BUCKET BRIGADES
A self-organizing scheme for sharing work

John J. Bartholdi, III
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205 USA
john.bartholdi@isye.gatech.edu *

Donald D. Eisenstein
Graduate School of Business
The University of Chicago
Chicago, Illinois 60637 USA
don.eisenstein@gsb.uchicago.edu †

Abstract

Self-organizing systems do not require a centralized authority to manage them. Instead, they achieve global coördination spontaneously through the interaction of many simple components.

When workers are organized into "bucket brigades" they can function as a self-organizing system that spontaneously achieves its own optimum configuration without conscious intention of the workers, without guidance from management, without any model of work content, indeed without any data at all. The system in effect acts as its own computer. We give examples in manufacturing and in warehousing.

Introduction

A self-organizing system is one in which global organization spontaneously evolves from myriad local interactions of the pieces. Here is an example: Consider a hive of honeybees. Each day they face a logistics problem of how to coördinate their efforts to harvest nectar. The measure of success is a social one: the good of the colony. But bees have no blueprint, no mechanism of central planning. Instead, each bee follows a simple "algorithm" that determines what she does next; and when many bees follow the same algorithm, an allocation of foragers evolves that is close to the best imaginable. In effect the colony acts as a computer that finds the (nearly) optimal allocation of effort (Bartholdi, Seeley, Tovey, and Vande Vate 1993).

Among the advantages of this self-organization are that:

- It requires no central planning or higher organizational entity. There is no management function because each entity simply follows a local rule.
- It is adaptive: It will spontaneously reallocate effort in response to changes in the environment.

* Supported by the Air Force Office of Scientific Research through grant #F49620-94-1-0232 and the Office of Naval Research through grant #N00014-89-J-1571.

† Supported by the Graduate School of Business at the University of Chicago.

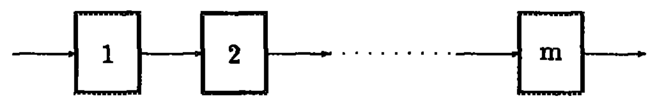

Figure 1: A simple flow line in which each item requires processing on the same sequence of work stations.

Exploring these simple ideas has led to some practical applications within management science/industrial engineering. Here are two: one in manufacturing and another in warehousing.

Bucket brigade manufacturing

Bucket brigade manufacturing is a way of organizing workers on a flow line, in which there are fewer workers than stations and each worker carries her work from station to station. When the last worker completes an item, she returns to take over the item of her predecessor, who relinquishes it and returns to take over the work of *her* predecessor, and so on until the first worker starts a new item on the line.

A model

Call each instance of the product an *item* and consider a flow line in which each of a set of items requires processing on the same sequence of m work stations, as in Figure 1. A station can process at most one item at a time, and exactly one worker is required to accomplish the processing. 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". Let the processing requirement at station j be p_j, a fixed percentage of the total standard work content of the product.

Note that, because there are no buffers and no passing is allowed, a worker can be *blocked* if she is ready to work at a station, but that station is occupied by another worker; then she must wait for the station to become available.

The time required to walk back and take over work is small relative to the total work content, and so we assume that when the last worker finishes an item, then—*at the same instant*—worker n takes over from worker $n-1$, who takes over from worker $n-2, \ldots,$ who takes over from worker 1, who introduces a new item into the system. We say that the line *resets* at such an instant. This simplification frees us from worry about the details of the continuous-time evolution of the bucket brigade line; instead we can restrict our attention to the positions of the workers at those instants immediately after the line resets.

We model each worker i by a constant work velocity v_i, which represents her performance as a fraction of work standard. This is reasonable in lines for which most of the tasks are variations on a single skill, such as sewing in the apparel industry.

Bartholdi and Eisenstein (1996a) have described the behavior of bucket brigade production lines. Their main results, slightly simplified, are that:

- There is a unique balanced partition of the effort wherein worker i performs the interval of work:

$$\text{from } \frac{\sum_{j=1}^{i-1} v_j}{\sum_{j=1}^{n} v_j} \text{ to } \frac{\sum_{j=1}^{i} v_j}{\sum_{j=1}^{n} v_j}, \qquad (1)$$

  so that each worker invests the same clock time in each item produced.

