Specialization, teamwork, and production efficiency

Stephen G. Powell*

Amos Tuck School of Business Administration, Dartmouth College, Hanover, NH 03755, USA

Received 21 September 1998; accepted 4 November 1999

Abstract

The division of labor has been a fundamental principle in the design of production processes since Adam Smith’s pin factory. More recently, teams have been employed widely to bring the benefits of cooperation and ownership to production and service processes. The fundamental questions raised here concern the relationships between these two approaches to the design of business processes. For example, under what conditions is it preferable to assign small tasks to isolated individuals, rather than assigning complex sets of tasks to large teams? We model the production process as a serial production line with variable processing times for each task. We develop models for the effect of dividing a complex production process into smaller tasks, and the effect of assigning teams of two or more workers to perform a given set of tasks. We determine the optimal size of teams and the optimal division of labor for a given set of assumptions about the underlying production environment. Our results show that, depending on the underlying parameters, a wide range of solutions to the production design problem can be found, including highly specialized processes with large teams, and unspecialized processes with small teams. These findings are in contrast to the principles enunciated in the literature on Business Process Reengineering, in which the case worker design is recommended as a universal solution. We also determine the regions in parameter space in which particular solutions can be found, and we study the sensitivity of the optimal choices to changes in the fundamental parameters. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Business process; Teams; Division of labor; Serial line

1. Introduction

In the production of physical goods or services two design choices are fundamental. One involves the division of labor: to what extent should the individual tasks involved in the production process be assigned to individual workers? At one extreme we have craft work, in which one worker performs all the tasks necessary to produce a finished product; at the other we have the extreme case of the division of labor in which each worker repetitively performs a single task. Another fundamental design choice involves whether workers will work separately or in teams. This choice is logically distinct from the choice of the division of labor, since teams can work together to produce the entire finished product (what might be called team craft work), or can work together on a small set of all the tasks needed to produce a final product. In this case multiple teams are needed to produce the finished product.

Similar design choices arise within the domain of Business Process Reengineering (BPR), which has
been defined as the application of industrial engineering principles to the design of service sector processes [1]. One of the fundamental insights in the early literature on BPR is that processes that have traditionally been carried out using an extreme division of labor can be drastically improved by replacing the serial production process with a single worker known as a case manager. Hammer cites an example involving the processing of insurance applications [2]. The original process involved 30 distinct steps, 19 workers, and five departments. In the reengineered process this serial production process is replaced by a parallel process, in which each application is handled from start to finish by a case worker (with support as needed from computer systems and a few specialists). There has been a consistent theme in the BPR literature that serial processes, with their extreme divisions of labor, multiple delay points, and limited ownership, are outdated and should be replaced by case-worker-based processes.

In fact, some see BPR as a revolution against traditional production principles:

**BPR is a direct attack against the principles of functional specialization and incremental improvement that lie at the very foundation of the industrial revolution.** [3]

Only recently has the research community begun to examine the claims of the BPR movement seriously. Buzacott, for example, used queueing theory to investigate the conditions under which the commonly cited principles of BPR are applicable [4]. In a series of papers, Seidmann and Sundararajan investigated the influence of task assignment, technology support, and performance incentives on optimal process design [5–8]. These papers suggest that some of the claims made within the BPR movement on the superiority of certain types of production processes may be exaggerated. This paper examines a related set of issues, using the tools of modeling and simulation to generate insights into the optimal design of production processes for goods or services.

In this paper we seek to answer the following general question: given a business process with a fixed number of elementary tasks and a fixed labor force, what is the optimal number of tasks and workers to assign to each team? To answer this question, we must first develop models that incorporate the changes in production efficiency that occur as tasks are assigned to a single worker, and as workers are assigned to a team. Once we have assigned the number of tasks and workers we have implicitly determined the length of the production process and the number of parallel processes. We can then determine the throughput of each production line and of the entire collection of lines. This allows us to determine the optimal assignments within the given constraints.

The fundamental trade-offs involved in the optimal assignment of tasks and workers involve the effects of specialization, teamwork, and production efficiency in serial production lines. For example, one would expect a relative loss of efficiency as more tasks are assigned to each worker since specialization is reduced. However, as each worker performs more tasks the length of the production process is reduced and this will result in less stochastic interference between workers and an increase in efficiency. Similarly, as more team members are assigned to a set of tasks one might expect efficiency to increase due to cooperation among workers (perhaps with diminishing returns) but the result will also be fewer parallel lines and perhaps less efficiency. These are the trade-offs we explore in this paper.

