1} = r - rx^k$ if $x^k \geq 1 - (s_1 + s_2)/r$.

In case (B), $s_2 \geq r/(1 + r) - [r/(1 + r)]s_1 \Rightarrow s_2/v_1 \geq [1 - (s_1 + s_2)]/v_2$.

- If $x^k \leq s_1$ then worker 1 will be blocked at point s_1 after initiating a new item because $s_2/v_2 > s_1/v_1$. Since $s_2/v_1 \geq [1-(s_1+s_2)]/v_2$, worker 1 is still on station 2 when worker 2 reaches the end of the line. Thus, $x^{k+1} = s_1+v_1[(1-(s_1+s_2))/v_2] = s_1 + r[1 - (s_1 + s_2)]$.

- If $s_1 < x^k \leq s_1 + s_2$ then there are two different cases:

  - If $s_1/v_1 < (s_1 + s_2 - x^k)/v_2 \Rightarrow x^k < (1 - 1/r)s_1 + s_2$ then worker 1 will be blocked at point s_1 after initiating a new item. Since $s_2/v_1 \geq [1-(s_1+s_2)]/v_2$, worker 1 is still on station 2 when worker 2 reaches the end of the line. Thus, $x^{k+1} = s_1 + r[1 - (s_1 + s_2)]$.

  - If $s_1/v_1 \geq (s_1 + s_2 - x^k)/v_2 \Rightarrow x^k \geq (1 - 1/r)s_1 + s_2$ then worker 1 will not be blocked when he reaches point s_1 after initiating a new item. Since $x^k \geq (1 - 1/r)s_1 + s_2 \geq 1 - (s_1 + s_2)/r$, which implies that $(s_1 + s_2)/v_1 \geq (1 - x^k)/v_2$, worker 2 will reach the end of the line before worker 1 reaches point $s_1 + s_2$. Thus, $x^{k+1} = v_1[(1 - x^k)/v_2] = r - rx^k$.

Thus, in case (B), $x^{k+1} = s_1 + r[1-(s_1+s_2)]$ if $x^k < (1-1/r)s_1+s_2$; and $x^{k+1} = r-rx^k$ if $x^k \geq (1 - 1/r)s_1 + s_2$.

(II) $s_2 \leq s_1/r$

There are two cases that lead to different behaviors: (A) $s_2 < r - (1 + r)s_1$, and (B) $s_2 \geq r - (1 + r)s_1$.

In case (A) $s_2 < r - (1+r)s_1 \Rightarrow (s_1+s_2)/v_1 < (1-s_1)/v_2$.

- If $x^k \leq s_1$ then worker 1 will not be blocked when he reaches point s_1 after initiating a new item because $s_2 \leq s_1/r \Rightarrow s_2/v_2 \leq s_1/v_1$. Since $(s_1+s_2)/v_1 < (1-s_1)/v_2$, worker 1 will wait at point s_1+s_2 before worker 2 walks back to take over work from him. Thus, $x^{k+1} = s_1 + s_2$.

- If $s_1 < x^k \leq s_1 + s_2$ then $x^k > s_1 \geq s_1 - s_1/r + s_2 \Rightarrow s_1/v_1 > (s_1+s_2-x^k)/v_2$. Worker 1 will not be blocked when he reaches point s_1 after initiating a new item. If $(s_1+s_2)/v_1 < (1-x^k)/v_2 \Rightarrow x^k < 1-(s_1+s_2)/r$ then worker 1 will wait at point s_1+s_2 before worker 2 walks back to take over work from him. Thus, $x^{k+1} = s_1+s_2$. On the other hand, if $(s_1+s_2)/v_1 \geq (1-x^k)/v_2 \Rightarrow x^k \geq 1-(s_1+s_2)/r$ then worker 2 will reach the end of the line before worker 1 reaches point s_1+s_2. Thus, $x^{k+1} = v_1[(1-x^k)/v_2] = r - rx^k$.