- If the workers are sequenced from slowest to fastest then, during the normal operation of the line, work is spontaneously and constantly reallocated to reach this balance and the production rate converges to

$$\sum_{i=1}^{n} v_i \text{ items per unit time,}$$

  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 and at suboptimal rate. Furthermore, the line can behave in counterintuitive ways, such as production rate *decreasing* when a worker increases his velocity.

Figure 2 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 $\mathbf{v} = (1, 2, 3)$.

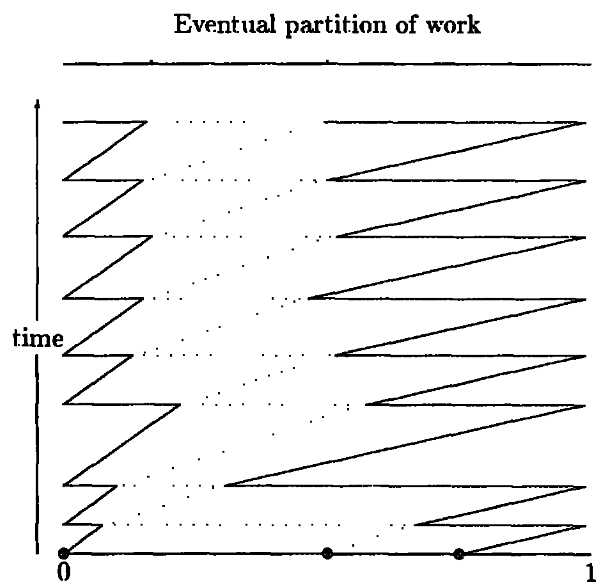

Figure 2: A time-expanded view of a bucket brigade production line with three workers sequenced from slowest to fastest. The solid horizontal line represents the total work content of the product and the solid circles represent the initial positions of the workers. The zigzag vertical lines show how these positions change over time and the rightmost spikes correspond to completed items. The system quickly stabilized so that each worker repeatedly executes the same portion of work content of the product.

This analysis suggests an effective way to partition a workforce into teams. Current practice in the apparel industry is to base the pay of each individual on the production of her team. Consequently, the fastest workers prefer to work with other fast workers. This is unattractive, however, from the perspective of management because it does not help integrate newer, slower workers into the workforce. If slower workers form teams to themselves there can be morale problems. In addition, the newer workers will not be in a position to learn directly from more experienced workers.

It seems better for management to put very different workers on the same team, sequenced from slowest to fastest; then each production line will be self-balancing and will achieve the maximum production rate. Furthermore, the greater the range of velocities on a team, the more powerfully the line will be drawn to balance. Finally, when there are large differences in the velocities of team members then the system will remain self-balancing even allowing for the inevitable variations in the velocities of the team members.

Improvements that are not

It is tempting to try to improve the performance of bucket brigade lines by modifying the protocol; however, the variants that come first to mind actually perform *worse*. For example, an appealing but flawed variation of the bucket brigade protocol is to allow any worker, when blocked, to leave his partially-completed item in a buffer before the busy station and walk back to take over the work of his predecessor. This variant protocol will increase work-in-process inventory and can even *reduce* the production rate! This can be seen in simulations, where workers tend to collect in the region of the line preceding any station that is frequently busy. This increases the production rate of the preceding segment of the line, which only accelerates the accumulation of in-process inventory immediately preceding the highly-utilized station. This, in turn, decreases overall production rate of the line for two reasons:

- Fewer workers remain to staff the final segment of the line so each tends to assume a larger share of work and the time between product completions increases.
- Because no one waits in front of the frequently busy station, it is idle every time a worker leaves it, which is contrary to the principal of keeping bottleneck stations constantly busy.

