4-2010

Self-Organizing Logistics Systems

John J. Bartholdi, III
Georgia Institute of Technology - Main Campus

Donald D. Eisenstein
University of Chicago

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

Published in Annual Reviews in Control, April 2010, Volume 34, Issue 1, pp 111–117.
http://dx.doi.org/10.1016/j.arcontrol.2010.02.006

Self-Organizing Logistics Systems

John J. Bartholdi, III * **Donald D. Eisenstein** ** **Yun Fong Lim** ***

* *Georgia Institute of Technology, Atlanta, GA 30332-0205 USA (e-mail: john.bartholdi@gatech.edu).*
** *University of Chicago, Chicago, IL 60637 (e-mail: don.eisenstein@chicagobooth.edu).*
*** *Singapore Management University, Singapore 178899 (e-mail: yflim@smu.edu.sg).*

Abstract: When a logistics system is "self-organizing" it can function without significant intervention by managers, engineers, or software control. The social insects, such as ants or bees, provide models of self-organizing logistics systems that may be profitably emulated. We illustrate some of these ideas for the problem of balancing assembly lines.

Keywords: Agile manufacturing, manufacturing systems, adaptive control, dynamic systems

1. INTRODUCTION

The term "logistics" in the title has been chosen to suggest a broad view of manufacturing, in which the final product is the delivery of the item to the customer, and the entire supply chain is the assembly line that manufactures this product.

"Self-organization" is the informal name given to the phenomenon wherein large-scale structure arises with seeming spontaneity from the myriad interactions of local agents. We are all familiar with an example that employs thousands of workers, functions at near optimality, and yet employs no management, no engineers, no consultants, and no IT department: The social insects, such as ants or bees. To use ants as an example, no ant is in charge: The queen is merely an egg-laying machine, and there are no castes devoted to planning or control. Instead each ant follows some simple instincts or urges — they must be simple because an ant's brain has very few neurons — but the aggregate result, such as a highly-structured nest or a seemingly high-organized pattern of food retrieval, seems to transcend the planning ability of any individual.

2. A MODEL FOR SELF-ORGANIZING LOGISTICS

Self-organization is evident in many of the activities of social insects that relate to logistics, such as nest-building, foraging, food-retrieval, and storage. More generally, ants must solve the problem of task allocation: What ants should be assigned to what task? There are many tasks to be performed within an ant colony, including nest building, nest defense, foraging, food processing (in some species), food storage, queen care, brood care, and so on. It is critical that ants get the balance right. Consider, for example, the so-called "honeypot ant", the main food of which is nectar. Some ants of each colony devote themselves to food storage by serving as living storage vessels. If too few ants are committed to food storage, then there is a mismatch between foraging and storage and the survival of the colony is at risk. Conversely, if too many ants are devoted to storage, then there is a misallocation of resources.

Biologists have studied such questions in detail and found that the social insects solve problems of resource allocation with near optimality (an excellent description of these issues may be found in Wilson and Hölldobler (2009)). In this case, by "solve", we mean that a nearly optimal solution emerges from the interactions of many individuals. The successes of self-organization for ants is most visible in the collection of activities that we might term *logistics* because they have this in common with the human activities of the same name: They are concerned with the processing, transport, and storage of goods deemed valuable to the society (Anderson and J. J. Bartholdi (2000)).

Self-organization is such an appealing way of solving human logistics problems that it is worth a moment to itemize the advantages one might expect.

Ease of implementation: Typically one need only establish the process, without much attention to precision. The system will subsequently fine-tune itself as it self-organizes.

Adaptivity: For any logistics system, the main challenge is to adapt to changes in the environment, such as interrupted schedules, machine failures, unforeseen surges in demand, and so on. If a system is self-organizing, then it automatically reacts to events. There is no need for an external control system, such as management, to monitor the system and intervene. The control system is inherent in the operation and does not reside in any particular individuals.

Minimal data requirements: Self-organization typically occurs when information is embedded in the timing and location of interactions among agents. In a sense the data is "read" directly by the society of interacting agents and so does not need to be collected or maintained.

(NB: It is worth remarking here that self-organization can be to both good and bad ends. We shall have more to say about the latter in Section 4.)

