

1 Throughput, Work-in-Process, and Cycle Time

The purpose of this section is to obtain a basic understanding of the relationship that exists between throughput, work-in process, and cycle times. We will assume a production system where parts go through a specific routing. The *throughput*, denoted by TH , is the average output per unit time, e.g., number of widgets per hour. The inventory between the start and the end points of a production system (not counting crib inventory or finished goods inventory) is called *work-in-process* (WIP). The *cycle time* (CT) of a given production system is the average time parts spend as WIP.

The bottleneck rate r_b is the rate of the process center having least long-term capacity. The raw processing time T_o is the sum of the long-run average processing times of each workstation in the routing. Critical WIP is defined to be $W_o = r_b T_o$.

Example 1: Consider a production line with four stations, and assume that the processing time at each workstation is two hours. Since the processing rate at each of the workstations is 0.5 parts per hour, it follows that $r_b = 0.5$. Since producing the part takes 2 hours in each workstation it follows that $T_o = 8$ so the bottleneck rate is half a unit per hour and the raw processing time is eight hours. Finally, critical WIP is $W_o = r_b T_o = 0.5 * 8 = 4$.

Let us find the cycle time and the throughput rate for different WIP values w assuming the machines are perfectly reliable and the processing times are deterministic. What is the average cycle time when $w \leq W_o$? When $w \leq W_o$ parts do not have to wait so $CT = T_o$ for $w \leq W_o$. What happens if $w > W_o$. In this case units have to wait so $CT = w/r_b$ for $w > W_o$. For throughput we have $TH = w/T_o$ for $w \leq W_o$, and $TH = r_b$ for $w > W_o$.

Example 2: For the data of example 1 we see that $CT = 8$ for $w \leq 4$ and $CT = 2w$ for $w > 4$. As for throughput we have $TH = .125w$ for $w \leq 4$ and $TH = 0.5$ for $w > 4$.

Observations: For any level of WIP w , we have $TH = w/CT$. This relationship actually holds quite generally, and it is known as Little's Law. A balance deterministic line is the best we can hope for, so in effect we have

$$CT \geq CT_{best} = \begin{cases} T_o & \text{if } w \leq W_o \\ \frac{w}{r_b} & \text{if } w > W_o \end{cases}$$

and

$$TH \leq TH_{best} = \begin{cases} \frac{w}{T_o} & \text{if } w \leq W_o \\ r_b & \text{if } w > W_o \end{cases}$$

Increasing the bottleneck rate r_b improves both CT and TH on WIP levels above W_o . Reducing T_o improves CT and TH only for WIP levels below W_o . In general, increasing the processing rate of bottleneck workstations, to the point that all workstations are balanced results in a system that is more congested, compared to another one with the same bottleneck rate.

To find the cycle time and the throughput under the worst case, imagine that we transport jobs through the line on pallets. Whenever a job is finished, it is removed from its pallet and the pallet immediately returns to the front of the line to carry a new job. In this way, the WIP is kept constant, but now parts queue in front of each workstation. To see this, consider what happens for our balance line example with $w = 4$. When the pallet arrives at a workstation it finds the workstation idle. The pallet stays at this workstation until all the units in the pallet are processed. This takes 8 hours. Since there are 4 workstations, the cycle time will be 32 hours and the throughput will be 1/8. More generally, with a WIP equal to w we have $CT_{worst} = wT_o$ and $TH_{worst} = 1/T_o$ for all $w \geq 1$.

$$CT \leq CT_{worst} = wT_o$$

and

$$TH \geq TH_{worst} = \frac{1}{T_o}$$

Virtually no real-world line behaves according to either the best or the worst case and it is instructive to consider intermediate cases. We now consider a case where processing times are

random. We will assume that the processing times are exponentially distributed. We say that a random variable T is exponentially distributed with parameter λ if $P(T > t) = e^{-\lambda t}$. The exponential distribution is the only continuous time distribution with the memoryless property. This property states that if s units have already elapsed and the processing has not been completed, the distribution of the additional time until completion is the same as if we were starting again. Mathematically, this can be written as

$$P(T > s + t | T > s) = P(T > t).$$

It is because of this property that some people say that the exponential distribution represents the maximum randomness although this is technically debatable.

It can be shown that under the exponential distribution,

$$CT = T_o + \frac{w - 1}{r_b}$$

and

$$TH = \frac{w}{W_o + w - 1} r_b.$$

We will not go over the detailed derivation of these formulas. However, it is possible to give an intuitive justification of the formulas as follows. Suppose that you want to determine the cycle time for a particular job. Suppose that when you arrive at a station you expect the other $w - 1$ jobs to be equally distributed among N workstations. In this case, you expect to see $\frac{w-1}{N}$ jobs ahead of you. If the processing time at each station is T_o/N then the time you spend at each station is equal to $\frac{T_o}{N} (1 + \frac{w-1}{N})$. Multiplying by N we obtain the cycle time

$$CT = T_o (1 + \frac{w-1}{N}).$$

Using Little's Law we can find the throughput

$$TH = \frac{w}{CT} = \frac{w}{W_o + w - 1} r_b,$$

where we have used the fact that $W_o = r_b T_o$. The last two formulas define the practical worst case. If the performance of a system is between the best case and the practical worst case then the system is been managed effectively. On the other hand, if the performance of a system is between the practical worst case and the worst case, then there may be significant opportunities for improvements.

2 Push and Pull Systems

A push system schedules the release of work based on demand forecasts, while a pull system authorizes the release of work based on system status. In a push system we control the release of work also known as the daily going rate and observe WIP. In a pull system we control WIP and observe throughput.

Example: Five machines in tandem, each station processes jobs at a rate of one per hour, and processing times are exponentially distributed. Suppose we release jobs so that WIP stays at level w . Then, the throughput is given by

$$TH = \frac{w}{4 + w}.$$

Now suppose we release jobs into the same system at rate TH . It can be shown that the WIP level will be

$$WIP = 5 \frac{TH}{1 - TH}.$$

Thus, for example, to achieve $TH = w/(4 + w)$, a push system would require $WIP = 1.25w$, or 25% more WIP than a pull system with constant WIP.

Suppose our profit function is of the form $pTH - hw$. In a push system we need to determine TH to maximize the profit function. In a pull system we would control w to maximize the system. For the above problem, the PUSH profit function becomes

$$pTH - h\frac{5TH}{1 - TH},$$

while for a pull system the profit function becomes

$$p\frac{w}{4 + w} - hw.$$

Drawing these functions reveals that we can make more profit under a pull system, and that it is more robust to errors in the optimal quantities.