Eschewing buffers seems to contradict conventional wisdom that it is important to have buffers near a bottleneck — until one realizes that in bucket brigade production one must buffer both work-in-process *and* a worker, which is done by requiring the blocked worker to remain waiting at the bottleneck station.

One might also think that the bucket brigade protocol could be improved by requiring the workers to *circle* through the work stations. This avoids any delay in handing off work but it requires that every worker perform every task. There are several objections to be made to this. First, when real workers are finally assigned to the line they will not be of identical skill levels and so the production rate will eventually be determined by that of the slowest worker, behind whom all the others will accumulate. The production rate will remain suboptimal even if faster workers are allowed to preëmpt a slower worker and pass him: The slower worker would have to remain idle until his work station became free again and so the line could not keep all workers busy. Moreover, when workers are asked to perform every task on the line then the learning effect and so realized production rate will be reduced.

Use of bucket brigades in manufacturing

Bucket brigade manufacturing has many attractive properties, including:

- It is a pull system, so work-in-process inventory is strictly controlled.
- It requires no special material handling system because the workers themselves carry the items from station to station.
- Because the line can be made self-balancing, it does not require accurate measurement of task times and so can avoid some of the expense of time-motion studies.
- It is consistent with other trends in manufacturing: For example, it exploits the advantages of work teams and the grouping of technology into cells.
- The protocol is simple and identical for each worker: Workers are not confused as to what task to perform next and management need not intervene to keep work flow balanced and production rate high.

To the best of our knowledge, bucket brigades are currently used only in apparel manufacturing, where they have been introduced very recently; but the idea may be suited to other types of manufactuing. Bucket brigade manufacturing seems most appropriate when:

- ***All the work is based on a single skill.*** This ensures that workers can move among the stations to where the work is, without worrying about whether they can do the work there. It also allows workers to be ranked by a single score, their velocity along the production line, so that the line can be made self-balancing.

Economic forces ensure tend to move production lines in this direction, in which the primary worker skills are simple dexterity and enthusiasm.

- *A worker can move easily among stations and can easily take over work in process.* This ensures that the bucket brigade protocol does not introduce additional wasted time to pass work.
- *Work stations are inexpensive relative to labor costs.* This is to avoid significant penalty for the lower station utilization inherent in bucket brigade manufacturing.
- *Demand for the products varies significantly.* Bucket brigade manufacturing can more easily track changeable demand because cross-training of workers and low work-in-process inventory mean flexibility of configuration, and short production lead times. In addition, a bucket brigade line can be configured quickly: The assignment of tasks to stations need not be carefully balanced because the movement of the workers balances the line; this reduces the time required to lay out a new line and so shortens changeovers. Finally, because the line is self-balancing, production rates are easily adjustable by simply adding or removing workers from a team.

Bucket brigades in the warehouse

In many warehouses supporting retail sales, fast moving items are picked from cases stored in a type of shelving called *flow rack*. Within each bay (section of storage) are shelves with rollers and the shelves are tilted to bring the cases forward.

The bays of flow rack are arranged in aisles and a conveyor system runs down each aisle. The *start of an aisle* is the end that is upstream with respect to the movement of the conveyor. For clarity we will describe a single-aisle of flow rack. (Even when there are multiple aisles of flow rack, each aisle is generally operated as an independent module within the warehouse)

An *order* is a list of items for a single customer together with quantities to be picked. It is typical that orders are released in a batch each day to the picking operation. Then each order is picked by "progressive assembly": The order is picked by no more than one person at a time and the items are accumulated as the order is picked (rather than picking all orders simultaneously and sorting the items afterward).

Paperwork describing orders to be picked waits at the start of the aisle. Each order sheet lists the items and quantities to be picked in the sequence in which items will be encountered along the aisle. The first picker takes the next order sheet, opens a cardboard carton, and slides it along the passive lane of the conveyor as he moves down the aisle picking the items for that order. At some point the second picker takes over and continues picking that order while the first picker returns to the start to begin the next order. When the order is complete the carton(s) are pushed onto the powered portion of the conveyor, which takes them to the packing and shipping department.

