Performance of Cellular Bucket Brigades with Hand-Off Times

Yun Fong Lim
Lee Kong Chian School of Business, Singapore Management University,
50 Stamford Road, Singapore 178899, Singapore
yflim@smu.edu.sg

May 25, 2017

Abstract

A cellular bucket brigade is a way to coordinate workers along an aisle with work content on both sides. Each worker in a cellular bucket brigade works on one side of the aisle when he proceeds in one direction, and he works on the other side when he proceeds in the reverse direction. Although the cellular bucket brigade eliminates the unproductive walk-back, it requires more hand-offs to assemble a product than a traditional (serial) bucket brigade. These hand-offs may waste significant production capacity as each of them requires an exchange of work, which can be complicated and time consuming in practice. This motivates us to investigate the impact of hand-off times on the cellular bucket brigade's performance. We identify sufficient conditions to ensure no workers are idle in the long run and for the system to self-balance in a model with hand-off times. Our results suggest that even with significant hand-off times, the cellular bucket brigade can remain substantially (about 50%) more productive than the traditional bucket brigade especially if the team size is small and the workers' work velocities are close to their walk velocity.

Key words: bucket brigades; assembly lines; dynamic line balancing; work-sharing; self-organizing systems

1 Introduction

A common challenge in managing an assembly line is to allocate tasks for workers such that their workload is balanced. This is important because we want to ensure that every worker is constantly busy and the system's production capacity is fully utilized. One way to address this issue is to coordinate workers by forming a bucket brigade (Bartholdi and Eisenstein 1996a,b). In a bucket brigade each worker follows a simple rule: Continue to assemble a product along a line until either your colleague downstream takes over your work or you finish your work at the end of the line (if you are the last worker); then you walk back to get more work, either from your colleague upstream or from a buffer at the start of the line (if you are the first worker).

Bartholdi and Eisenstein (1996a) consider a model with deterministic work content. They assume that each worker has a deterministic, finite work velocity and an infinite walk-back velocity. They show that if the workers are sequenced from slowest to fastest according to their work velocities in the direction of production flow, then a bucket brigade will *self-balance* such that the hand-offs between any two neighboring workers will converge to a fixed location. Eventually, every worker will repeatedly work on a fixed segment of the assembly line. Furthermore, if the work content is continuously and uniformly distributed along the line, then the long-run average throughput will achieve the maximum possible value for the system. For generalizations of the above bucket brigade model, see Bartholdi et al. (1999, 2001, 2006, 2010), Armbruster and Gel (2006), Armbruster et al. (2007), Lim and Yang (2009), and Webster et al. (2011).

The ideas of bucket brigades have been applied in the production of garments, packaging of cellular phones, and assembly of tractors, large-screen televisions, and automotive electrical harnesses (Bartholdi and Eisenstein 1996a,b, Bartholdi and Eisenstein 2005, Villalobos et al. 1999a,b). Bucket brigades are effective and attractive for practitioners due to the following reasons: (1) The rule is easy for workers to follow. (2) They require neither a work-content model nor computation for work balance, which are necessary for any static work-allocation policy. (3) They constantly balance the workload for the workers subject to variability, and they spontaneously adapt to disruptions.

However, it remains challenging for a bucket brigade to perform efficiently in a *long* assembly line. This is because the bucket brigade rule requires each worker to walk back to get more work after he hands off his work to a colleague, or after he completes his work at the end of the line.

The travel to get more work is unproductive and is especially significant for a long assembly line. The significant, unproductive walk-back time makes the bucket brigade less efficient in this situation.

To improve the efficiency of a bucket brigade for a long assembly line, Lim (2011) proposes the ideas of *cellular bucket brigades* that eliminate the unproductive walk-back inherent in traditional bucket brigades. Under this new design, each worker works in both directions along an aisle: He assembles the product on one side of the aisle while he proceeds in one direction, but he works on the other side of the aisle while he proceeds in the reverse direction. When a worker proceeding in the forward direction meets his colleague, who is working in the backward direction, a hand-off occurs: The two workers exchange their work. Lim (2011) demonstrates that a cellular bucket brigade, even with fewer workers, can be significantly more productive than its traditional counterpart if the aisle is sufficiently narrow. However, the author does not consider hand-off times. Lim (2012) conducts a case study on order-picking by cellular bucket brigades using data from a distribution center in North America. Lim and Wu (2014) study the dynamics and performance of cellular bucket brigades on U-lines with discrete work stations.

Although the ideas are promising, a cellular bucket brigade requires more hand-offs to assemble a product than a traditional, serial bucket brigade. Furthermore, each hand-off corresponding to an exchange of work in the cellular bucket brigade can be complicated and time consuming in practice (Bartholdi and Eisenstein 2005). As a result, the cellular bucket brigade may waste significant production capacity in the hand-offs if it has many workers. This motivates us to investigate the impact of hand-off times on the dynamics and performance of the cellular bucket brigade. Compared to Lim (2011), the analysis of our model with hand-off times in this paper is more complicated. This is because some workers may be idle as they wait for their colleagues, who spend a long time in each hand-off. Fortunately, we find a sufficient condition to ensure that no workers are idle when the system operates on a unique fixed point. We also identify a sufficient condition for the system to converge to the fixed point.

It is worth noting that Bartholdi and Eisenstein (2005) consider a model where each worker spends a constant walk-back time and a constant hand-off time to get work from his upstream colleague. They assume the worker's constant walk-back time is independent of his upstream colleague. In contrast, we assume the time for each worker to meet with his colleague along the aisle depends on the distance between the two workers. Furthermore, any two adjacent workers in our model may spend different time durations in a hand-off between them, whereas Bartholdi and Eisenstein (2005) assume that the two workers spend the same amount of time in any hand-off between them. Our model is also related to Bartholdi et al. (2009), where each worker has a constant walk-back velocity and they are allowed to overtake or pass each other. The authors show that the system may behave chaotically if it is not configured properly. Compared to our model, they do not consider hand-off times.

This paper is organized as follows. Section 2 studies the dynamics and determine the long-run average throughput of a cellular bucket brigade with hand-off times. We consider two different hand-off time models. We do a similar study on a traditional bucket brigade with hand-off times in Section 3. Section 4 compares the performance of the cellular bucket brigade with its traditional counterpart and gives some concluding remarks. We find that even with significant hand-off times, the cellular bucket brigade can remain substantially (about 50%) more productive than the traditional bucket brigade if the team size is small and the workers' work velocities are close to their walk velocity.

2 Cellular bucket brigades with hand-off times

In this section, we analyze the dynamics and determine the long-run average throughput of a cellular bucket brigade with hand-off times.

2.1 Definitions and rules

Consider an assembly line shown in Figure 1 where work content is distributed along both sides of a central aisle with length 1/2. A *job* represents an instance of the product assembled. Each job is initiated from the start (left end) of the aisle, and is progressively assembled in the forward

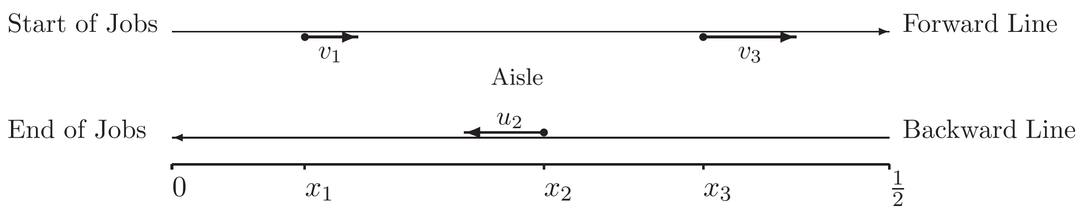


Figure 1: **An assembly line with work on both sides of an aisle.**

direction along the *forward line*. When the job reaches the end of the forward line, it is moved across the aisle and is then assembled in the backward direction along the *backward line*. The job is completed when it reaches the end of the backward line.

We assume the work content is distributed such that each worker i works with a constant velocity v_i along the forward line, and with a constant velocity u_i along the backward line. Both v_i and u_i are finite. For example, the forward line corresponds to the main assembly work and the backward line corresponds to a simpler packaging process such that $v_i < u_i$ for all i.

Let $x_i \in [0, 1/2]$ denote the coordinate of worker i *along* the aisle, for $i = 1, \ldots, n$. Figure 1 illustrates the coordinates and the velocities of three workers along the aisle. We require the workers to remain in a fixed sequence from 1 to n along the aisle so that $x_1 \leq x_2 \leq \ldots \leq x_n$. This requirement will not be a limitation of our model as the workers cannot overtake or pass each other if the aisle is sufficiently narrow. Workers $i-1$ and $i+1$ are called the *predecessor* and the *successor*, respectively, of worker i.

A worker i assembles his job as he moves along the forward line until he meets his successor, who is working in the backward direction. A *hand-off* between the two workers then occurs: The two workers exchange their work by first relinquishing their jobs, crossing the aisle, and then taking over each other's job. See Figure 2 for an example of a hand-off. To take over a job, each worker must fully understand what remains to be done before he assumes responsibility for its assembly. After exchanging their work, worker i works on the backward line while his successor proceeds in the forward direction.