3. "BUCKET BRIGADE" ASSEMBLY LINES

One goal of this paper is to convince the reader of the practical value of the principle of self-organization and to this end will concentrate on the phenomena of self-organization with which the author is most familiar: that which has been given the name *bucket-brigades.* This is a variant of a traditional assembly line in that it requires workers to pass work sequentially from one worker to another; but unlike a traditional assembly line, there is no fixed assignment of workers to stations. Instead, workers, numbered $1, \ldots, n$, each follow a simple local rule that determines what to do next: *Carry your work forward from station to station until you have finished assembly or else another worker has taken it, in which case walk back upstream and take over the work of the first worker of smaller index.*

Note that workers are not restricted to any subset of stations; rather each one carries his work as far toward completion as possible and then walks back to get more.

All items are identical and so each requires the same total processing time according to some work standard, which we normalize to one "time unit".

The essential model rests on these two assumptions:

Assumption 1. (Characterization Of Workers By Velocity). Each worker i can be characterized by a work velocity v_i with which he proceeds along the direction of material flow, and a velocity w_i at which he walks back to get more work.

Assumption 2. (Smoothness And Predictability Of Work). The work-content of the product is spread continuously and uniformly along the flow line (the length of which we normalize to 1).

The Characterization Of Workers By Velocity is likely to hold in an mass-production environment, where work has been "de-skilled" so that velocity is based on a single dimension, such as motivation or eye-hand coördination. (This point is more fully documented in Bartholdi and Eisenstein (1996b)).

There is clearly some license in the assumption of Smoothness And Predictability Of Work; nevertheless, this assumption is reasonable in many instances, detailed by us elsewhere Bartholdi and Eisenstein (1996b). This is the goal towards which management and engineering strive, so as to enable smooth and continuous flow of work along the assembly line by removing variance from work and eliminating bottlenecks.

When the total work-content of a product greatly exceeds the total time for workers to hand off their work and walk back to get more, then we may take $w_i = \infty$ and the bucket brigade assembly line becomes self-organizing in a very useful way, as described by Bartholdi and Eisenstein (1996b); Bartholdi et al. (1999). Their main results, slightly simplified, are as follows.

- There exists a unique, balanced partition of the effort wherein worker i performs the interval of work:
$$\text{from } \frac{\sum_{j=1}^{i-1} v_j}{\sum_{j=1}^{n} v_j} \text{ to } \frac{\sum_{j=1}^{i} v_j}{\sum_{j=1}^{n} v_j}, \tag{1}$$
so that each worker invests the same clock time in each item produced and both product starts and completions occur metronomically.
- If the workers are sequenced from slowest to fastest ($v_1 < \cdots < v_n$) then, during the normal operation of the line, work is spontaneously and constantly reallocated to reach this balance. Furthermore, even if no passing is allowed, the production rate converges to
$$\sum_{i=1}^{n} v_i \text{ items per unit time,} \tag{2}$$
which is the maximum possible for the given set of workers.
- If the workers are *not* sequenced from slowest to fastest, then the line will "sputter": that is, it will produce erratically. If, in addition, faster workers are not allowed to pass slower workers, then the production rate of the assembly line can be suboptimal rate and the line can behave in counterintuitive ways, such as production rate *decreasing* when a worker increases his velocity.

Details may be found in Bartholdi and Eisenstein (1996b) and a survey of related results in Bratcu and Dolgui (2005) (of which this paper may be seen as an update).

Figure 1 shows an example of how the movement of the workers stabilizes, with the faster workers eventually allocated more work. This figure was generated by a simulation of three workers of velocities $v_i = 1, 2, 3$, respectively.

This self-balancing seems to be very robust in the sense that many natural generalizations of the model lead to essentially the same conclusion: When properly configured — in this case, when the workers are indexed so that $v_i < v_{i+1}$ — then the assembly line will balance itself and will, moreover, resist perturbations. This has been proven to hold when, for example, the work content at each station is random, as in Bartholdi et al. (2001), or when the instantaneous speeds of the workers may vary, as in Bartholdi and Eisenstein (1996b), or when there is worker learning and consequent speed-up, as in Armbruster et al. (2007). The simplicity and persistence of this behavior makes it very useful in practice, for it is easy to implement and works under a variety of conditions, many documented in Bartholdi and Eisenstein (1996a).