In the next section we will review the literature in two areas related to this work: production lines and BPR. In Section 3 we will develop models for task times as a function of the number of tasks and the number of team members. We will then use these models to determine line and factory throughput. Our strategy is to develop simple and flexible models at this basic level, so that we have a minimal number of easily interpreted parameters to work with. In Section 4 we use the model to explore the optimal design of the production process. In Section 5 we illustrate the sensitivity of the optimal solution to the underlying parameters, and analyze the parameter spaces within which certain important process designs (such as the parallel, case worker design) can be expected to be optimal. We conclude the paper with suggestions for further research.
2. Literature review

The analytical study of production processes has a long history in the management science and industrial engineering disciplines. Business Process Reengineering is a more recent field, with an extensive popular literature but only a thin research base. In this review we will concentrate on summarizing useful results from the production literature, and describe the handful of related papers in the BPR field.

2.1. Production literature

The literature on production processes is too extensive to review in detail here. Useful overviews can be found in [9–12]. Four principles summarize what is known in this field [9,13]:

1. line throughput (output per time period) decreases at a decreasing rate as the number of workstations increases;
2. throughput decreases as the CV increases for a line of given length;
3. small numbers of buffers are sufficient to recover a large proportion of the lost throughput due to stochastic interference, but large numbers are required to recover all of it;
4. the throughput of a line can be improved by allocating slightly less work than average to the center stations in the line, or by allocating extra buffers toward the center of the line.

Hillier and So [14] have recently investigated several phenomena of direct relevance to our work. They consider a production process in which there is a fixed amount of work that must be performed to make a given product. Management can choose how many workstations to use, how much of the total work to assign to each station, and how many servers to assign to each station. In general, the process Hillier and So envision is one in which \( N \) identical workstations operate in series, and at each station there are \( M \) parallel and identical servers.

Hillier and So find that if the number of stations is unrestricted, the optimal number is one. In other words, all the work on a given unit of output should be done by a single worker, with the entire labor force operating as parallel servers. This is a direct consequence of their assumption that division of labor has no effect on processing times. That is, one worker can do all the work, or two workers can each do half, without changing the total processing time. When one worker performs all tasks there is no blocking and starving possible; as soon as we divide up the work we have multiple workstations and throughput begins to drop due to blocking and starving. This parallel arrangement of servers also benefits from the effects of pooling the queue of arriving work.

If the number of workstations is fixed at \( N > 1 \), Hillier and So find that the optimal allocation of work is L-shaped: assign the great majority of work to the last station in the line, and a small amount to each of the other \( N - 1 \) stations. Also, one worker (the minimum) should be assigned to all stations but the last, which gets the majority of the labor force. This result, too, can be seen as a consequence of the assumption that division of labor does not increase work rates. The L-shaped allocation of work and workers is as close as the system can get (under the constraint of \( N \) stations) to the pure parallel system where each worker does all tasks.

Hillier and So discuss two limitations to their work. One is that the process time distributions are assumed to be exponential, regardless of the work allocation among the stations. A more detailed treatment would derive the distribution from assumptions about the distributions for individual tasks. Although not mentioned, another limitation of the exponential distribution is that when more work is allocated to a worker both the mean and standard deviation of processing times increase equally. The second shortcoming cited by the authors is that there is no benefit in their model for dividing up the labor required to produce a unit of output among multiple workers. Logic would suggest that there are situations in which, despite blocking and starving, it is advantageous to divide work up and create a serial line because specialization makes each worker sufficiently more effective. Finally, workers can be assigned to the same workstation but they do not work together as a team. Rather, multiple workers simply operate as independent, parallel servers for the incoming flow of
work. In the research reported here we explicitly model the benefits to specialization and the effects of assigning workers to cooperative teams.

2.2. Business process reengineering literature

Relatively little research has been reported specifically addressing the design of business processes, as opposed to production processes. Buzacott [4] cites nine principles of popular BPR, ranging from combining several tasks into one to providing a single point of customer contact with a case manager. For each of these principles he builds queuing models to compare system performance under alternative designs. His overall conclusion is that the recommended designs are optimal only when high levels of variability characterize task times, or when work arrives at a highly variable rate. For example, a parallel server system is preferred to a serial line only when the processing time variability exceeds a critical level, and to a lesser extent when the variability in arriving work is high. Buzacott models teams by assuming that two people performing two tasks involves each sampling from an independent exponential distribution with task time parameter \( \mu \). The overall task is completed only when the longer of these two tasks is finished. The distribution of the maximum task times then has the simple form \((1 - e^{-\mu t})^2\). Under these assumptions, the serial division of labor will perform better than the team once utilization is high. Buzacott notes that “Case teams must be set up to achieve genuine collaboration on individual task performance in order for them to be advantageous”.

In a related set of studies, Seidmann and Sundarajan [5–8] have undertaken a series of theoretical inquiries into the conditions under which typical BPR solutions may be optimal. They focus on the division of labor, performance incentives, and the information characteristics of business processes (for example, the knowledge required for a task and the symmetry of information between team members). Among many findings, they show that processes in which individual tasks vary significantly in size are good candidates for the case worker design. They also find that the case worker model in these cases is more robust and reliable in that system performance is steady through periods of fluctuating demand.