When worker n, who is working forward, reaches the end of the forward line the system *resets* itself: Worker n carries his job, say job j, across the aisle, and starts processing it on the backward line. He continues to work backward until he meets worker $n-1$, who is working forward. After exchanging their work, worker $n-1$ continues job j in the backward direction until he meets and exchanges work with worker $n-2$, and so on until worker 1 completes job j at the end of the backward line. Worker 1 relinquishes the completed job, crosses the aisle, and then initiates a new job. Each reset triggers a sequence of hand-offs from the end to the start of the aisle (from right to left of Figure 1), followed by a completion of a job and an initiation of a new job. Only worker n can induce a reset and only worker 1 can initiate and complete a job. All jobs are released and are subsequently completed at the start of the aisle (left of Figure 1).

Since assembling a job requires $2(n-1)$ hand-offs and each hand-off involves two workers, the system may waste significant production capacity in the hand-offs if there are many workers. Furthermore, each hand-off can be complicated and time consuming depending on the work content. To capture the impact of the hand-offs on the system's performance, let h_i^- and h_i^+ denote the hand-off times for worker i to exchange work with his predecessor and successor respectively. These hand-off times can be determined as follows. To exchange work with a colleague, each worker must first relinquish his job and then accept another job from his colleague. Let r_i be the time required by worker i to relinquish his job. For example, this may represent the time required to mark down the last task done and clean the table for the next worker. For notational convenience, define $r_0 = r_{n+1} = 0$. Let s_i be the time required by worker i to accept work from his colleague. This includes the time to understand what remains to be done for the job. Note that the time r_i to relinquish a job and the time s_i to accept a job for each worker i are also adopted in Bartholdi and Eisenstein (2005).

Table 1 shows two possible ways to determine the hand-off times of each worker i. For Type I, the hand-off times of worker i are independent of his colleagues. Worker i starts accepting a new job as soon as he has relinquished his current job. For Type II, the hand-off times of worker i depend on his predecessor and successor. Worker i starts accepting a new job only after he and his colleague have completely relinquished their jobs. For both types, we have $h_1^- = r_1 + s_1$.

We assume the aisle is narrow such that the travel time across the aisle is negligible. Thus, we set $h_n^+ = 0$ for both types. It is noteworthy that h_i^+ may not equal h_{i+1}^-, and so workers i and $i+1$ may spend different time durations in the same hand-off between them. Let $h_i = h_i^- + h_i^+$ denote the total hand-off time of worker i. Since each hand-off in the system corresponds to an exchange of work between two workers, the hand-off time models discussed above are more complicated than that considered by Bartholdi and Eisenstein (2005).

Table 1: **Hand-off times of each worker i in a cellular bucket brigade.**

Type	h_i^-	h_i^+
I	$r_i + s_i$	$r_i + s_i$
II	$\max\{r_{i-1}, r_i\} + s_i$	$\max\{r_i, r_{i+1}\} + s_i$

Figure 2 illustrates the movements of two workers after a hand-off. Let $x_i^t \in [0, 1/2]$ denote the hand-off point along the aisle between worker i and his successor due to the t-th reset. The movements of workers i and $i+1$ are represented by bold solid arrows and dotted arrows respectively. The start and the end of each worker's path are represented by a circle and a square respectively. Worker i spends time h_i^+ to exchange work in the hand-off, whereas worker $i+1$ spends time h_{i+1}^-. After the hand-off, worker i works backward with velocity u_i until he meets worker $i-1$, who is working forward, at point x_{i-1}^t. After exchanging work with worker $i-1$, which takes time h_i^-, worker i works forward again with velocity v_i.

Meanwhile, worker $i+1$ assembles the job that he receives from worker i in the forward direction with velocity v_{i+1}. He meets worker $i+2$, who is working backward, at point x_{i+1}^{t+1}. After exchanging work with worker $i+2$, which takes time h_{i+1}^+, worker $i+1$ works backward again with velocity u_{i+1}. The next hand-off between workers i and $i+1$ occurs at point x_i^{t+1}.

To keep workers in the same sequence along the aisle, we need rules to handle situations in which a worker catches up with his successor or predecessor. If worker i catches up with worker $i+1$ when both workers are working forward (this is possible only if $v_i > v_{i+1}$), we say

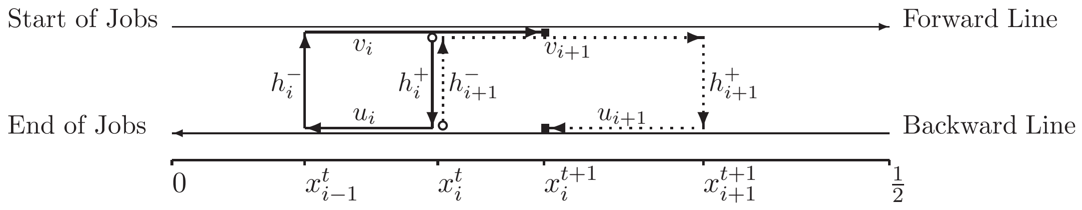


Figure 2: **Movements of workers i and $i+1$ between two successive hand-offs at points x_i^t and x_i^{t+1}.**

worker i is *blocked* by his successor. To maintain the same sequence of the workers, we require worker i to work with velocity v_{i+1} when he is blocked by worker $i+1$. During a hand-off, a worker's coordinate remains unchanged. Thus, if worker i catches up with worker $i+1$, who is in a hand-off at point x_{i+1}, then worker i *waits* for worker $i+1$ at the same point $x_i = x_{i+1}$.

Similarly, if worker i catches up with worker $i-1$ when both workers are working backward (this is possible only if $u_{i-1} < u_i$), then worker i is blocked by his predecessor. We require worker i to work with velocity u_{i-1} when he is blocked by his predecessor. If worker i catches up with worker $i-1$, who is in a hand-off at point x_{i-1}, then worker i waits for his predecessor at the same point $x_i = x_{i-1}$.

Each worker independently follows the cellular bucket brigade rules below along the aisle:

Work forward: Continue to assemble your job in the forward direction until

1. you exchange work with your successor, then **work backward**; or
2. you reach the end of the aisle if you are worker n, then **work backward**; or
3. you catch up with your successor, who is exchanging work with his colleague, then **wait**.

Work backward: Continue to assemble your job in the backward direction until

1. you exchange work with your predecessor, then **work forward**; or
2. you complete your job at the start of the aisle if you are worker 1, then initiate a new job and **work forward**; or
3. you catch up with your predecessor, who is exchanging work with his colleague, then **wait**.

Wait: Stay with your job,

1. if you are on the forward line, remain idle until your successor has finished exchanging work with his colleague, then exchange work with your successor and **work backward**; or
2. if you are on the backward line, remain idle until your predecessor has finished exchanging work with his colleague, then exchange work with your predecessor and **work forward**.

In the long run each worker i must travel as far forward as he does backward, and so he has an *effective* production rate of

$$\theta_i = \left(\frac{1/2}{v_i} + \frac{1/2}{u_i} \right)^{-1}.$$

The waiting rules above can potentially waste production capacity because they require workers to stand idle. However, as we will see from our analysis, the waiting rules will not be invoked in the long run for a properly configured system that satisfies the following condition:

$$\sum_{j=1}^{n} \theta_j (h_i - h_j) \leq 1, \qquad i = 1, \ldots, n. \tag{1}$$

Condition (1) can be interpreted as follows. The term $h_i - h_j$ represents the extra time spent in a hand-off by worker i compared to worker $j \neq i$. The term $\theta_j(h_i - h_j)$ represents the work that can be done by worker j during this extra time. Condition (1) requires that during the extra time worker i spends in a hand-off, the total work that can be accomplished by all the other workers must not exceed the total work content (which equals 1), for $i = 1, \ldots, n$. Otherwise, some workers will be constantly idle as they wait for their predecessors or successors. This condition can be easily satisfied if the total hand-off times of different workers are similar. It is worth noting that Condition (1) always holds if all the workers have the same total hand-off time ($h_1 = h_2 = \cdots = h_n$).

2.2 Dynamics and throughput

Let $\mathbf{x}^t = \left(x_1^t, x_2^t, \ldots, x_{n-1}^t \right)$. Since the system resets itself when worker n reaches the end of the aisle, we have $x_n^t = 1/2$ for all t. For convenience, we define $x_0^t = 0$ for all t. Let f be a function, defined implicitly by the cellular bucket brigade rules, such that $\mathbf{x}^{t+1} = f(\mathbf{x}^t)$. We say $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_{n-1}^*)$ is a *fixed point* if $\mathbf{x}^* = f(\mathbf{x}^*)$. It is straightforward to see that no blocking can occur on a fixed point in a cellular bucket brigade. The following lemma shows that if Condition (1) holds, then a unique fixed point of the hand-off locations exists. Furthermore, there is no waiting and thus no workers are idle on the fixed point. All proofs can be found in the online appendix.