3.1 Uses

Manufacturing Bartholdi and Eisenstein (1996a) and Bratcu and Dolgui (2005) describe a selection of practical applications in manufacturing and manufacturing-like environments. One of the main advantages of the self-organizing ability of bucket brigades is that it makes no requirements for data. Traditional approaches to balancing assembly lines require that one define task elements and estimate the work-content of each. This is unnecessary for

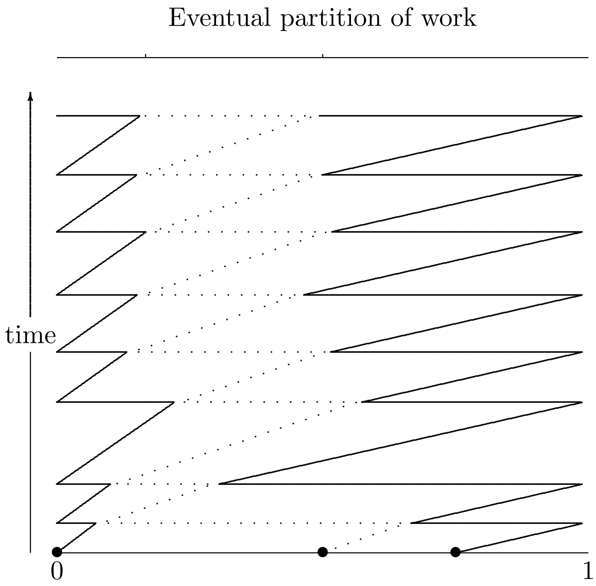

Fig. 1. A time-expanded view of a bucket brigade production line with three workers sequenced from slowest to fastest. The solid horizontal line represents the total work content of the product and the solid circles represent the initial positions of the workers. The zigzag vertical lines show how these positions change over time and the rightmost spikes correspond to completed items.

bucket brigades, and so they have proven attractive when the enterprise cannot wait to complete task definitions (see Bartholdi and Eisenstein (2005) for example). Another reason to use bucket brigades is that the work is spontaneously reassigned even as workers learn and increase their proficiency (in our model, work velocities v_i). Case studies appear in Villalobos et al. (1999), Munoz and Villalobos (2002), and Bartholdi and Eisenstein (2005). Armbruster et al. (2007) provides a particularly interesting study of worker learning during bucket brigades and the effects on the dynamics of the system.

Production Imagine Figure 1 turned on its side and the series of peaks may remind the reader of a graph of inventory level over time when subjected to constant demand. This was noticed by Eisenstein (2005), who studied a facility that follows a cyclic schedule to replenish the inventory of a set of items through production by a shared machine. Eisenstein's concern was how to reëstablish an intended cyclic schedule after disruption. He embedded bucket-brigade-like logic into standard produce-up-to policies so that they become self-organizing; that is, the scheduling policy, which says what product to produce next and in what quantity, recovers a target cyclic schedule after disruption. Eisenstein's scheduling policy is controlled by a single parameter that is an analogue of the work velocity of the fastest worker in a bucket brigade; through manipulation of this parameter, his policy can tune recovery to be aggressive, with frequent setups and small batches, or methodical, with fewer setups and larger batches.

Distribution In practice bucket brigades have been most successful as a way of organizing order-pickers in a distribution center (DC). In this case the product being "assembled" is the customer order, which will have been assigned to a box that an order-picker will carry along an aisle of shelves, picking the particular items requested. A typical high-volume DC in the US might have hundreds of order-pickers. Furthermore, there will be a considerable range of work-velocities because many workers might be temporaries, hired in advance of the busy retail shopping season.

Most DCs use work-content models to define worker "zones", which are effectively work stations. A worker will pick all the items within his zone into a box and then pass the box to the subsequent zone. But the effort to allocate work evenly is never and can never be successful in this case. First, the customer orders are all different and so the assembly line is, in effect, a mixed-model assembly line, which is notoriously hard to balance. Furthermore, the balance is based on time-motion studies, which are statistical aggregations suggesting the time expected of a mythical "standard worker", which may rarely be realized by actual humans in the DC. Finally, the static assignment of work to order-pickers is fundamentally unable to adapt to disruptions, which may be considered normal part of the working day. Instead, it is typical that a level of management be devoted to monitoring the workers and reassigning work as necessary to correct spot imbalances. Bartholdi and Eisenstein (1996a) report adoption of bucket brigades within the DCs of a range of significant companies in the US, with most reporting increases in productivity of 20–40%.

4. UNDESIRABLE SELF-ORGANIZATION

Sometimes the end state of a self-organization is not what one could wish. For example, to return to the assembly line: We interviewed the operations manager of a company with thousands of small restaurants that prepared sandwiches to order. They saw their problem as one of running a mixed-model assembly line in which the successive products on the line might differ according to the request of the customer. In this case the last station on the line was the cashier and the problem was how to make use of this worker when the queue of customers was at the other end of the assembly line. The company experimented with having the workers circle back from the cashier's station to the beginning of the line, but with the predictable consequence that soon the assembly line operated at the speed of the slowest worker. In other words, the system organized itself into the least productive configuration possible.