The overall conclusion from these analytical studies is that no one process design is optimal for all circumstances. This would be a less remarkable conclusion if simplistic generalizations were not as prevalent as they are in the popular literature on BPR. Among the most often-cited is the prescription that the case worker model is the ideal way to design service sector processes. Buzacott suggests that low task time variability is one condition under which this may not be the case; Seidmann and Sundarajan suggest that high information symmetry may be another. Our research is designed to further explore the trade-offs involved in the design of business processes.

3. Task and team effects

We consider two critical aspects of the design of business processes: how many workers and tasks to assign to each team. In order to make the problem more concrete, we assume we have a total labor force of 256 identical workers, and the item we are producing or service we are providing requires the execution of 16 separate tasks in a prescribed sequence. The time required to complete each of these tasks by a single worker in isolation is random, with a common mean \( \mu \) and standard deviation \( \sigma \). We will generally set \( \mu = 1.0 \) and \( \sigma = 0.5 \) so as to economize on parameters in the analysis.

Our ultimate goal is to find the best choice for the size of the teams and the number of tasks assigned to each team. This requires that we determine the efficiency of all possible designs for a business process in which all 256 workers are used and all 16 tasks are completed. At one extreme, we could assign one task to each worker and specify teams of size one as well. Then our “factory” would consist of 16 serial production lines with 16 workers each. At another extreme we could employ all 256 workers (in parallel) as craft workers so that each performed all 16 tasks. Yet another choice would involve teams of 16 workers doing a single task; in this case we would have one production line with 16 teams. Finally, we could employ 16 workers
to do all 16 tasks together; this gives us 16 parallel “lines” with a single team each. Of course there are many possible solutions between these extremes.

In order to determine the throughput of any of the possible designs that involve serial lines we need to know the probability distribution for task times at each station. Rather than modeling the complete probability distribution, we will rely on previous research [15,16], which suggests that the throughput of a serial line depends primarily on the first and second moments of the distribution of task times. Accordingly, we will use simulation to determine the throughput of the corresponding serial line using lognormal distributions with the given mean and standard deviation. The lognormal distribution is widely used in this context because it has the skewness observed in actual practice and it allows for continuous variation of its first two parameters [17,18].

We will use the following notation:

- $T_0$: number of tasks required to produce finished product
- $Y_0$: number of workers available
- $N$: number of tasks assigned to each team
- $M$: number of workers assigned to each team
- $L$: length of production line
- $X$: number of lines in factory
- $\text{MTT}(N, M)$: mean task time for $M$ workers to perform $N$ tasks
- $\text{STT}(N, M)$: standard deviation of task time for $M$ workers to perform $N$ tasks
- $\text{LCT}$: line cycle time (average time between completions of product or service)
- $\text{FCT}$: factory cycle time (average time between completions of product or service)

Several tautologies relate these variables. In particular:

- $L = T_0/N$: the number of stations in the production line is the total number of tasks divided by the number of tasks per team
- $X = Y_0/L \cdot M$: the number of lines in the factory is the total number of workers divided by the number of workers per line
- $\text{FCT} = \text{LCT}/X$: factory cycle time is line cycle time divided by the number of lines

Our modeling strategy is as follows. First we develop a model for the effects of task specialization on the work time of isolated workers (teams of one). Then we develop a model for the effects of team size when a team performs a single task. These models are then combined into a single model that gives the mean and standard deviation of task times for any combination of team and task size (choice of $N$ and $M$). We then use simulation and an approximation approach to develop an expression for the line cycle time for any $N$ and $M$. Finally, we can express factory cycle time as a function of line cycle time using the tautology above. Minimizing factory cycle time by choosing $N$ (the number of tasks) and $M$ (the size of teams) is our objective.

### 3.1. Task effects in isolation

Consider first the effects of adding tasks to the work of a single, isolated worker ($N = 1$). If it takes $\mu$ time units to perform one task (i.e., $\text{MTT}(1,1) = \mu$), it is likely to take somewhat more than $2\mu$ time units to perform two tasks. A worker performing two tasks has to switch between tasks, which will usually impose a setup time; in addition, such a worker may not become as proficient at either task as one whom performs only one task. Thus we would expect the total task time to increase faster than linearly with the number of tasks. A convenient model for this loss of efficiency due to decreased division of labor is the power curve

$$\text{MTT}(N, 1) = \mu N^a \quad \text{where } a > 1.$$  

The parameter $a$ represents the reduction in efficiency due to decreased specialization; the larger $a$ is the highly the penalty. For example, if $a = 1.1$, 16 tasks which individually take one time unit take 21.11 time units. The difference between 21.11 time units and the nominal time of 16 time units represents the loss in task time due to loss of specialization.
We next turn to the question of the variance of total task times as a single worker takes on more and more tasks. Recall that $\text{STT}(1, 1) = \sigma$ by assumption. The simplest plausible model suggests that a worker performing two tasks takes a random sample from the distributions for each task separately and then adds the resulting times to get a total processing time. If these samples are independent, and if the efficiency loss included in the mean has no variability, then the variance of the sum of two tasks is simply the sum of the variances of the individual tasks. Thus the standard deviation of task times rises with the square root of the number of tasks:

$$\text{STT}(N, 1) = \sigma N^{1/2}. \quad (2)$$

### 3.2. Team effects in isolation

Now assume that one task will be performed by each team ($N = 1$), and consider what happens to task times as we add successive workers to the minimal team of one. We reiterate that, in contrast to Hillier and So [14], we are not modeling the operation of workers in parallel but rather the operation of a true team, in which multiple workers cooperate to execute the tasks assigned to them in the most efficient manner. A natural starting assumption would be that both the mean and standard deviation of task times decline in proportion to the number of workers as team size increases. In other words, two workers complete one task in one-half the time on average with one-half the standard deviation. If this is the case, the throughput of any serial line exactly doubles when two workers are placed at each workstation. In general, throughput itself increases in proportion to the number of workers assigned to each team, and the overall throughput of the factory does not change. This zeroth-order model provides a useful reference point, but it does not appear to capture the true behavior of teams.

In the general case we will assume that, as successive members are added to a team, the team works out ways to do their tasks faster and with less variability, but that both effects show diminishing returns. Thus both the mean and standard deviation of task times fall faster than $1/M$ initially, but eventually level off above $1/M$. Simple models with these properties are

$$\text{MTT}(1, M) = \mu[(1 - c)M^{-b} + c], \quad b \geq 1, \ c \geq 0, \quad (3)$$

$$\text{STT}(1, M) = \sigma[(1 - e)M^{-d} + e], \quad d \geq 1, \ e \geq 0. \quad (4)$$

The parameters $b$ and $d$ represent the rate at which the task time mean and standard deviation decline as workers are added to the team. The higher these parameters are absolutely, the faster the mean and standard deviation decline. The parameters $c$ and $e$, on the other hand, represent floors below which the mean and standard deviation cannot fall. Thus they reflect the possibility of diminishing returns to team size. High values of $b$ and $c$, for example, represent a situation in which the first few members added to the team result in marked gains in processing efficiency, but beyond that point additional team members just get in the way and the processing time improves little more.

### 3.3. Combined task and team model

We now have simple models for the effects of adding tasks to a single worker and adding team members for a single task. Both models have a mean task time of $\mu$ and a standard deviation of $\sigma$ when $N = M = 1$. Thus it is natural to multiply the isolated task and team models to derive a composite model for all values of $N$ and $M$. The result is our general model:

$$\text{MTT}(N, M) = \mu N^a[(1 - c)M^{-b} + c], \quad (5)$$

$$\text{STT}(N, M) = \sigma N^{1/2}[(1 - e)M^{-d} + e]. \quad (6)$$

### 3.4. A model for line and factory cycle times

It is well known [18] that no analytic method exists to determine the cycle time of a serial production line with arbitrary processing time distributions. Most analytic studies rely on the exponential or Erlang distributions. These are not appropriate here because we wish to vary the mean and standard deviation independently. We will use the lognormal distribution, with mean and standard
deviation given by the models developed above, because it is convenient and it has the skewness known to characterize real processing time distributions. Given any choice of \( N \) and \( M \), and the values of the seven key parameters \( (\mu, \sigma, a, b, c, d, e) \), we can use simulation to determine the line cycle time (and thus factory cycle time) to any desired accuracy.

However, there is a practical drawback to using simulation in a study of this type. Theoretically, we could set the values of the underlying parameters and simulate all 256 separate cases for \( N \) and \( M \) from 1 to 16 to find the optimal work assignment. A simpler approach is suggested by the work of Martin and Lau [19], who show that the cycle time of a serial production line can be closely approximated by a relationship of the form

\[
b_1 + b_2 \ln(L),
\]

where \( L \) is the length of the line (number of stations) and the parameters \( b_1 \) and \( b_2 \) are themselves functions of the mean and standard deviation, respectively, of the processing time distribution.

Following this lead we performed a series of nonlinear regressions on simulation data for the 25 cases involving \( N \) and \( M \) that give integer numbers of lines \( (N, M = 1, 2, 4, 8, 16) \) to determine the best-fitting curve. We found for realistic values of the parameters that the following function fits with an average error on the order of 0.016 time units:

\[
\text{LCT} = 1.0 \text{MTT}(N, M) + 0.0675 \text{STT}(N, M) \ln(L).
\]  
(7)

This approximation gives us the following analytic expression for our objective function:

\[
\text{FCT} = (M/Y_0)(T_0/N)[\mu N^a((1 - c)M^{-b} + c) \\
+ 0.0675\sigma N^{1/2}((1 - e)M^{-d} + e) \ln(T_0/N)].
\]  
(8)

### 4. Optimal selection of teams and tasks

In this section we will show how the model given in (8) above can be used to determine the optimal assignment of tasks \( N \) and team size \( M \). We begin with a zeroth-order model, in which the parameters are selected to eliminate most of the interest-

ing trade-offs in the problem. This is not a realistic case, but it does provide a helpful benchmark. We then discuss a realistic base case, in which all the effects of the model are present and in which the optimal solution is interior.