Lemma 1. *If Condition (1) holds, then there exists a unique fixed point* $\mathbf{x}^*$ *on which no waiting occurs in a cellular bucket brigade, where*

$$x_i^* = \frac{1}{2}\left\{ \frac{\sum_{j=1}^{i} \theta_j}{\sum_{j=1}^{n} \theta_j} \left[1 - \sum_{j=2}^{n} \theta_j (h_1 - h_j) \right] + \sum_{j=2}^{i} \theta_j (h_1 - h_j) \right\},$$

for $i = 1, \ldots, n$.

We note that if Condition (1) does not hold then a cellular bucket brigade may have multiple fixed points, which depend on its initial state (the locations and the directions of workers). The system's production capacity is not fully used because waiting occurs on these fixed points.

Figure 3 shows a four-worker cellular bucket brigade operating on its fixed point. Hand-offs occur at points x_1^*, x_2^*, and x_3^* along the aisle. When the system operates on the fixed point each worker i repeats a simple loop for each job assembled: He exchanges work with his successor at point x_i^*. He then works backward until point x_{i-1}^*, where he exchanges work with his predecessor. After that he works forward until point x_i^*, where he completes the loop. In Figure 3 the loops of workers 1, 2, 3, and 4 are shown from left to right. Since each worker moves in a "cell" when the system operates on a fixed point, we call the system cellular bucket brigade. Lemma 2 determines the system's average throughput on its fixed point.

Lemma 2. *The average throughput of a cellular bucket brigade on the fixed point* $\mathbf{x}^*$ *is*

$$\rho_c = \frac{\sum_{j=1}^{n} \theta_j}{1 + \sum_{j=1}^{n} \theta_j h_j}.$$

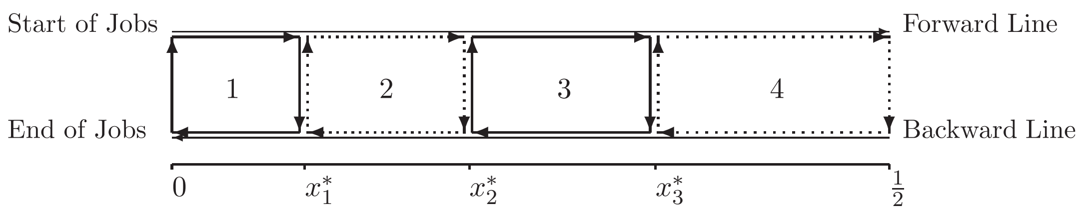


Figure 3: **Fixed point of a four-worker cellular bucket brigade.**

Lemma 2 shows that the average throughput of the system equals its total production capacity $\sum_{j=1}^{n} \theta_j$ offset by the waste $\sum_{j=1}^{n} \theta_j h_j$ incurred in the hand-offs.

Theorem 1 shows that if a cellular bucket brigade is configured properly then the fixed point $\mathbf{x}^*$ is an attractor (Alligood et al. 1996), which means the system will always be attracted to $\mathbf{x}^*$ if it is sufficiently close to the fixed point.

Theorem 1. *If Condition (1) holds and*

$$\frac{1}{v_1} - \frac{1}{u_1} > \frac{1}{v_2} - \frac{1}{u_2} > \cdots > \frac{1}{v_n} - \frac{1}{u_n}, \tag{2}$$

then the fixed point $\mathbf{x}^*$ *of a cellular bucket brigade is an attractor.*

The proof of Theorem 1 shows that a cellular bucket brigade will always converge to the fixed point $\mathbf{x}^*$ if it is sufficiently close to the fixed point. The fixed point $\mathbf{x}^*$ is at least a *local attractor* and will assert itself to restore balance after perturbations, as long as they are not too disruptive. As supported by a myriad of simulation results, we believe the fixed point $\mathbf{x}^*$ is also a *global attractor* if Conditions (1) and (2) hold, which means a cellular bucket brigade will always converge to the fixed point given any initial state.

The term $1/v_i - 1/u_i$ in Condition (2) represents the difference of the forward and the backward work times for worker i. Theorem 1 implies that the workers should be sequenced in the forward direction along the aisle in decreasing order of the work-time difference. One should assign a lower index to a worker (he should work nearer to the start of the aisle) who has a larger difference of the forward and the backward work times.

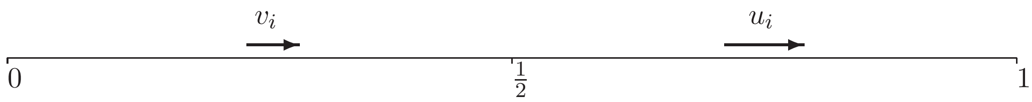


Figure 4: **A serial assembly line.**

3 Serial bucket brigades with hand-off times

Before we compare a cellular bucket brigade with its traditional counterpart, we need to study the dynamics and find the long-run average throughput of a traditional (serial) bucket brigade with hand-off times.

3.1 Definitions and rules

Consider an assembly line shown in Figure 4. We assume all jobs are released from the start (left end) of the line. We conceptualize the assembly line as a real line segment with length 1. Each job is initiated at point 0 and is progressively assembled along the line until it is completed at point 1. We distribute the work content such that each worker i works with velocities v_i and u_i in the intervals $[0, 1/2)$ and $[1/2, 1]$ respectively, for $i = 1 \ldots, n$.

Under the traditional bucket brigade design, workers remain in a fixed sequence from 1 to n along the production flow shown in Figure 4. Workers $i-1$ and $i+1$ are called the predecessor and successor, respectively, of worker i. Each worker works forward until he hands off his job to his successor. When worker n completes his job at point 1, the system 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 job at point 0. Since each job is transferred from one worker to his successor in a sequential way, we call this a serial bucket brigade. We assume each worker spends significant time to walk to his predecessor because the line can be long. As a result, the reset is not instantaneous. To see the effect of this unproductive travel without making the final result too complicated, we assume all the workers walk back with a finite, constant velocity w to receive work from their predecessors. Thus, the time for worker i

to walk back to his predecessor depends on the locations of both workers. This is different from the model studied in Bartholdi and Eisenstein (2005), where the walk-back time of worker i is a worker-specific constant (independent of the workers' locations).

Let g_i^- and g_i^+ denote the hand-off times for worker i to receive work from his predecessor and to relinquish work to his successor respectively. Table 2 shows two possible ways to determine the hand-off times of worker i in a serial bucket brigade. The types of hand-off times considered are consistent with that of Table 1. For Type I, the hand-off times of worker i are independent of his colleagues. For Type II, worker i starts accepting a job only after his predecessor has relinquished it. Since g_i^+ may not equal g_{i+1}^-, workers i and $i+1$ may spend different time durations in a hand-off between them. Let $g_i = g_i^- + g_i^+$ denote the total hand-off time of worker i in a serial bucket brigade. It is noteworthy that the above hand-off time models are different from that adopted by Bartholdi and Eisenstein (2005), where any two adjacent workers spend the same amount of time in any hand-off between them.

Table 2: **Hand-off times of each worker i in a serial bucket brigade.**

Type	g_i^-	g_i^+
I	s_i	r_i
II	$r_{i-1} + s_i$	r_i

Tables 1 and 2 show that h_i^- and h_i^+ are no smaller than g_i^- and g_i^+, respectively, in each type. Each hand-off in a cellular bucket brigade corresponds to an exchange of work between two workers, and therefore hand-offs are more complicated and time consuming than those in a serial bucket brigade.

For workers to maintain the same sequence along the line in Figure 4, worker i is blocked by his successor if the former catches up with the latter when both of them are working forward (this is possible only if $v_i > v_{i+1}$ or $u_i > u_{i+1}$). In this situation, worker i works with the velocity of worker $i+1$. If worker i catches up with worker $i+1$ when the latter is in a hand-off at point x_{i+1}, then worker i waits for worker $i+1$ at the same point $x_i = x_{i+1}$. Similarly, if worker i, who is walking backward, catches up with worker $i-1$, who is in a hand-off at point x_{i-1}, then worker i waits for worker $i-1$ at the same point $x_i = x_{i-1}$.