Thus, in case (A), $x^{k+1} = s_1 + s_2$ if $x^k < 1-(s_1+s_2)/r$; and $x^{k+1} = r - rx^k$ if $x^k \geq 1-(s_1+s_2)/r$.

In case (B) $s_2 \geq r - (1+r)s_1 \Rightarrow (s_1+s_2)/v_1 \geq (1-s_1)/v_2$.

- If $x^k \leq s_1$ then worker 1 will not be blocked when he reaches point s_1 after initiating a new item because $s_1/v_1 \geq s_2/v_2$. Since $(s_1+s_2)/v_1 \geq (1-s_1)/v_2$, worker 2 will reach the end of the line before worker 1 reaches point s_1+s_2. Thus, $x^{k+1} = v_1[(1-s_1)/v_2] = r(1-s_1)$.

- If $s_1 < x^k \leq s_1 + s_2$ then $x^k > s_1 \geq s_1 - s_1/r + s_2 \Rightarrow s_1/v_1 > (s_1+s_2-x^k)/v_2$. Worker 1 will not be blocked when he reaches point s_1 after initiating a new item. Since $x^k > s_1 \geq 1-(s_1+s_2)/r \Rightarrow (s_1+s_2)/v_1 > (1-x^k)/v_2$, worker 2 will reach the end of the line before worker 1 reaches point s_1+s_2. Thus, $x^{k+1} = v_1[(1-x^k)/v_2] = r - rx^k$.

Thus, in case (B), $x^{k+1} = r(1-s_1)$ if $x^k \leq s_1$; and $x^{k+1} = r - rx^k$ if $x^k > s_1$. $\square$

A.2 Asymptotic behavior and throughput of a partially cross-trained team

Sequencing workers from slowest to fastest ($r < 1$)

The feasible region is $s_2 < 1 - s_1$, which can be partitioned into four mutually exclusive regions. Each of these regions is characterized by a distinct asymptotic behavior:

Region 1: This region is defined by $s_2 \leq r/(1+r) - s_1$. In this region both s_1 and s_2 are so small that worker 1 can complete the work on his item on both stations 1 and 2 before worker 2 takes over work from him. Thus, worker 1 is always halted at point $s_1 + s_2$ (in front of station 3) before the line resets. The system converges to the fixed point $x^* = s_1 + s_2$ with average throughput $\rho = [(1 - x^*)/v_2]^{-1} = v_2/(1 - s_1 - s_2)$.

Region 2: This region is defined by $s_1 < r/(1+r)$, $s_2 > r/(1+r) - s_1$, and $s_2 < r/(1+r) + [(1-r)/r]s_1$. The system converges to the fixed point $x^* = r/(1+r)$. The first two inequalities imply that the fixed point x^* can neither fall in the first nor the last stations. The last inequality $s_2 < r/(1+r) + [(1-r)/r]s_1 \Rightarrow (s_1 + s_2 - x^*)/v_2 < s_1/v_1$, which implies that after the system converges to the fixed point x^* worker 1 will not be blocked at point s_1. Thus, hand-offs will repeatedly occur at point x^*. The average throughput of the system is $\rho = [(1 - x^*)/v_2]^{-1} = v_1 + v_2$, the maximum possible. This is the only region in which workers are fully utilized.

Region 3: This region is defined by $s_2 \geq r/(1+r) + [(1-r)/r]s_1$ and $s_2 < 1 - s_1$. In this region s_1 is sufficiently small (relative to s_2) that worker 1 will get blocked at point s_1 (in front of station 2) before station 2 becomes available. The system converges to the fixed point $x^* = s_1 + r(1 - s_1 - s_2)$ with average throughput $\rho = [(1 - x^*)/v_2]^{-1} = v_2/[(1 - r)(1 - s_1) + rs_2]$.