The optimal solutions presented here are determined by explicit enumeration of the 256 cases for \( N \) and \( M \) between one and 16. Although we have a closed-form expression for the factory cycle time in (8), its first partial derivatives cannot be solved analytically.

#### 4.1. A zeroth-order model

An important, although unrealistic, special case of our model is one in which there is no benefit to specialization and both the mean and standard deviation decline in proportion to \( 1/M \). In this case the key parameters are \( a = b = d = 1 \) and \( c = e = 0 \). Under these assumptions, the mean task times increase linearly with \( N \) for fixed \( M \), and decline with \( 1/M \) for fixed \( N \). Likewise, the standard deviation of task times increases with the square root of \( N \) for fixed \( M \) and declines as \( 1/M \) for fixed \( N \). When we input these parameters into the equation for the factory cycle time we find that cycle time declines monotonically with \( N \) for any \( M \), and is constant as \( M \) varies with fixed \( N \) (Fig. 1). Thus the optimal design of the production process for this special case is to set \( N = 16 \) and to choose any value of \( M \) desired.

When specialization has no effect on processing time, so that two tasks can be done exactly twice as fast as one, the dominant effect in this model is the length of the production line. When \( N = 1 \) we have 16-station lines; when \( N = 16 \) we have 1-station lines. One-station lines cannot suffer blocking and starving, so they are more efficient in all cases (regardless of \( M \)). By the same token, when both the mean and standard deviation decline with \( 1/M \), the effect of adding a second worker to each workstation is simply to double the output rate. The line remains the same length, so stochastic interference is not affected. Doubling the workers on a line simply doubles the throughput rate while doubling the labor force employed. Thus all values of \( M \) give the same factory cycle time.
4.2. Base case model

A more realistic case is illustrated in Fig. 2, in which the specialization effect and diminishing returns to team size play a role. The relevant parameters in this case are $a = 1.13$, $b = 1.10$, $c = 0.01$, $d = 1.00$, and $e = 0.00$. Here the mean task time grows faster than $N$ as tasks are added, and it initially declines faster than $1/M$ as team size increases. The optimal choices of team size and number of tasks in this case are $N = M = 8$. Eight tasks and eight team members are assigned to each workstation, and the production lines (of which there are 16) consist of 2 stations in series.

In this case the beneficial effect of shortening the line by adding tasks to each worker is offset by the increase in mean task time. Instead of assigning all 16 tasks to one worker, the optimal solution is to assign 8 tasks. Also, as we add team members the mean task time initially drops faster than $1/M$, which has the beneficial effect of decreasing factory cycle time. However, diminishing returns to team size limits the team to eight members. The overall solution employs a combination of large teams with a large number of tasks to perform.

Fig. 2 also suggests a number of hypotheses about the relationship between team and task time and the resulting factory throughput. First, it is evident in this example that the worst choices lie at the boundaries. Small teams ($M = 1$) and a small number of tasks per team ($N = 1$) lead to the highest values of factory cycle time and thus the most inefficient use of resources. As we increase either or both of $N$ and $M$, factory cycle time declines until the optimum is reached at $N = M = 8$. However, the cycle time surface is quite flat in the region of
the optimum. In fact, if we choose $N$ and $M$ arbitrarily between 6 and 10 (that is, within two of the optimal values) the maximum loss in cycle time would be 0.8%. This contrasts to a maximum loss of cycle time of almost 50% when $N$ and $M$ are set to the worst possible values (i.e., 1). Of course, the significance of a one-percent loss in cycle time depends on the specific circumstances. Even an amount this small could be of significance in high-volume, high-cost operations.

5. Sensitivity analysis

The most interesting aspects of a model such as ours are its sensitivities to key parameters, not its solution for a single set of parameters. Accordingly, in this section we will discuss how the optimal solution for the number of tasks and team size changes as we change the parameters $a, b, c, d, \text{ and } e$. We begin by describing some of the fundamental properties of all solutions to this model. Then we perform a parametric sensitivity analysis around the base case of the previous section. Finally, we characterize sets of parameters that give rise to solutions lying in particular regions of $N-M$ space.