3.2 Dynamics and throughput

Let $\mathbf{x}^t = \left(x_1^t, x_2^t, \dots, x_{n-1}^t\right)$, where x_i^t is the location at which worker i hands off work to worker $i+1$ due to the t-th reset. Since worker n finishes each job at point 1, we have $x_n^t = 1$ for all t. For convenience, we define $x_0^t = 0$ for all t. Define $\psi_i = (1/v_i + 1/w)^{-1}$ and $\phi_i = (1/u_i + 1/w)^{-1}$ as the effective production rates of worker i in the intervals $[0, 1/2)$ and $[1/2, 1]$ respectively, for $i = 1 \dots, n$. Let worker k be the one that repeatedly crosses point 1/2 on the line in Figure 4 when the system operates on a fixed point. Similar to Condition (1), the following condition ensures that no waiting occurs on a fixed point of the serial bucket brigade, for $i \in [1, n]$:

$$\begin{cases} \frac{1}{\psi_k} \sum\limits_{j=1}^{k-1} \psi_j (g_i - g_j) + (g_i - g_k) + \frac{1}{\phi_k} \sum\limits_{j=k+1}^{n} \phi_j (g_i - g_j) \le \frac{1}{2} \left(\frac{1}{\psi_k} + \frac{1}{\phi_k} \right), & i \ne k; \\ \sum\limits_{j=1}^{k-1} \psi_j (g_i - g_j) + \sum\limits_{j=k+1}^{n} \phi_j \left(g_i - g_j\right) \le 1, & i = k. \end{cases} \tag{3}$$

Condition (3) has a similar interpretation as Condition (1). The term $g_i - g_j$ represents the extra time spent in a hand-off by worker i compared to worker $j \ne i$. For workers $j = 1, \dots, k-1$, the sum $\sum_{j=1}^{k-1} \psi_j (g_i - g_j)$ represents the total work that they can accomplish in the line interval $[0, 1/2)$ during the extra time worker i spends in a hand-off. Likewise, for workers $j = k+1, \dots, n$, the sum $\sum_{j=k+1}^{n} \phi_j (g_i - g_j)$ represents the total work that they can accomplish in the line interval $[1/2, 1]$ during the extra time worker i spends in a hand-off. Since worker k works in both intervals $[0, 1/2)$ and $[1/2, 1]$, we use him as a standard worker. Thus, the left hand side of the first inequality of Condition (3) represents the total work that can be accomplished by all workers $j \ne i$ during the extra time worker i spends in a hand-off, normalized by the production rates of worker k. This normalized total work should not exceed the system's total work content normalized by the production rates of worker k.

The following lemma shows that if Condition (3) holds, then a unique fixed point of hand-off locations exists in a serial bucket brigade and there is no waiting on the fixed point.

Lemma 3. *If Condition (3) holds, then there exists a unique fixed point* $\mathbf{x}^*$ *on which no waiting occurs in a serial bucket brigade, where* $\mathbf{x}^*$ *is determined as follows:*

If $\frac{1}{2}(1 - \frac{1}{\psi_1}\sum_{j=2}^n \phi_j) \geq \sum_{j=2}^n \phi_j(g_1 - g_j)$*, then* $k = 1$ *(worker 1 repeatedly crosses point 1/2) and*

$$x_1^* = \frac{1 - \frac{1}{2}\left(\frac{1}{\psi_1} - \frac{1}{\phi_1}\right)\sum_{j=2}^n \phi_j - \sum_{j=2}^n \phi_j(g_1 - g_j)}{\frac{1}{\phi_1}\sum_{j=1}^n \phi_j};$$

$$x_i^* = \frac{x_1^*}{\phi_1}\sum_{j=1}^i \phi_j + \frac{1}{2}\left(\frac{1}{\psi_1} - \frac{1}{\phi_1}\right)\sum_{j=2}^i \phi_j + \sum_{j=2}^i \phi_j(g_1 - g_j), \quad i = 2, \ldots, n-1.$$

Otherwise,

$$x_1^* = \psi_1 \frac{\frac{1}{2}\left(\frac{1}{\psi_k} + \frac{1}{\phi_k}\right) - \frac{1}{\psi_k}\sum_{j=2}^{k-1}\psi_j(g_1 - g_j) - \frac{1}{\phi_k}\sum_{j=k}^n \phi_j(g_1 - g_j)}{\frac{1}{\psi_k}\sum_{j=1}^{k-1}\psi_j + \frac{1}{\phi_k}\sum_{j=k}^n \phi_j};$$

$$x_i^* = \frac{x_1^*}{\psi_1}\sum_{j=1}^i \psi_j + \sum_{j=2}^i \psi_j(g_1 - g_j), \quad i = 2, \ldots, k-1;$$

$$x_i^* = \phi_k\left[\frac{x_1^*}{\psi_1}\left(\frac{1}{\psi_k}\sum_{j=1}^{k-1}\psi_j + \frac{1}{\phi_k}\sum_{j=k}^i \phi_j\right) - \frac{1}{2}\left(\frac{1}{\psi_k} - \frac{1}{\phi_k}\right) + \frac{1}{\psi_k}\sum_{j=2}^{k-1}\psi_j(g_1 - g_j) + \frac{1}{\phi_k}\sum_{j=k}^i \phi_j(g_1 - g_j)\right], \quad i = k, \ldots, n-1;$$

and k *is the smallest index such that* $\frac{x_1^*}{\psi_1}\sum_{j=1}^k \psi_j + \sum_{j=2}^k \psi_j(g_1 - g_j) \geq \frac{1}{2}$.

Upon the fixed point, each worker $i = 1, \ldots, k-1$ repeatedly works on an interval that lies in $[0, 1/2)$ with velocity v_i, while each worker $i = k+1, \ldots, n$ repeatedly covers an interval in $[1/2, 1]$ with velocity u_i. Worker k is the only one that repeatedly works in both intervals $[0, 1/2)$ and $[1/2, 1]$ with velocities v_k and u_k respectively.

Lemma 4 determines the average throughput of a serial bucket brigade on its fixed point.

Lemma 4. *The average throughput of a serial bucket brigade on the fixed point* $\mathbf{x}^*$ *is*

$$\rho_s = \begin{cases} [(1/2)/\psi_1 + (x_1^* - 1/2)/\phi_1 + g_1]^{-1}, & \textit{if } k = 1; \\ (x_1^*/\psi_1 + g_1)^{-1}, & \textit{otherwise.} \end{cases}$$

Note that the fixed point and throughput have different expressions if $k = 1$ (worker 1 is the one that repeatedly crosses point 1/2). Theorem 2 identifies the conditions for a serial bucket brigade to self-balance.

Theorem 2. *If Condition (3) holds and* $v_1 < v_2 < \cdots < v_k$ *and* $u_k < u_{k+1} < \cdots < u_n$*, then the fixed point* $\mathbf{x}^*$ *of a serial bucket brigade is an attractor.*

Theorem 2 shows that the fixed point $\mathbf{x}^*$ is at least a local attractor: If the system is properly configured and is sufficiently close to $\mathbf{x}^*$ such that worker k is the only one that crosses point 1/2 all the time, then the serial bucket brigade will converge to the fixed point.

4 Comparison and conclusion

Which design of bucket brigade assembly lines is more productive when hand-off times are significant? A cellular bucket brigade requires $2(n-1)$ hand-offs per job, compared to $n-1$ hand-offs per job for a serial bucket brigade. Since a cellular bucket brigade requires twice as many hand-offs to complete a job and each hand-off is more time consuming, the system may waste significant production capacity if it has too many workers (due to too many hand-offs per job). On the other hand, workers in a serial bucket brigade perform substantial unproductive travel to get work from their colleagues. The production capacity may be wasted considerably if the unproductive travel time is significant compared to the time to assemble a job.

We compare the throughput expressions in Lemmas 2 and 4 and find that the cellular bucket brigade can be significantly more productive than its traditional counterpart if the team size n is small. For example, the former is about 50% more productive than the latter for a two-worker team with the work velocities at about 80% of the walk velocity, and the time to relinquish and the time to accept a job at approximately 10% of the time to walk the line. This is because a small team does not require many hand-offs to assemble a job, which causes the workers in a serial bucket brigade to spend more time in unproductive travel than in hand-offs. The cellular bucket brigade is significantly more efficient in this situation as it eliminates the unproductive travel. As the team size n increases, the performance of the cellular bucket brigade relative to that of the serial bucket brigade deteriorates because the number of hand-offs per job in the former grows two times as fast as that in the latter.

The cellular bucket brigade is also substantially more productive than its traditional counterpart if the work velocities v_i and u_i are close to the walk velocity w. This is because in this situation the unproductive walk-back time in the serial bucket brigade becomes very significant compared to the time to assemble jobs. Thus, it is important to eliminate the unproductive walk-back, which makes the cellular bucket brigade more effective.

The above observations suggest that *the cellular bucket brigade is especially effective for a small team with the work velocities close to the walk velocity.* In the context of order-picking in warehouses, the work velocities of workers are comparable to their walk velocity if the pick density is low. This implies that the cellular bucket brigade is especially effective for order-picking if the team size is small and the pick density is low. Our analysis also suggests that the cellular bucket brigade is especially effective if each worker has a small walk velocity or spends a short time per hand-off. Details can be found in the online appendix.

The production flow defined in Figure 1 requires every worker to assemble a job on their *left* as they proceed. An equivalent design is to require every worker to work on the line on their *right*. Depending on the actual environment, one may be more suitable than the other.

Finally, our model can be generalized so that workers change their work velocities at some arbitrary point on the assembly line, and each worker i walks with a velocity w_i. This generalized model leads to similar, but more complicated, expressions of the fixed point and throughput as well as conditions for no waiting and for convergence to the fixed point. These generalized results can be derived based on an analysis similar to that in the Electronic Companion of Lim (2011) and are therefore omitted here. The simplified model discussed in this paper captures the main ideas of both cellular and serial bucket brigades with hand-off times and the essence of their dynamics, which allows us to extract useful insights without complex algebraic expressions.