The problem in this case was that the system was set up in a way that seems, after a moment's consideration, obviously flawed. A more dramatic example of unfortunate self-organization may be found among the army ants. These ants, which are nearly blind, follow the scent of their nest mates to pour out from the nest along foraging trails. Occasionally a disruption will leave a subgroup cut off from the main river of ants, and this subgroup may reform into a closed loop, a "circular mill", along which they continue to follow their predecessors until they die of exhaustion and starvation (see Couzin and Franks (2003) and references therein). This is a graphic reminder that one must understand *all* the possible modes of behavior of a system before relying on it.

Consider, for example, a bucket brigade in which each worker i has a forward velocity v_i and a backward velocity w_i that need not be infinite. This bucket brigade scheme is descriptive of order-picking in a high-service, low-volume distribution center such as those dispensing service parts. In such an environment, there may be a relatively large amount of travel for few picks, in which case the times to work forward is comparable to the time to walk back. In such a case the behavior of bucket brigades — if they are set up incorrectly — can be surprising.

First note that more complicated patterns of movement are possible

Passing: in which two workers walk past each other. This can happen only if the worker who is walking upstream has smaller index than the worker who is moving downstream (otherwise the bucket brigade rules would call for a hand off of work).

Overtaking: in which a faster worker overtakes a slower worker moving in the same direction.

(It should be remarked that some applications, especially in DCs, overtaking may not possible due to material-handling considerations. Bucket brigades can still be self-balancing in these cases.)

If overtaking is allowed then In the long run each worker must travel as far forward as he does backward and so worker i has a net production rate of $(1/v_i + 1/w_i)^{-1}$ and the long-run average production rate of the n workers is the sum of their net production rates, and this is independent of their starting positions.

4.1 Stability

In this section we omit the proofs because they are generally extensions of those in previous papers, such as Bartholdi and Eisenstein (1996b); or else the arguments are too long and may be found in Lim (2005) or Bartholdi et al. (2009).

It seems natural to guess that convergence to a stable allocation of work must require indexing the workers by their net production rates. Surprisingly, this is not so.

Theorem 1. A bucket brigade assembly line will spontaneously balance itself if the workers are indexed so that

$$\frac{1}{v_1} - \frac{1}{w_1} > \cdots > \frac{1}{v_n} - \frac{1}{w_n}; \tag{3}$$

or, in other words, from most-slowed to least-slowed.

(See Bartholdi et al. (2009).)

In the condition of Theorem 1 the term $1/v_i - 1/w_i$ represents the difference in the encumbered and the unencumbered transit times of worker i and so gives the extent to which he is slowed by work. Therefore the workers should be indexed according to the extent to which each is slowed by work.

Note that this may require a worker who is *slower* in both directions to be the one who sets the pace for the bucket brigade. For example, a worker of forward velocity 9 and backward velocity 20 should be given a higher index than one of forward velocity 10 and backward velocity 40.