5.1. Analytic properties of the solution

Our objective function for factory cycle time (8) is sufficiently complex that no analytic simultaneous solution of its first-order conditions is possible. However, we can analyze the first-order conditions independently and derive some properties of the solution.
The first-order condition for optimality with respect to $N$ is
\[
\frac{\partial \text{FCT}}{\partial N} = \frac{(MT_0/Y_0)[N^{a-2}(a-1)(1-c)M^{-b} + c] - N^{-3/2}(1 + (1/2)\ln(T_0/N))0.0675}{\sigma((1 - e)M^{-d} + e)} = 0.
\] (9)

**Proposition 1.** If $a = 1$, factory cycle time is strictly decreasing in $N$.

**Proof.** Since $1 \leq N \leq T_0$, $\ln(T_0/N) \geq 1$, thus the second term in (9) is negative. The first term is zero when $a = 1$. □

When $a = 1$ there is no penalty in our model to combining tasks at a single workstation. Since combining tasks reduces the length of the production line, it reduces stochastic interference and thus increases factory cycle time. This effect is illustrated in Fig. 1.

**Proposition 2.** There exists a value for $a$ for which $\partial \text{FCT}/\partial N = 0$ for some $1 < N < T_0$.

**Proof.** When $\partial \text{FCT}/\partial N = 0$, $Na^{-2}(a-1)$ and $N^{-3/2}(1 + (1/2)\ln(T_0/N))$ differ only by a constant independent of $N$. The first term is increasing in $a$ and the second term is independent of $a$, thus we can find a value of $a$ for which the optimal $N$ is strictly interior.

This proposition establishes the useful fact that different values of the parameter $a$ can result in any given solution for $N$. In effect, as we increase $a$ from its nominal value of 1.0, we find that factory cycle time is maximized at lower and lower values of $N$.

The first-order condition for optimality with respect to $M$ is
\[
\frac{\partial \text{FCT}}{\partial M} = \frac{(T_0/Y_0 N)[\mu N^a(1-c)(1-b)M^{-b} + c]}{\sigma N^{1/2}(1 - e)(1 - d)M^{-d}} + \frac{0.0675\sigma N^{1/2}\ln(T_0/N)}{\ln(T_0/N)} = 0.
\] (10)

**Proposition 3.** If $b = d = 1.0$ and $c = e = 0.0$, factory cycle time is independent of $M$.

**Proof.** Follows immediately from (10).

This proposition reflects the importance of comparing the effect of $M$ on the mean and standard deviation of task times to the nominal case $1/M$. When both the mean and standard deviation decline as $1/M$, the increase in throughput is exactly proportional to the number of workers used, so all choices of $M$ result in the same factory cycle time. This effect is also illustrated in Fig. 1.

**Proposition 4.** If $b > 1$ or $d > 1$ (or both), and $c = e = 0.0$, factory cycle time declines with $M$.

**Proof.** Under the given conditions,
\[
\text{sgn}[\partial \text{FCT}/\partial M] = \text{sgn}[\mu N^a(1 - b)M^{-b} + 0.0675\sigma N^{1/2}\ln(T_0/N)(1 - d)M^{-d}].
\]
If either one of $b$ or $d$ is 1 the proof is immediate. If both are strictly greater than 1, $\text{sgn}[\mu N^a(1 - b)M^{-b} + 0.0675\sigma N^{1/2}\ln(T_0/N)(1 - d)M^{-d}] < 0$ when $(1 - b)(1 - d)M^{d - b} > 0$, which follows from our assumptions.

This proposition leads to the conclusion that if either the mean or standard deviation of task times declines faster than $1/M$ and there is no floor to this effect, then the optimal solution is $M^* = 16$, that is, teams should be as big as possible. This solution is no doubt unrealistic in most practical cases, but it helps to illustrate the significance of the various parameters in our model.

5.2. Parameter sensitivities

Table 1 presents the optimal solution for the number of tasks ($N$) and the number of team members ($M$) for various values of the key parameters, starting with the base case solution of Section 4.2. In each of these sensitivity analyses we vary one parameter at a time, keeping all other parameters fixed at their base case levels. We then observe how the optimal solution moves away from the base case solution $N^* = M^* = 8$.

We first vary the parameter $a$, which represents the loss in efficiency due to decreased specialization. The larger $a$ is the longer it takes a worker to perform multiple tasks. As we vary $a$ from a low
of these parameter values, when
number of tasks. To give a sense of the magnitude
penalty for combining tasks the fewer the optimal
take two time periods. This result suggests that the
value of 1.0 to a high of 1.50, the optimal design
changes from one in which large teams (M = 8) do
do all 16 tasks to one in which the same large teams do
only one task. As we would expect, the larger the
penalty for combining tasks the fewer the optimal
take two time periods; when a = 1.15 two tasks
take 2.22 time periods. This result suggests that the
optimal number of tasks per team is quite sensitive
to the parameter a, but that the team size itself is
independent of it.