References

Alligood, K.T., T.D. Sauer, J.A. Yorke. 1996. *Chaos: An Introduction to Dynamical Systems*. Springer, New York. ISBN 0-387-94677-2.

Armbruster, D., E.S. Gel. 2006. Bucket brigades revisited: Are they always effective? *Eur. J. Oper. Res.* **172**(1) 213–229.

Armbruster, D., E.S. Gel, J. Murakami. 2007. Bucket brigades with worker learning. *Eur. J. Oper. Res.* **176**(1) 264–274.

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

Bartholdi, J.J. III, D.D. Eisenstein. 1996a. A production line that balances itself. *Oper. Res.* **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 Oper. Management* **7**(2) 121–129.

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

Bartholdi, J.J. III, D.D. Eisenstein, Y.F. Lim. 2006. Bucket brigades on in-tree assembly networks. *Eur. J. Oper. Res.* **168**(3) 870–879.

Bartholdi, J.J. III, D.D. Eisenstein, Y.F. Lim. 2009. Deterministic chaos in a model of discrete manufacturing. *Naval Res. Logist.* **56**(4) 293–299.

Bartholdi, J.J. III, D.D. Eisenstein, Y.F. Lim. 2010. Self-organizing logistics systems. *Ann. Rev. in Control* **34**(1) 111–117.

Lim, Y.F. 2011. Cellular bucket brigades. *Oper. Res.* **59**(6) 1539–1545.

Lim, Y.F. 2012. Order-picking by cellular bucket brigades: A case study. R. Manzini, ed. *Warehousing in The Global Supply Chain*. Springer-Verlag, London, 71–85.

Lim, Y.F., Y. Wu. 2014. Cellular bucket brigades on U-lines with discrete work stations. *Production Oper. Management* **23**(7), 1113–1128.

Lim, Y.F., K.K. Yang. 2009. Maximizing throughput of bucket brigades on discrete work stations. *Production Oper. Management* **18**(1) 48–59.

Martelli, M. 1999. *Introduction to Discrete Dynamical Systems and Chaos*. John Wiley & Sons, Inc., New York. ISBN 0-471-31975-9.

Ross, S.M. 1996. *Stochastic Processes, Second Edition*. John Wiley & Sons, Inc., New York. ISBN 0-471-12062-6.

Villalobos, J.R., L.F. Muñoz, L. Mar. 1999a. Assembly line designs that reduce the impact of personnel turnover. *Proc. 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.

Webster, S., R.A. Ruben, K.K. Yang. 2012. Impact of storage assignment decisions on a bucket brigade order picking line, *Production Oper. Management* **21**(2), 276–290.

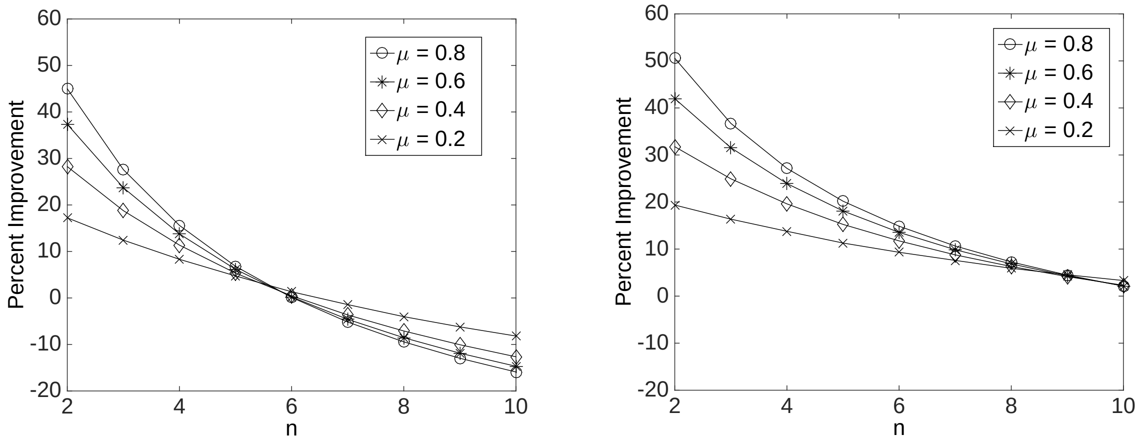
Online appendix

A Comparing the cellular and the serial bucket brigades

Define the *percent improvement* in throughput by a cellular bucket brigade over its traditional counterpart as $(\rho_c - \rho_s)/\rho_s \times 100\%$. Figures 5(a) and 5(b) show the percent improvement for Type-I and Type-II hand-offs respectively. For both figures, we set the walk velocity $w = 1.0$ and the work velocities $v_i = \mu w + [i-(n+1)/2] \times 0.01$ and $u_i = (\mu+0.1)w + [i-(n+1)/2] \times 0.01$, for $i = 1, \ldots, n$, where the scaling factor $\mu \in (0, 1)$ is chosen so that the largest work velocity in the team is smaller than w. We report the case where faster workers spend less time in hand-offs so that $r_i = s_i = \tau - [i - (n+1)/2] \times 0.001$, for $i = 1, \ldots, n$. We set $\tau = 0.1$. Similar results are observed for the case where faster workers spend more time in hand-offs.

Figure 5 shows that cellular bucket brigades significantly outperform their traditional counterparts when the team size n is small, with an improvement as high as 50%. However, as the team size gets larger the percent improvement decreases and it can be negative in some cases (see Figure 5(a)). If the team size n is small, the workers in a serial bucket brigade spend more time in unproductive travel than in hand-offs. Thus, the design of cellular bucket brigades (which eliminates the unproductive travel) leads to a significant improvement in throughput. As the team size increases, it requires more hand-offs to complete each job for both serial and cellular bucket brigades. Specifically, the number of hand-offs per job in the cellular bucket brigade grows two times as fast as that in the serial bucket brigade. Furthermore, workers spend a longer time in each hand-off in the cellular bucket brigade ($h_i^- \geq g_i^-$ and $h_i^+ \geq g_i^+$). This causes the percent improvement to drop as n increases.

It is noteworthy that the scaling factor μ also plays a significant role in Figure 5. If μ is large, the workers' work velocities v_i and u_i are close to their walk velocity w. This implies that the workers' *walk time* (the total time spent in unproductive travel per job) is significant compared to their *work time* (the total time spent in assembling a job). Under this situation, it is important to eliminate the unproductive travel. Both Figures 5(a) and 5(b) show that the



(a)

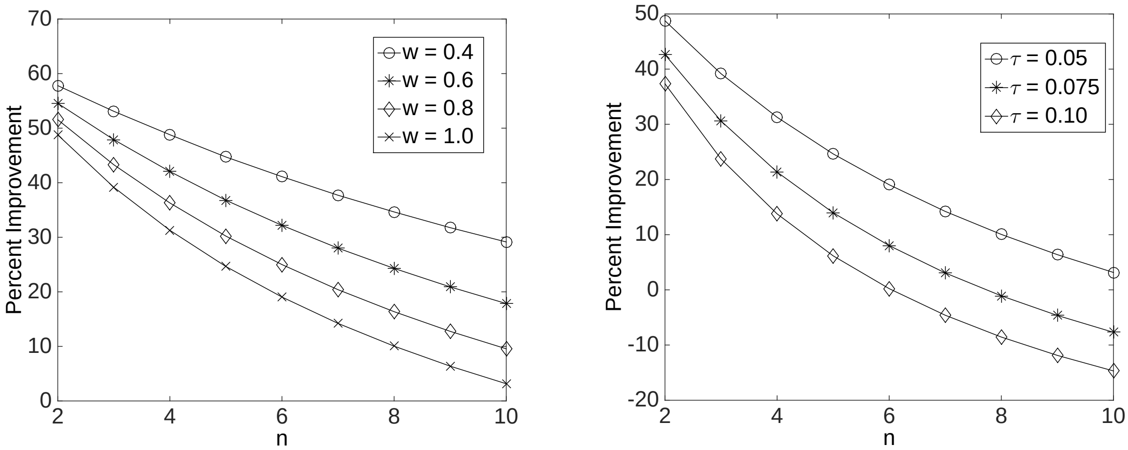
(b)

Figure 5: **Throughput improvement by cellular bucket brigades.**

cellular bucket brigade is the most effective (with the highest percent improvement) when the team size is small and μ is large. However, as the team size increases the percent improvement drops with a faster rate for a larger μ.

Figure 5 also suggests that the percent improvement is higher for Type-II hand-offs (Figure 5(b)) than Type-I hand-offs (Figure 5(a)). This is because the hand-off time g_i^- of each worker i in a serial bucket brigade is higher for Type II than Type I (see Table 2). Thus, the serial bucket brigade is less productive for Type-II hand-offs than Type-I hand-offs. In contrast, under a cellular bucket brigade the hand-off times are quite similar for both Types I and II (see Table 1). This implies that the throughput of the cellular bucket brigade is less sensitive to the hand-off type. As a result, the cellular bucket brigade yields a larger percent improvement over its traditional counterpart for Type-II hand-offs.