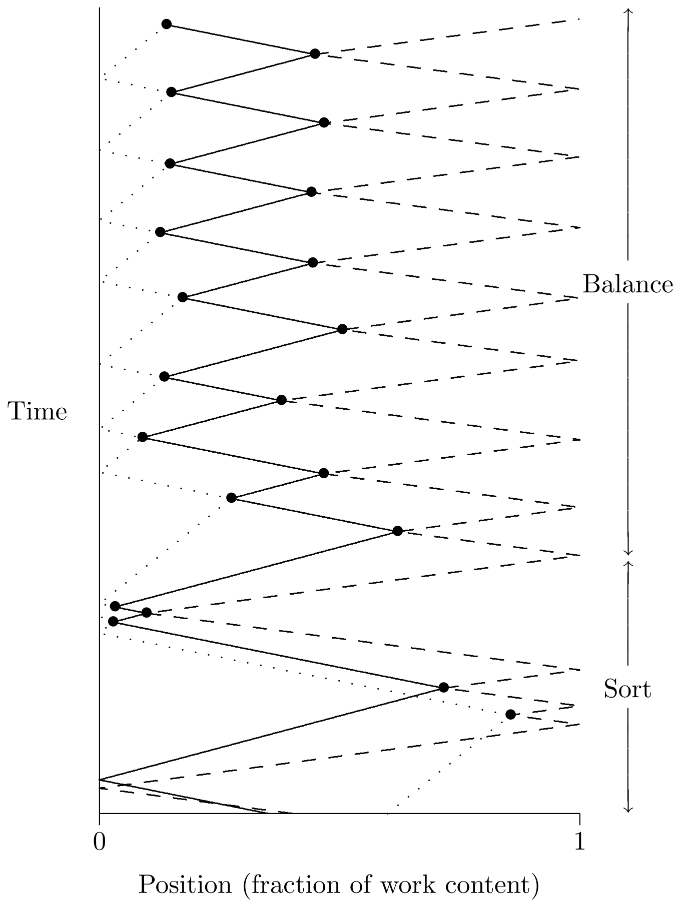

Fig. 2. Movement of three workers in a convergent bucket brigade. During an initial transient period workers sort themselves by index, after which all passing and overtaking cease. Eventually workers repeat the same intervals of work-content on successive products.

When the workers are indexed from most-slowed to least-slowed, then independently of the initial positions and directions of movement of the workers, behavior develops according to this pattern: After a transient period in which the workers spontaneously sort themselves by index, workers move so that work is reallocated to approach perfect balance. Such behavior may be seen in Figure 2.

Under bucket brigades, workers share work-content by handing off items to successors. The locations at which hand-offs occur determine how the work is shared. The bucket brigade assembly line is *balanced* if each worker invests the same clock time and repeats the same interval of work content for each item produced, and, moreover, those intervals are non-overlapping. Let the balance point at which worker i hands off work, given as a fraction of work-content completed, be x_i^*.

The proofs of the next two theorems appear in Bartholdi et al. (2009).

Theorem 2. For any bucket brigade the point

$$x_i^* = \frac{\sum_{j=1}^{i} (1/v_j + 1/w_j)^{-1}}{\sum_{j=1}^{n} (1/v_j + 1/w_j)^{-1}} \quad \text{for i=1,...,n-1.} \tag{4}$$

is a fixed point and is, moreover, the unique point of balance.

Theorem 2 establishes the existence of balance, but for this to be useful in practice, the assembly line must spontaneously seek balance.

Theorem 3. If workers are sequenced on the assembly line from most-slowed to least-slowed then $x^* = (x_1^*, \ldots, x_n^*)$ is an attractor.

It is important to note that when $v_i \ll w_i$ then the condition for self-balancing, that $\frac{1}{v_i} - \frac{1}{w_i} > \frac{1}{v_{i+1}} - \frac{1}{w_{i+1}}$, reduces to $v_i < v_{i+1}$. In other words, it is sufficient to index the workers from slowest to fastest. This is the case in all implementations of which we are aware. If encumbered velocity is much less than unencumbered velocity then it is sufficient to index the workers by their encumbered velocities v_i.

4.2 Chaotic behavior

When the condition for self-balancing fails to hold, then a bucket brigade is capable of chaotic behavior. Bartholdi et al. (2009) establish this by showing a particular bucket brigade that emulates a system that is well-known to be chaotic.

Consider the bucket brigade composed of workers with the following velocities: $v_1 = 1$, $w_1 = 1/3$; $v_2 = 1$, $w_2 = 1$. This bucket brigade fails to satisfy the condition of Theorem 1 and it is straightforward to verify that the dynamics function relating the positions of successive hand-offs is given by the following, where x^k denotes the location of the k-th hand-off from worker 1 to worker 2.

$$x^{k+1} = 1 - \left(2x^k \bmod 1\right). \qquad (5)$$

This is an *expanding* map; that is, it has slope of absolute value strictly greater than 1, where defined (it has discontinuities at 1/2 and 1). The point 1/3 is the unique point of balance, but it is a *repelling* fixed point, which means that the system spontaneously avoids balance. The point 2/3 is another repelling fixed point.

This dynamics function is a reflection of the *Bernoulli map*

$$x^{k+1} = 2x^k \bmod 1.$$

While there are several alternative definitions of chaos, all fairly technical, all agree that the Bernoulli map is chaotic (see, for example, Martelli (1999), Devaney (1989)). The reflected Bernoulli map (5) is also chaotic, as may be seen by considering the values of the x^k to be represented by their binary expansions. Then each iteration of either map simply shifts digits leftward one position and drops the integer part. The reflected Bernoulli map (5) then complements each bit. A consequence is that the two-fold composition of the Bernoulli map is identical to the two-fold composition of the reflected Bernoulli map (except at $0, 1/4, 1/2, 3/4$, and 1).