Next we investigate the influence of the para-
meter b, which affects how steeply the mean task
time drops as additional members are added to
each team. As we vary b from 1.0 to 1.5, we see that
the optimal design changes from N = 7, M = 1 to
N = 16, M = 13. Unlike a, the parameter b has an
influence on both elements of the design. If b is near
1.0, the mean task time drops at close to the rate
1/M with increasing team size. At this rate, adding
team members has no effect on the optimal design,
as our zeroth-order model showed. As b increases,
the mean task time for a team drops faster than
1/M, increasing the incentive to choose larger
tasks. As M increases, the standard deviation of task
times decreases according to (4) so the penalty
to assigning multiple tasks to the team drops, and
the optimal value increases. To give a sense of the
magnitude of these effects, when b = 1.5 (and
c = 0.01), two team members take 0.36 time units to
perform one task (versus 0.505 time units when
b = 1.0).

The parameter c sets a floor for the mean task
time as a function of team size. The higher c, the
sooner diminishing returns set in to increasing
team size. In Table 1 we see that increasing c has
a dramatic effect on optimal team size, but little
effect on the optimal number of tasks. In our base
case, with c = 0.01, the optimal solution is
N = M = 8. When c = 0.0, the optimal solution changes rather dramatically to N = 9, M = 16.
With no floor to the mean task time it is optimal to
assign the maximum number to each team. At the
same time, the optimal number of tasks increases
only slightly. At the other extreme, when c = 0.10,
the optimal solution is teams of one and seven tasks
per team.

The parameter d plays an analogous role to that
played by b in mean task times: it governs the rate
at which the standard deviation of task times
declines with team size. In our base case, d = 1.0,
which means that the standard deviation of task
times drops at the rate 1/M. As we increase d in
Table 1 we see that the optimal solution moves in
the direction of a lower number of tasks and higher
team size. When d reaches 1.6, the solution has
moved to the extreme case N = 1, M = 16, in
which teams of 16 workers perform a single task.
To give a point of reference, when d = 1.6 (and
e = 0.0), two team members can perform a task
with a standard deviation of 0.323σ (versus 0.5σ
when d = 1.0). When d is high, the variability of
task times is so low that the penalty in lost produc-
tion from long serial production lines is all but
eliminated. Thus the optimal design shifts towards
long lines with large teams.

| Table 1 | Sensitivity of optimal solution to parameters (base case par-
<p>| parameters in bold) |</p>
<table>
<thead>
<tr>
<th>a</th>
<th>N*</th>
<th>M*</th>
<th>b</th>
<th>N*</th>
<th>M*</th>
<th>c</th>
<th>N*</th>
<th>M*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>16</td>
<td>8</td>
<td>1.00</td>
<td>7</td>
<td>1</td>
<td>0.00</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>1.10</td>
<td>11</td>
<td>8</td>
<td>1.05</td>
<td>7</td>
<td>5</td>
<td>0.01</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>1.11</td>
<td>9</td>
<td>8</td>
<td>1.10</td>
<td>8</td>
<td>8</td>
<td>0.02</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>1.12</td>
<td>9</td>
<td>8</td>
<td>1.15</td>
<td>8</td>
<td>10</td>
<td>0.03</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>1.13</td>
<td>8</td>
<td>8</td>
<td>1.20</td>
<td>9</td>
<td>12</td>
<td>0.04</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1.15</td>
<td>6</td>
<td>8</td>
<td>1.30</td>
<td>11</td>
<td>14</td>
<td>0.05</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1.20</td>
<td>4</td>
<td>8</td>
<td>1.40</td>
<td>14</td>
<td>14</td>
<td>0.10</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>1.25</td>
<td>3</td>
<td>8</td>
<td>1.50</td>
<td>16</td>
<td>13</td>
<td>0.15</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>1.30</td>
<td>3</td>
<td>8</td>
<td>1.40</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.50</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>d</th>
<th>N*</th>
<th>M*</th>
<th>e</th>
<th>N*</th>
<th>M*</th>
<th>σ</th>
<th>N*</th>
<th>M*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>8</td>
<td>8</td>
<td>0.10</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>1.10</td>
<td>6</td>
<td>9</td>
<td>0.01</td>
<td>8</td>
<td>8</td>
<td>0.20</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>1.20</td>
<td>5</td>
<td>10</td>
<td>0.02</td>
<td>9</td>
<td>7</td>
<td>0.30</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1.30</td>
<td>4</td>
<td>10</td>
<td>0.03</td>
<td>9</td>
<td>7</td>
<td>0.40</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>1.40</td>
<td>3</td>
<td>10</td>
<td>0.04</td>
<td>10</td>
<td>7</td>
<td>0.50</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>1.50</td>
<td>2</td>
<td>13</td>
<td>0.05</td>
<td>10</td>
<td>7</td>
<td>0.60</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>1.60</td>
<td>1</td>
<td>16</td>
<td>0.06</td>
<td>10</td>
<td>7</td>
<td>0.70</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>0.07</td>
<td>11</td>
<td>7</td>
<td>0.80</td>
<td>12</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.08</td>
<td>11</td>
<td>7</td>
<td>0.90</td>
<td>14</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.09</td>
<td>12</td>
<td>7</td>
<td>1.00</td>
<td>15</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.10</td>
<td>12</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>16</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The parameter $e$ provides a floor for the standard deviation of task times just as $c$ does for the mean. In Table 1 we see that as we increase $e$ from the base case value of 0.0 the optimal solution moves gradually toward higher numbers of tasks for slightly smaller teams. The effect of this parameter is muted, both because it is only a floor and because its influence is on the second moment of task times.