We also investigate the effects of the walk velocity w and the parameter τ, which represents the average time to relinquish or to accept work by a worker. Figure 6(a) shows that the percent improvement increases as the walk velocity becomes smaller, where we set $\mu = 0.6$ and $\tau = 0.05$. Figure 6(b) illustrates the impact of τ, where we set $\mu = 0.6$ and $w = 1.0$. As the average time to relinquish and to accept work gets shorter, the percent improvement increases. The above



(a) (b)

Figure 6: **Effects of the walk velocity w and the average time to relinquish (or to accept) work τ.**

observations suggest that the cellular bucket brigade is significantly more productive than the serial bucket brigade when each worker has a small walk velocity or spends a short time per hand-off. These observations are confirmed in Figure 7 as we vary both w and τ with $\mu = 0.6$.

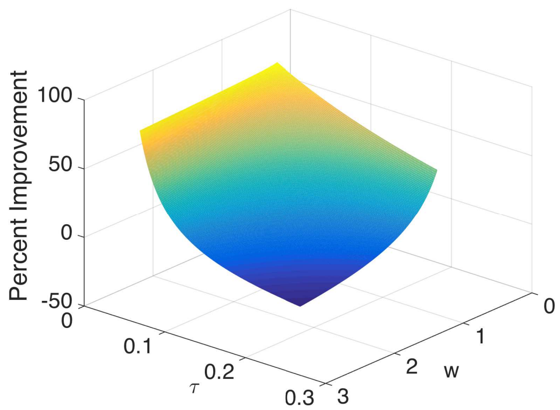


Figure 7: **Sensitivity of the throughput improvement to w and τ.**

B Technical details

B.1 Proof of Lemma 1

Proof. We first show that if Condition (1) holds then no waiting occurs on a fixed point $\mathbf{x}^*$. We prove by contradiction. Assume Condition (1) holds and suppose worker $i \in [1, n-1]$ causes his *successor* to wait. This implies that $x^*_{i-1} = x^*_i$ and

$$\begin{aligned} h_i &> h^-_{i+1} + \frac{x^*_{i+1} - x^*_i}{v_{i+1}} + h^+_{i+1} + \frac{x^*_{i+1} - x^*_i}{u_{i+1}}; \\ x^*_i &> x^*_{i+1} - \frac{\theta_{i+1}}{2}(h_i - h_{i+1}). \end{aligned} \tag{4}$$

For other workers $j \neq i+1$ on the fixed point $\mathbf{x}^*$, we have

$$\begin{aligned} h_i &= h^-_j + \frac{x^*_j - x^*_{j-1}}{v_j} + h^+_j + \frac{x^*_j - x^*_{j-1}}{u_j}; \\ x^*_j &= x^*_{j-1} + \frac{\theta_j}{2}(h_i - h_j). \end{aligned} \tag{5}$$

For $j = 1, \ldots, i-1$, Equation (5) leads to the following result:

$$x^*_{i-1} = \frac{1}{2}\sum_{j=1}^{i-1} \theta_j (h_i - h_j).$$

For $j = i+2, \ldots, n$, since $x^*_n = 1/2$, Equation (5) leads to the following result:

$$x^*_{i+1} = \frac{1}{2} - \frac{1}{2}\sum_{j=i+2}^{n} \theta_j (h_i - h_j).$$

Since $x^*_{i-1} = x^*_i$, combining the last two equations with Inequality (4), we have

$$\begin{aligned} \frac{1}{2}\sum_{j=1}^{i-1} \theta_j (h_i - h_j) &> \frac{1}{2} - \frac{1}{2}\sum_{j=i+1}^{n} \theta_j (h_i - h_j); \\ \sum_{j=1}^{n} \theta_j (h_i - h_j) &> 1. \end{aligned}$$

This contradicts Condition (1). Thus, if Condition (1) holds, worker $i \in [1, n-1]$ cannot cause his successor to wait on the fixed point $\mathbf{x}^*$. It can be proved in a similar way that if Condition (1) holds, worker $i \in [2, n]$ cannot cause his *predecessor* to wait on the fixed point $\mathbf{x}^*$.

Thus, if Condition (1) holds there is no waiting on the fixed point $\mathbf{x}^*$ on which each worker i repeats a simple loop for each job assembled: He exchanges work with his successor at point x_i^*. He then works backward with velocity u_i until he reaches point x_{i-1}^*, where he exchanges work with his predecessor. After the hand-off he works forward with velocity v_i until he completes the loop at point x_i^*. The fixed point $\mathbf{x}^*$ can be found by solving the following equations:

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

for $i = 1, \ldots, n-1$. Lemma 1 follows by simple algebra. □

B.2 Proof of Lemma 2

Proof. On the fixed point the system assembles a job every time worker 1 completes a simple loop. The average throughput is $\rho_c = \left(h_1^- + x_1^*/v_1 + h_1^+ + x_1^*/u_1\right)^{-1}$. □

B.3 Proof of Theorem 1

Proof. The proof is similar to the proof of Theorem 1 in the Electronic Companion of Lim (2011) and the proof of Theorem 2 in Bartholdi et al. (2009). Iteration t follows the sequence of hand-off points from the end to the start of the aisle caused by the t-th reset. A hand-off occurs at point x_i^t where worker i, who is working forward, meets worker $i+1$, who is working backward. Since the time spent by each worker from one iteration to the next (see Figure 2) is the same, we have

$$h_i^+ + \frac{x_i^t - x_{i-1}^t}{u_i} + h_i^- + \frac{x_i^{t+1} - x_{i-1}^t}{v_i} = h_{i+1}^- + \frac{x_{i+1}^{t+1} - x_i^t}{v_{i+1}} + h_{i+1}^+ + \frac{x_{i+1}^{t+1} - x_i^{t+1}}{u_{i+1}}, \tag{6}$$

for $i = 1, \ldots, n-1$.

Rewriting Equation (6) yields:

$$x_i^{t+1} = \frac{1/v_i + 1/u_i}{1/v_i + 1/u_{i+1}} x_{i-1}^t - \frac{1/v_{i+1} + 1/u_i}{1/v_i + 1/u_{i+1}} x_i^t + \frac{1/v_{i+1} + 1/u_{i+1}}{1/v_i + 1/u_{i+1}} x_{i+1}^{t+1} - \frac{h_i - h_{i+1}}{1/v_i + 1/u_{i+1}},$$

for $i = 1, \ldots, n-1$. Or we can write

$$x_i^{t+1} = (1+\alpha_i)\gamma_i x_{i-1}^t - \alpha_i x_i^t + (1+\alpha_i)(1-\gamma_i) x_{i+1}^{t+1} - \frac{h_i - h_{i+1}}{1/v_i + 1/u_{i+1}}, \tag{7}$$

where

$$\alpha_i = \frac{1/v_{i+1} + 1/u_i}{1/v_i + 1/u_{i+1}},$$
$$\gamma_i = \frac{1/v_i + 1/u_i}{1/v_i + 1/u_i + 1/v_{i+1} + 1/u_{i+1}},$$

for $i = 1, \ldots, n-1$. Note that $0 < \alpha_i < 1$ corresponds to Condition (2).

Equation (7) can be expressed as an affine system (Martelli 1999):

$$\mathbf{y}^{t+1} = A\mathbf{y}^t + \mathbf{b},$$

where $\mathbf{y}^t = \left(x_1^t, x_2^t, \ldots, x_{n-2}^t, x_{n-1}^{t+1}\right)^T$. The first $n-2$ components of the vector $\mathbf{y}^t$ correspond to the last $n-2$ hand-offs of iteration t and the last component corresponds to the first hand-off of iteration $t+1$. The matrix A can be factored as $A = A_{n-1}A_1A_2\ldots A_{n-2}$, where each matrix A_i updates x_i^t according to Equation (7), and

$$\mathbf{b} = A_{n-1}\left[\sum_{i=1}^{n-2}\left(\prod_{j=1}^{i-1} A_j\right)\mathbf{b}_i\right] + \mathbf{b}_{n-1},$$

where $\mathbf{b}_i$ is a zero vector except for the i-th component, which equals $-(h_i - h_{i+1})/(1/v_i + 1/u_{i+1})$, for $i = 1, \ldots, n-2$, and $\mathbf{b}_{n-1}$ is also a zero vector except for the $(n-1)$-st component, which equals $-(h_{n-1} - h_n)/(1/v_{n-1} + 1/u_n) + (1 + \alpha_{n-1})(1 - \gamma_{n-1})/2$. In this way we first update x_{n-2}^t, then x_{n-3}^t, and so on until x_1^t, and then finally x_{n-1}^{t+1}.