There are several ways of coördinating the pickers. One way is to divide the bays into regions and to ask each picker to work within an assigned region: Worker 1 is responsible for picking all items lying within bays $1, \ldots, b_1$; worker 2 is responsible for picking all items lying within bays $b_1 + 1, \ldots, b_2$; and so on.

In designing such order-picking systems managers try to balance the expected work among the pickers. For example, a manager might look at last year's data and set values of $b_1, b_2, \ldots$ so that the total work over the previous year is about equally distributed. The trouble with this is that it balances the work only *on the average over the preceding year.* But orders can be very different from the average: Among other sources of variance, there are typically seasonalities; for example, orders might include many items from one part of the aisle in the summer and many items from another part of the aisle in the winter. Such a system might be balanced on the average but significantly out of balance every day.

The order-picking system will constantly seek balance if configured as a bucket-brigade with pickers sequenced from slowest to fastest. However, there is an important difference here: Unlike manufacturing the "items" produced on this line (that is, orders picked) are *not identical* and in fact are best modelled as "random". For example, one might think of each sku i in the warehouse as being in the next order with probability p_i independently of all other sku's. Because of this, the system converges to a state of balance only in a stochastic sense: It will be balanced on average. This is still an improvement over a static balance because:

- It is balanced on average *every hour*, not just over the year; therefore it will be out of balance much less often and so more productive.
- It spontaneously adapts to perturbations and seasonalities.
- It does not require anyone to compute a balance.

These advantages have been dramatically illustrated in an experiment we have been conducting since March

1995 in collaboration with a major chain retailer. We implemented a bucket brigade style of picking on a line at their national distribution center. This line is staffed by five workers, who typically pick about 35,000 units each day from flow rack. After changing to the bucket brigade protocol, their productivity, measured in average number of picks per person-hour, increased over 30% (Bartholdi and Eisenstein 1996b).

Previously, work on this line was assigned by a computer-based model of work content that was run the preceding night. Such a model cannot be accurate because

- It cannot economically account for all the relevant detail that determines work content, such as:
  - location, which might be at waist level or on an inconveniently high shelf.
  - shape and weight, which might make an item easy to grab or hard to handle.
  - velocities of the workers, who can range from 50–150% of standard.
  - distribution of locations: One worker might have her picks distributed over three bays while another has as many picks distributed over five bays.
  - additional work such as disposing of empty containers, sealing a full tote and opening another, prepping an sku, reaching to pull additional stock to the front of the flow rack, and so on.
  - economies of scale: most sku's picking two units is less than twice the work of picking one unit.
- Even though it might appear balanced on average, the allocation of work can nevertheless be quite unbalanced for every order.
- A static balance cannot adjust to unforeseen events such as equipment malfunction, employee absences, and so on.

Because the model of work content was inaccurate, as all such must be, considerable management time was devoted to adjusting the allocation of work during the day. (In fact, the retailer dedicated a manager to this.) The bucket brigade protocol has made this centralized managerial effort unnecessary — yet still results in better performance.

Summary

The ideas behind bucket brigades may be summarized as follows.

- Abolish rigid assignments of work, which prevent people from going to where the work is.
- Sequence the workers from slowest to fastest to make a "pull system" that is self-organizing.
- Emphasize teamwork. Base the incentive scheme at least partly on team productivity.

Finally, we emphasize that workers should follow the bucket brigade protocol strictly, even when they think they can make local improvements. In our experience workers have sometimes found it hard to resist making what they (mistakenly) thought to be improvements; but these almost always interfered with the global dynamics, which spontaneously reallocate the work to improve the production rate.

References

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

Bartholdi, J. J., III and Eisenstein, D. D. 1996b. Bucket brigades: A self-balancing order-picking system for a warehouse. Forthcoming.

Bartholdi, J. J., III; Seeley, T. D.; Tovey, C. A.; and Vande Vate, J. H. 1993. The pattern and effectiveness of forager allocation among food sources in honey bee colonies. *Journal of Theoretical Biology* 160:23–40.