Finally we come to the parameter $p$, which is the standard deviation of task times for the case $N = M = 1$. As Table 1 shows, this parameter has a very simple effect on the solution: the higher the value of $\sigma$ the larger the optimal number of tasks per team. Variability in task times induces the blocking and starving that make long lines inefficient. Thus an increase in variability, other things equal, leads to shorter lines with less opportunity for stochastic interference. The optimal team size itself, on the other hand, is not influenced by $\sigma$.

5.3. Parameter spaces

Another interesting angle from which to view this model is to ask what combinations of the key parameters lead to extreme solutions for the optimal team size and number of tasks. For example, which values of $a$, $b$, $c$, $d$, $e$, and $\sigma$ lead to small values for $N$ and $M$, and thus to small teams specializing in a few tasks. Table 2 reports on a series of experiments in which we determined combinations of parameters leading to extreme solutions. We will discuss four cases, one for each combination of high and low values for $N$ and $M$.

Case 1 represents solutions with low values for both team size and tasks per team. These solutions typically come about with high values of $a$ and $c$, and low values of the remaining parameters. High values of $a$ penalize the assignment of multiple tasks to a team, leading to low values of $N$. Low values for $\sigma$ reinforce this tendency, since low values of $N$ lead to long lines but low $\sigma$ implies limited blocking and starving. High values for $c$ and low values for $b$ penalize large teams with high mean task times, leading to low values for $M$.

Case 2 represents solutions with low values for the number of tasks and high values for team size. These solutions typically come about with high values for $a$, $b$, and $d$, and low values for the other parameters. High values for $b$ and low values for $c$ encourage the use of large teams, since the mean task time falls much faster than $1/M$. Small numbers of tasks per team are ensured by high values for $a$.

Case 3 represents solutions with high values for the number of tasks and low values for team size. This case represents the opposite of the previous one. Here, high values for $c$, $e$, and $\sigma$ and low values for the other parameters lead to small teams performing larger numbers of tasks.

Finally, Case 4 represents solutions with high values for both team size and tasks per team. These solutions can be brought about with high values for $b$, $d$, and $\sigma$, and low values for the other parameters. A low value of $a$ ensures large numbers of tasks per team, while a high $\sigma$ ensures small teams.

6. Future research

In this paper we have attempted to determine conditions under which it is preferable to assign a small number of tasks to isolated individuals, and those under which it is preferable to assign complex sets of tasks to large teams. We have developed a simple model that determines the cycle time of a factory or service process that uses a fixed number of workers in any feasible combination of
assignments of number of tasks and team size. This model requires us to determine the mean and standard deviation of task times for a team of a given size performing a set number of tasks. Our models for these relationships use flexible functional forms and a small number of parameters, each of which can easily be interpreted.

Determining the optimal choices for team size and the number of tasks to assign to each team involves complex trade-offs among specialization, teamwork, and production efficiency. Our model allows for all these influences to be taken into account and the combination of team size and number of tasks chosen that minimizes factory cycle time. Our results show that with plausible parameters, any combination of team size and task size could be optimal.

Sensitivity analysis reveals the impact of critical parameters on the optimal choices. We find that three parameters are especially critical. One represents the effect of specialization on mean task times, a second represents the effect of increasing team size on mean task times, and the third measures the underlying variability in task times.

These results reinforce and deepen those of previous researchers into BPR, in particular Buzacott [4] and Hillier and So [14]. In particular, they cast further doubt on the folk wisdom in the BPR literature that the case worker solution is generally to be preferred. Moreover, this research helps to identify the critical issues in determining an appropriate business process in a specific situation, and suggests which parameters governing the underlying behavior of teams are critical. In this way it can help inform managerial judgments on the design of business processes.

Future research can lead in many directions. Our models for the mean and standard deviation of task times are generic in the sense that they are designed to reflect broad effects in a flexible way. Empirical research that would refine or improve on these underlying models would be helpful. Our models also assume away many aspects of human behavior, such as responses to different incentive or compensation plans. An interesting extension of this research would be to determine the optimal compensation scheme simultaneously with the optimal team size and task assignment.

Acknowledgements

I would like to thank John Harris, Michelle Moore, and Joe Cooper for assistance in this research; the comments of Ken Baker and Praveen Kopalle are also appreciated.

References