Each matrix A_i is an identity matrix except for row i. Each $A_2, A_3, \ldots, A_{n-2}$ has three non-zero terms in row i that sum to 1, with values $(1 + \alpha_i)\gamma_i$, $-\alpha_i$, and $(1 + \alpha_i)(1 - \gamma_i)$ in columns $i-1$, i, and $i+1$ respectively. For A_1 the first term $(1 + \alpha_1)\gamma_1 > 0$ is omitted from row 1, and thus the sum of the first row has absolute value less than 1. For A_{n-1} the last term $(1 + \alpha_{n-1})(1 - \gamma_{n-1}) > 0$ is omitted from row $n-1$, thus the sum of the last row has absolute value less than 1.

For the full transition matrix A, all eigenvalues have modulus less than one. In short, this follows because each $A_2, A_3, \ldots, A_{n-2}$ can be replaced by a stochastic matrix, while both A_1 and A_{n-1} can be replaced by a strictly sub-stochastic matrix. Since all states communicate, the system tends to the zero matrix. Thus, the orbit $\mathbf{y}^0, \mathbf{y}^1, \mathbf{y}^2, \ldots$ converges to the unique fixed point $\mathbf{y}^*$ of hand-off locations. (See, for example, Martelli (1999) for dynamics of affine systems.) □

B.4 Proof of Lemma 3

Proof. We first show that if Condition (3) holds then no waiting occurs on a fixed point $\mathbf{x}^*$. We prove by contradiction. Assume Condition (3) holds and suppose worker $i \in [2, k-1]$ causes his *predecessor* to wait. This implies that $x^*_{i-1} = x^*_i$ and

$$\begin{aligned} g_i &> g^-_{i-1} + \frac{x^*_{i-1} - x^*_{i-2}}{v_{i-1}} + g^+_{i-1} + \frac{x^*_{i-1} - x^*_{i-2}}{w}; \\ x^*_{i-1} &< x^*_{i-2} + \psi_{i-1}(g_i - g_{i-1}). \end{aligned} \tag{8}$$

For other workers $j \neq i-1$ on the fixed point $\mathbf{x}^*$, we have

$$g_i = g^-_j + \frac{x^*_j - x^*_{j-1}}{v_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = 1, \ldots, i-2; \tag{9}$$

$$g_i = g^-_j + \frac{x^*_j - x^*_{j-1}}{v_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = i+1, \ldots, k-1; \tag{10}$$

$$g_i = g^-_j + \frac{1/2 - x^*_{j-1}}{v_j} + \frac{x^*_j - 1/2}{u_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = k; \tag{11}$$

$$g_i = g^-_j + \frac{x^*_j - x^*_{j-1}}{u_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = k+1, \ldots, n. \tag{12}$$

Equations (9) lead to the following result:

$$x^*_{i-2} = \sum_{j=1}^{i-2} \psi_j(g_i - g_j).$$

Since $x^*_n = 1$, Equations (10)–(12) lead to the following result:

$$x^*_i = \frac{1}{2} - \sum_{j=i+1}^{k} \psi_j(g_i - g_j) + \frac{\psi_k}{\phi_k}\left(\frac{1}{2} - \sum_{j=k+1}^{n} \phi_j(g_i - g_j)\right).$$

Since $x^*_{i-1} = x^*_i$, combining the last two equations with Inequality (8), we have

$$\begin{aligned} \sum_{j=1}^{i-1} \psi_j(g_i - g_j) &> \frac{1}{2} - \sum_{j=i+1}^{k} \psi_j(g_i - g_j) + \frac{\psi_k}{\phi_k}\left(\frac{1}{2} - \sum_{j=k+1}^{n} \phi_j(g_i - g_j)\right); \\ \frac{1}{2}\left(\frac{1}{\psi_k} + \frac{1}{\phi_k}\right) &< \frac{1}{\psi_k}\sum_{j=1}^{k} \psi_j(g_i - g_j) + \frac{1}{\phi_k}\sum_{j=k+1}^{n} \phi_j(g_i - g_j). \end{aligned}$$

This contradicts Condition (3). It can be proved in a similar way that if worker $i \in [k+1, \ldots, n]$ causes his predecessor to wait on the fixed point $\mathbf{x}^*$, then Condition (3) will be violated.

Now, suppose worker k causes his predecessor to wait on the fixed point $\mathbf{x}^*$. This implies that $x^*_{k-1} = x^*_k = 1/2$ and

$$\begin{aligned} g_k &> g^-_{k-1} + \frac{x^*_{k-1} - x^*_{k-2}}{v_{k-1}} + g^+_{k-1} + \frac{x^*_{k-1} - x^*_{k-2}}{w}; \\ x^*_{k-1} &< x^*_{k-2} + \psi_{k-1}(g_k - g_{k-1}). \end{aligned} \quad (13)$$

For other workers $j \neq k-1$ on the fixed point $\mathbf{x}^*$, we have

$$g_k = g^-_j + \frac{x^*_j - x^*_{j-1}}{v_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = 1, \ldots, k-2; \quad (14)$$

$$g_k = g^-_j + \frac{x^*_j - x^*_{j-1}}{u_j} + g^+_j + \frac{x^*_j - x^*_{j-1}}{w}, \quad j = k+1, \ldots, n. \quad (15)$$

Equations (14) lead to the following result:

$$x^*_{k-2} = \sum_{j=1}^{k-2} \psi_j(g_k - g_j).$$

Since $x^*_n = 1$, Equations (15) lead to the following result:

$$x^*_k = 1 - \sum_{j=k+1}^{n} \phi_j \left(g_k - g_j\right).$$

Since $x^*_{k-1} = x^*_k$, combining the last two equations with Inequality (13), we have

$$\begin{aligned} &\sum_{j=1}^{k-1} \psi_j(g_k - g_j) > 1 - \sum_{j=k+1}^{n} \phi_j \left(g_k - g_j\right); \\ &\sum_{j=1}^{k-1} \psi_j(g_k - g_j) + \sum_{j=k+1}^{n} \phi_j \left(g_k - g_j\right) > 1. \end{aligned}$$

This contradicts Condition (3). Thus, if Condition (3) holds, worker $i \in [2, n]$ cannot cause his predecessor to wait on the fixed point $\mathbf{x}^*$. It can be proved in a similar way that if Condition (3) holds, worker $i \in [1, n-1]$ cannot cause his *successor* to wait on the fixed point $\mathbf{x}^*$.

Thus, if Condition (3) holds there is no waiting on the fixed point $\mathbf{x}^*$ on which each worker i repeats a simple loop for each job assembled: He receives work from worker $i-1$ at point

x^*_{i-1}, which takes time g^-_i and relinquishes work for worker $i+1$ at point x^*_i, which takes time g^+_i. Recall that the loop of worker k overlaps with both intervals $[0, 1/2)$ and $[1/2, 1]$. If $k = 1$, then the fixed point $\mathbf{x}^*$ can be found by solving the following equations:

$$g^-_1 + \frac{1/2}{v_1} + \frac{x^*_1 - 1/2}{u_1} + g^+_1 + \frac{x^*_1}{w} \;=\; g^-_i + \frac{x^*_i - x^*_{i-1}}{u_i} + g^+_i + \frac{x^*_i - x^*_{i-1}}{w}, \quad i = 2, \ldots, n.$$

Otherwise, the fixed point $\mathbf{x}^*$ can be found by solving the following equations:

$$\begin{aligned}
g^-_1 + \frac{x^*_1}{v_1} + g^+_1 + \frac{x^*_1}{w} &= g^-_i + \frac{x^*_i - x^*_{i-1}}{v_i} + g^+_i + \frac{x^*_i - x^*_{i-1}}{w}, \quad i = 2, \ldots, k-1;\\
g^-_1 + \frac{x^*_1}{v_1} + g^+_1 + \frac{x^*_1}{w} &= g^-_k + \frac{1/2 - x^*_{k-1}}{v_k} + \frac{x^*_k - 1/2}{u_k} + g^+_k + \frac{x^*_k - x^*_{k-1}}{w};\\
g^-_1 + \frac{x^*_1}{v_1} + g^+_1 + \frac{x^*_1}{w} &= g^-_i + \frac{x^*_i - x^*_{i-1}}{u_i} + g^+_i + \frac{x^*_i - x^*_{i-1}}{w}, \quad i = k+1, \ldots, n.
\end{aligned}$$

Lemma 3 follows by simple algebra. □

B.5 Proof of Lemma 4

Proof. If $k = 1$, then worker 1 works in both intervals $[0, 1/2)$ and $[1/2, 1]$ on the fixed point. The system assembles a job every time worker 1 completes a simple loop. The average throughput is $\rho_s = \big(g^-_1 + (1/2)/v_1 + (x^*_1 - 1/2)/u_1 + g^+_1 + x^*_1/w\big)^{-1}$. If $k > 1$, then worker 1 works only in the interval $[0, 1/2)$. The average throughput is $\rho_s = \big(g^-_1 + x^*_1/v_1 + g^+_1 + x^*_1/w\big)^{-1}$. □