Figure 3 shows the transition of a bucket brigade from convergence to chaos as the velocity of one worker changes. In this example, the bucket brigade initially satisfies the condition of Theorem 1, with workers of velocities $v_1 = 0.1$, $w_1 = 1$, $v_2 = 3$, and $w_2 = 2$. Initially, worker 1 is unusually slow in the forward direction. Paradoxically, as worker 1 gains experience so that v_1 increases, then the bucket brigade changes from convergent to chaotic.

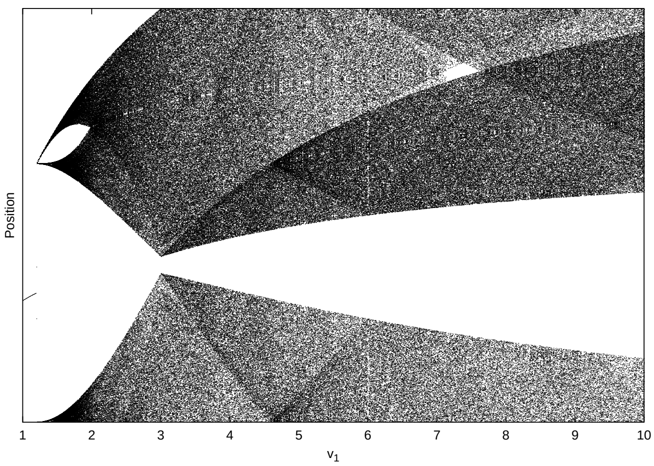

Fig. 3. Transition to chaos as v_1 increases. This bifurcation diagram plots the long-run locations of hand-offs as a function of the velocity v_1. As v_1 increases, the attractor changes from a fixed point to what appears to be a Cantor set.

We constructed this graph by stepping through values of $v_1 \in [0.1, 10]$, computing the positions of hand-offs through 10,000 iterations (presumably long enough for transients to fade away), and then plotted the positions of the next 1,000 hand-offs. For $v_1 < 6/5$ the self-balancing condition holds, and as expected, all hand-offs occurred at a fixed point, the value of which increases with v_1 as predicted by Theorem 2. At the threshold of chaos, $v_1 = 6/5$, the self-balancing condition fails to hold, and the formerly attracting fixed point appears to become explosively repelling. Here behavior appears to be nearly periodic; but on closer examination, each thin branch may be seen to be composed of two still thinner branches, and so on through ever finer levels of detail. In this region of chaos, the asymptotic sets corresponding to each value of v_1 appear Cantor-like (Alligood et al. (1996); Devaney (1992); Martelli (1999)). Another regime of behavior occurs as $v_1 > 3$; Lim (2005) explains much of the fine structure, including “gaps”, “shadows”, and “threads”.

4.3 Implications of chaos

There has been some confusion about chaos in the manufacturing literature, as researchers have used the word informally, claiming “chaos” when they observed patterns of behavior too complicated to comprehend. This has so plagued the literature that several papers have been written just to debunk such claims (example: Schmitz et al. (2002)). Chase et al. (1993) formally established chaotic behavior in a model of continuous manufacturing; ours is, as far as we know, the first example of deterministic chaos in a model of discrete manufacturing.

The most notable external implication of chaotic hand-offs is that the intercompletion times of products will appear to be random. This is worth emphasizing: *The chaotic assembly line will appear to complete products at random even though it is fully deterministic.* This is illustrated in Figure 4, which plots the cumulative distribution of intercompletion times of the bucket brigade that corresponds to the reflected Bernoulli shift. One-third

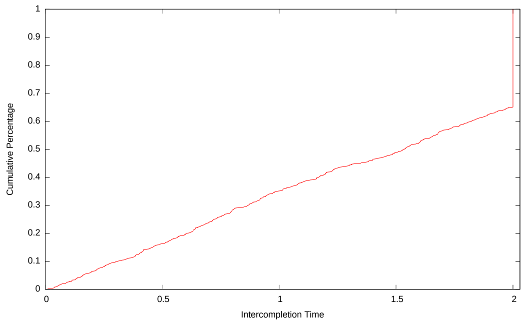

Fig. 4. Cumulative plot of intercompletion times. With probability $1/3$ the intercompletion time assumes value 2; with probability $2/3$ the intercompletion time is uniformly distributed in the interval $(0, 2)$. To avoid spurious results, the simulation was based on the java.math.BigDecimal class of the programming language Java, with precision of 10,000 decimal digits (more than 33,000 bits), and was terminated before exceeding the precision.