Region 4: This region is defined by $s_1 \geq r/(1+r)$ and $s_2 < 1 - s_1$. In this region

s_1 is sufficiently large (relative to s_2) that worker 2 is always starved at point s_1. The system converges to the fixed point $x^* = r(1 - s_1)$, which is located on station 1 due to the inequality $s_1 \geq r/(1+r)$. The average throughput is $\rho = [(s_1 - x^*)/v_1 + (1 - s_1)/v_2]^{-1} = v_1/s_1$.

Sequencing workers from fastest to slowest ($r > 1$)

The asymptotic behavior and average throughput in Regions 1, 3, and 4 remain unchanged from that of the slowest-to-fastest ordering ($r < 1$). However, Region 2 is now divided into three sub-regions 2a, 2b, and 2c. In all these sub-regions the system converges to a period-2 orbit that consists of points q and $r(1-q)$. The value of q and the average throughput are different in different sub-regions:

Region 2a: This region is defined by $s_2 > r/(1+r) - s_1$, $s_2 < r/(1+r) - [r/(1+r)]s_1$, and $s_2 \leq 1 - [(1+r)/r]s_1$. In this region $q = s_1 + s_2$ and the average throughput converges to $\rho = 2v_2/[2 - r + (r-1)q]$.

Region 2b: This region is defined by $s_2 \geq r/(1+r) - [r/(1+r)]s_1$, $s_2 < r/(1+r) + [(1-r)/r]s_1$, and $s_2 \geq (r-1)/r + [(1+r-r^2)/r^2]s_1$. In this region $q = s_1 + r(1 - s_1 - s_2)$ and the average throughput converges to $\rho = 2v_2/[2 - r + (r-1)q]$.

Region 2c: This region is defined by $s_2 > 1 - [(1+r)/r]s_1$, $s_2 < (r-1)/r + [(1+r-r^2)/r^2]s_1$, and $s_1 < r/(1+r)$. In this region

$$q = \begin{cases} s_1 + s_2, & \text{if } s_2 < r/(1+r) - [r/(1+r)]s_1 \text{ and} \\ & \quad s_2 < r - (1+r)s_1; \\ s_1 + r(1 - s_1 - s_2), & \text{if } s_2 \geq r/(1+r) - [r/(1+r)]s_1 \text{ and } s_2 > s_1/r; \\ r(1 - s_1), & \text{otherwise.} \end{cases}$$

However, independent of q, the average throughput always converges to $\rho = 2v_1/[r + (1-r)s_1]$.

A.3 Asymptotic behavior and throughput of a fully cross-trained team

For Regions 1 and 3 of the phase diagram the asymptotic behavior and average throughput remain unchanged from that of a partially cross-trained team. For Region 4 the system converges to the fixed point $x^* = r(1 - s_1) \leq s_1$. Since worker 2 is able to work on station 1 now, the hand-off locations will converge to x^* with average throughput $\rho = [(1 - x^*)/v_2]^{-1} = v_2/[1 - r(1 - s_1)]$.

For Region 2 the asymptotic behavior and average throughput remain unchanged from that of a partially cross-trained team if $r \leq 1$. On the other hand, if $r > 1$ Region 2 is partitioned into three new sub-regions. In all these sub-regions the system converges to a period-2 orbit that consists of points q and $r(1 - q)$ with average throughput $\rho = 2v_2/[2 - r + (r - 1)q]$. The value of q is different in different sub-regions:

Region 2a: This region is defined by $s_2 > r/(1+r) - s_1$, $s_2 < r/(1+r) - [r/(1+r)]s_1$, and $s_2 < r - (1+r)s_1$. In this region $q = s_1 + s_2$.

Region 2b: This region is defined by $s_2 \geq r/(1+r) - [r/(1+r)]s_1$, $s_2 < r/(1+r) + [(1-r)/r]s_1$, and $s_2 > s_1/r$. In this region $q = s_1 + r(1 - s_1 - s_2)$.

Region 2c: This region is defined by $s_2 \geq r - (1+r)s_1$, $s_2 \leq s_1/r$, and $s_1 < r/(1+r)$. In this region $q = r(1 - s_1)$.