B.6 Proof of Theorem 2

Proof. The proof is similar to the proof of Theorem 2 in the Electronic Companion of Lim (2011). Iteration t follows the sequence of hand-off points from the end to the start of the line caused by the t-th reset. When the system operates sufficiently close to the fixed point $\mathbf{x}^*$ there is only one worker (we assume worker k) that crosses point 1/2 in each iteration. A hand-off occurs at point x^t_i where worker i, who is working forward, meets worker $i+1$, who is walking backward. Since the time spent by each worker from one iteration to the next is the same, we have

$$
\begin{aligned}
g_i^+ + \frac{x_i^t - x_{i-1}^t}{w} + g_i^- + \frac{x_i^{t+1} - x_{i-1}^t}{v_i} &= g_{i+1}^- + \frac{x_{i+1}^{t+1} - x_i^t}{v_{i+1}} + g_{i+1}^+ + \frac{x_{i+1}^{t+1} - x_i^{t+1}}{w}, \\
&\qquad i = 1, \ldots, k-2;
\end{aligned}
$$

$$
\begin{aligned}
g_{k-1}^+ + \frac{x_{k-1}^t - x_{k-2}^t}{w} + g_{k-1}^- + \frac{x_{k-1}^{t+1} - x_{k-2}^t}{v_{k-1}} &= g_k^- + \frac{1/2 - x_{k-1}^t}{v_k} + \frac{x_k^{t+1} - 1/2}{u_k} + \\
&\quad g_k^+ + \frac{x_k^{t+1} - x_{k-1}^{t+1}}{w};
\end{aligned}
$$

$$
\begin{aligned}
g_k^+ + \frac{x_k^t - x_{k-1}^t}{w} + g_k^- + \frac{1/2 - x_{k-1}^t}{v_k} + \frac{x_k^{t+1} - 1/2}{u_k} &= g_{k+1}^- + \frac{x_{k+1}^{t+1} - x_k^t}{u_{k+1}} + \\
&\quad g_{k+1}^+ + \frac{x_{k+1}^{t+1} - x_k^{t+1}}{w};
\end{aligned}
$$

$$
\begin{aligned}
g_i^+ + \frac{x_i^t - x_{i-1}^t}{w} + g_i^- + \frac{x_i^{t+1} - x_{i-1}^t}{u_i} &= g_{i+1}^- + \frac{x_{i+1}^{t+1} - x_i^t}{u_{i+1}} + g_{i+1}^+ + \frac{x_{i+1}^{t+1} - x_i^{t+1}}{w}, \\
&\qquad i = k+1, \ldots, n-1.
\end{aligned}
$$

Rewriting the above equations yields:

$$
\begin{aligned}
x_i^{t+1} &= (g_{i+1} - g_i)\psi_i + x_{i-1}^t - \frac{\psi_i}{\psi_{i+1}} x_i^t + \frac{\psi_i}{\psi_{i+1}} x_{i+1}^{t+1}, \quad i = 1, \ldots, k-2; \\
x_{k-1}^{t+1} &= (g_k - g_{k-1})\psi_{k-1} + x_{k-2}^t - \frac{\psi_{k-1}}{\psi_k} x_{k-1}^t + \frac{\psi_{k-1}}{\phi_k} x_k^{t+1} + \frac{\psi_{k-1}}{2}\left(\frac{1}{\psi_k} - \frac{1}{\phi_k}\right); \\
x_k^{t+1} &= (g_{k+1} - g_k)\phi_k + \frac{\phi_k}{\psi_k} x_{k-1}^t - \frac{\phi_k}{\phi_{k+1}} x_k^t + \frac{\phi_k}{\phi_{k+1}} x_{k+1}^{t+1} - \frac{\phi_k}{2}\left(\frac{1}{\psi_k} - \frac{1}{\phi_k}\right); \\
x_i^{t+1} &= (g_{i+1} - g_i)\phi_i + x_{i-1}^t - \frac{\phi_i}{\phi_{i+1}} x_i^t + \frac{\phi_i}{\phi_{i+1}} x_{i+1}^{t+1}, \quad i = k+1, \ldots, n-1.
\end{aligned}
$$

Since $x_n^t = 1$ for all t, we have

$$
x_{n-1}^{t+1} = (g_n - g_{n-1})\phi_{n-1} + x_{n-2}^t + \frac{\phi_{n-1}}{\phi_n}\left(1 - x_{n-1}^t\right).
$$

Substituting x_i^{t+1} into the equation for x_{i-1}^{t+1}, from $i = n-1$ to $i = 2$ yields:

$$
\begin{aligned}
x_i^{t+1} &= (g_n - g_i)\psi_i + x_{i-1}^t + \frac{\psi_i}{\phi_n}\left(1 - x_{n-1}^t\right), \quad i = 1, \ldots, k-1; \\
x_k^{t+1} &= (g_n - g_k)\phi_k + \frac{\phi_k}{\psi_k} x_{k-1}^t + \frac{\phi_k}{\phi_n}\left(1 - x_{n-1}^t\right) - \frac{\phi_k}{2}\left(\frac{1}{\psi_k} - \frac{1}{\phi_k}\right); \\
x_i^{t+1} &= (g_n - g_i)\phi_i + x_{i-1}^t + \frac{\phi_i}{\phi_n}\left(1 - x_{n-1}^t\right), \quad i = k+1, \ldots, n-1.
\end{aligned}
$$

It follows by simple algebra that for $i = 1, \dots, k-1$,

$$\frac{x_i^{t+1} - x_{i-1}^{t+1}}{\psi_i} - (g_n - g_i) = \left(\frac{\psi_{i-1}}{\psi_i}\right)\left[\frac{x_{i-1}^t - x_{i-2}^t}{\psi_{i-1}} - (g_n - g_{i-1})\right] + \left(1 - \frac{\psi_{i-1}}{\psi_i}\right)\frac{1 - x_{n-1}^t}{\phi_n}. \quad (16)$$

For $i = k$,

$$\frac{x_k^{t+1} - 1/2}{\phi_k} + \frac{1/2 - x_{k-1}^{t+1}}{\psi_k} - (g_n - g_k) = \left(\frac{\psi_{k-1}}{\psi_k}\right)\left[\frac{x_{k-1}^t - x_{k-2}^t}{\psi_{k-1}} - (g_n - g_{k-1})\right] + \left(1 - \frac{\psi_{k-1}}{\psi_k}\right)\frac{1 - x_{n-1}^t}{\phi_n}. \quad (17)$$

For $i = k+1$,

$$\frac{x_{k+1}^{t+1} - x_k^{t+1}}{\phi_{k+1}} - (g_n - g_{k+1}) = \left(\frac{\phi_k}{\phi_{k+1}}\right)\left[\frac{x_k^t - 1/2}{\phi_k} + \frac{1/2 - x_{k-1}^t}{\psi_k} - (g_n - g_k)\right] + \left(1 - \frac{\phi_k}{\phi_{k+1}}\right)\frac{1 - x_{n-1}^t}{\phi_n}. \quad (18)$$

Similarly, for $i = k+2, \dots, n$,

$$\frac{x_i^{t+1} - x_{i-1}^{t+1}}{\phi_i} - (g_n - g_i) = \left(\frac{\phi_{i-1}}{\phi_i}\right)\left[\frac{x_{i-1}^t - x_{i-2}^t}{\phi_{i-1}} - (g_n - g_{i-1})\right] + \left(1 - \frac{\phi_{i-1}}{\phi_i}\right)\frac{1 - x_{n-1}^t}{\phi_n}. \quad (19)$$

Let

$$\begin{aligned}
y_i^t &= \frac{x_i^t - x_{i-1}^t}{\psi_i} + g_i - g_n, \quad i = 1, \dots, k-1; \\
y_k^t &= \frac{x_k^t - 1/2}{\phi_k} + \frac{1/2 - x_{k-1}^t}{\psi_k} + g_k - g_n; \\
y_i^t &= \frac{x_i^t - x_{i-1}^t}{\phi_i} + g_i - g_n, \quad i = k+1, \dots, n.
\end{aligned}$$

Equations (16–19) become

$$\begin{aligned}
y_i^{t+1} &= \left(\frac{\psi_{i-1}}{\psi_i}\right) y_{i-1}^t + \left(1 - \frac{\psi_{i-1}}{\psi_i}\right) y_n^t, \quad i = 1, \dots, k; \\
y_i^{t+1} &= \left(\frac{\phi_{i-1}}{\phi_i}\right) y_{i-1}^t + \left(1 - \frac{\phi_{i-1}}{\phi_i}\right) y_n^t, \quad i = k+1, \dots, n.
\end{aligned}$$

These equations can be expressed as a linear system

$$\mathbf{y}^{t+1} = A\mathbf{y}^t,$$

where $\mathbf{y}^t = \left(y_1^t, y_2^t, \ldots, y_n^t\right)^T$ and A is a transition matrix of a finite state Markov chain. Since the Markov chain is irreducible and aperiodic (see Ross (1996)), $A^t \to A^*$ as $t \to \infty$. Thus, the orbit $\mathbf{y}^0, \mathbf{y}^1, \mathbf{y}^2, \ldots$ converges to the unique fixed point $\mathbf{y}^*$. It can be shown that the hand-off points converge to the fixed point $\mathbf{x}^*$ by simple algebra. □