of the time the intercompletion times will assume the value 2, and two-thirds of the time intercompletion times will be uniformly distributed between 0 and 2. Figure 4 was generated by simulation, but its correctness is confirmed by the following informal argument: Some intercompletion times will have value 2. These occur when worker 2 takes work from worker 1 somewhere in $[1/2, 1]$, completes the item, overtakes worker 1, starts a new item, passes worker 1, and completes another item. The remaining intercompletion times arise when worker 2 takes over work from worker 1 at some position in $(0, 1/2)$. The average intercompletion time with value less than 2 must be 1 because hand-offs are uniformly distributed (the natural distribution of the reflected Bernoulli map). The long run average production rate of this bucket brigade is $3/4$ and so the average intercompletion time is $4/3$. If fraction y of intercompletion times assume value 2, the remaining $(1 - y)$ assume values averaging 1. Therefore it must be that $2y + 1\,(1 - y) = 4/3$, from which it follows that the intercompletion times of value 2 comprise one-third of the total and the others, which are uniformly distributed in $[0, 2)$ comprise the remaining two-thirds.

A downstream observer of this assembly line might recognize a shadow of a pattern (intercompletion times of value 2); but this pattern would be interrupted frequently and unpredictably. Erratic completions would interfere with subsequent downstream processes, such as further assembly, checking, packing, or shipping.

Product starts would be similarly erratic and so consumption of parts would also appear significantly random, which would, in turn, undermine just-in-time production and would inflate requirements for safety stock. The apparent randomness of starts may be seen by observing that every bucket brigade assembly line has a sort of *dual* "disassembly" line that is moving in the opposite direction. In this interpretation, the work content to assemble an item is identical to the work content to disassemble it. Imagine that, when worker i completes an item, he immediately begins disassembling it at rate w_i. Similarly when a worker i completes disassembly of an item, he immediately begins to re-assemble it at rate v_i. When worker $i < j$ working forward meets worker j working back, worker i exchanges his item being assembled for the item being disassembled by worker j. At all times there are n items in process—some being assembled and some disassembled.

If the i-th worker in the bucket brigade has forward velocity v_i and backward velocity w_i then in the dual bucket brigade the i-th worker has forward velocity w_{n-i+1} and backward velocity v_{n-i+1}. This change in perspective is useful because the bucket brigade and its dual are equivalent in some important ways. For example, any position x in one bucket brigade corresponds to the position $1 - x$ in the other. More importantly, if one is balanced then the other is as well. (Indeed, the condition of Theorem 1 is invariant under this transformation.) Similarly, if one is chaotic then the other is too.

5. BACK TO THE SOCIAL INSECTS

Reyes and Fernández-Haegar (1999) report that the ant species *Messor barbarus* employs bucket brigades in returning seeds to the nest: Slower ants pick up seeds from the ground and run back along the foraging trail until encountering a faster any, which takes the seed and continues toward the nest. Reyes and Fernández-Haegar (1999) report bucket brigades of three to six ants successively passing off a seed, always to a faster ant.

This phenomenon is analyzed in Anderson et al. (2002), which suggests how bucket brigades arise among *Messor barbarus* and what benefits they might derive. The key insight is that the larger the ant, the less slowed she is by carrying a seed. Now this fact is sufficient to generate a bucket brigade if each ant operates according to a simple impulse: Run out along the foraging trail and take the first seed you can, even if it must be wrested from another ant.

Consider the largest ant in the colony. As soon as it encounters an ant bringing back a seed, it can take that seed — because it is the biggest ant! It carries the seed back to the nest and then goes back out along the foraging path, where it can be expected to repeat this process. Consequently the largest (and least slowed) ant will work at the end of the foraging path (assembly line). On the other hand, the smallest ant is unable to take a seed from any other ant and must travel to the end of the foraging path to pick up a seed from the ground. As it travels back to the nest, it must relinquish its seed to the first ant it encounters. It then returns to the end of the foraging trail to pick up another seed. Thus the smallest (most-slowed) ant ends up working at the start of the assembly line. The bucket brigade protocol has emerged from two simple facts about ants. Thereafter, the ant bucket brigade is configured to satisfy the condition of Theorem 1, which tends to produce a smooth, regular flow of seeds into the nest.

This has several obvious advantages to the colony of ants. First, by smoothing the flow of seeds into the nest, there is reduced chance of congestion, exactly as in a human supply chain. Moreover, there is a certain economic rationality in the flow of seeds. Each seed may be imagined to increase in value as it gets closer to the nest, where its caloric contribution is closer to being realized. And as the seed increases in value, it accelerates towards the nest and is, moreover, carried by ever larger ants. This provides increased protection from other insects who might rob an ant of its seed or predate the ant itself.

6. CONCLUSIONS

Bucket brigades are an example of an idea from the social insects that translates naturally to human supply chain activities, and with notable beneficial results (see, for example, those reported in Bartholdi and Eisenstein (1996a)). Biologists are very actively exploring the logistics systems of social insects and there are sure to be additional ideas to emulate. Wilson and Hölldobler (2009) is an especially useful reference in this regard.

However useful self-organization can be, care must be taken to avoid forms of self-organization that can be unexpected, undesirable, or even disastrous. For example, as we have seen, ill-conditioned bucket brigades are capable of chaotic behavior, and this should sound a cautionary note for the management of manufacturing systems. A central goal of manufacturing systems control is the reduction of variability, such as results from machine breakdowns, vagaries in the positioning of work and in task execution, human inconsistency, and so on. But for a chaotic assembly line, even if every traditional source of variability has somehow been eliminated so that the system is purely deterministic, the product starts and completions can nevertheless appear irreducibly random. Such apparent randomness is inherent in the system and is resistant to the traditional tools of manufacturing control.

The possibility of chaotic behavior also has implications for study of the manufacturing systems. Most immediately, one must be extremely careful in simulating a system that may be chaotic. The chaotic bucket brigade we devised provides a vivid example, for almost all starting points of the workers lead to chaotic behavior; yet none of these would be seen in a simulation on a finite precision machine. In our chaotic bucket brigade, any simulation on a finite precision machine must always result in periodic behavior, so the chaos is effectively hidden from simulation, a traditional way of searching for problems in advance of building a real system.

ACKNOWLEDGEMENTS

We gratefully acknowledge support of: the Office of Naval Research through grant #N000140710228 (Bartholdi); the H. Milton Stewart School of Industrial Engineering at the Georgia Institute of Technology (Bartholdi); the Booth School of Business at the University of Chicago (Eisenstein); and the Lee Kong Chian School of Business, Singapore Management University (Lim).

REFERENCES

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

Anderson, C., Boomsma, J.J., and Bartholdi, III, J.J. (2002). Task partitioning in insect societies: bucket brigades. *Insectes Sociaux*, 49(2), 171–180.

Anderson, C. and J. J. Bartholdi, I. (2000). Centalized versus decentralized control in manufacturing: Lessons from social insects. In I.P. McCarthy and T. Rakotobe-Joel (eds.), *Complexity and Complex Systems in Industry*, 92–105. The University of Warwick.

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

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

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

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

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

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

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

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

Chase, C.J., Serrano, J., and Ramadge, P.J. (1993). Periodicity and chaos from switched flow systems: contrasting examples of discretely controlled continuous systems. *IEEE Transactions on Automatic Control*, 38(1), 70–83.

Couzin, I.D. and Franks, N.R. (2003). Self-organized lane formation and optimized traffic flow in army ants. *Proceedings of the Royal Society of London B*, 270, 139–146.

Devaney, R.L. (1989). *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley Publishing Company, 2nd edition. ISBN 0-8133-4085-3.

Devaney, R.L. (1992). *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. Addison-Wesley Publishing Company. ISBN 0-201-55406-2.

Eisenstein, D.D. (2005). Recovering cyclic schedules using dynamics produce-up-to policies. *Oper. Res.*, 53(4), 675–688.

Lim, Y.F. (2005). *Some generalizations of bucket brigade assembly lines*. Ph.D. thesis, Georgia Institute of Technology, School of Industrial and Systems Engineering.

Martelli, M. (1999). *Introduction to discrete dynamical systems and chaos*. John Wiley & Sons. ISBN 0-471-31975-9.

Munoz, L.F. and Villalobos, J.R. (2002). Work allocation strategies for serial assembly lines under high labor turnover. *Int. J. Prod. Res.*, 40(8), 1835–1852.

Reyes, J.L. and Fernández-Haegar, F. (1999). Sequential co-operative load transport in the seed-harvesting ant *Messor barbarus*. *Insectes Sociaux*, 46, 199–125.

Schmitz, J.P.M., van Beek, D.A., and Rooda, J.E. (2002). Chaos in discrete production systems? *Journal of Manufacturing Systems*, 21(3), 236–246.

Villalobos, J.R., Munoz, L.F., and Mar, L. (1999). Assembly line designs that reduce the impact of personnel turnover. In *Proceedings of IIE Solutions Conference, Phoenix, AZ*.

Wilson, E.O. and Hölldobler, B. (2009). *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*. W. W. Norton & Co.