12 Production lines

In this chapter we will consider production lines, which consist of several work stations in series, possibly with buffers in between. A work station is a group of (parallel) machines or operators performing one or more operations on the jobs.

One of the issues in the design of production lines is the (minimal) capacity (i.e., number of machines or operators) that is required to achieve a certain target throughput level TH. Denote the number of machines by m, and let W be the total work content of a job. Then, the maximum number of units work that can be processed per time unit is equal to m, and the amount of work that should be done per time unit is $TH \cdot W$. Hence, to realize a target throughput TH, we need

$$m \geq TH \cdot W. \tag{1}$$

Usually m should be greater than this minimum, due to line unbalance, variability in processing times, failures, etc. Other issues in the design and operation of production lines are, e.g., the degree of paralleling of work stations, location and size of buffers in between workstations, choice of the material handling system, allocation of tasks to work stations, and the assignment of operators to work stations.

Production lines may be divided into two groups: *synchronous* and *asynchronous* lines. In synchronous lines the movement of jobs is coordinated; all jobs move to the next work station simultanuously. So the number of jobs in the system remains constant, and there is no need to put buffers in between stations. This type of production line may be further split up between *paced* and *unpaced* lines. In a paced line the time allowed for an operator or machine to work on the job is limited. Once the time is up the job can be no longer worked on, and thus it is possible that the processing is not completed when the job moves on. In an unpaced line there is no maximum limit imposed on the processing time available to the operator or machine.

In asynchronous lines the movement of jobs is not coordinated. The operator or machine starts to process the next job as soon as one becomes available. And on service completion the job immediately moves to the next work station, as long as there is space for it. Thus an operator or machine can become *starved* (no job available) or *blocked* (no room to put a completed job in the downstream buffer). Asynchronous lines are almost always unpaced. The number of jobs in the system may fluctuate (considerably) and buffers are needed to prevent starvation and blocking (i.e., loss of capacity) as much as possible.

In the following sections we start to look at synchronous lines.

12.1 Unpaced synchronous lines

We consider an unpaced synchronous production line, with m machines in series, and we want to investigate the impact of the variability in the processing times on the throughput of the line. Let the random variable B_i, $i = 1, 2, \ldots, m$, denote the processing time at machine i, with distribution function

$$F_{B_i}(t) = P(B_i \leq t), \qquad t \geq 0.$$

Further, let C be the (random) cycle time, i.e., the time that elapses between two subsequent job transfers. Since the jobs are transfered to the next machine in the line, once all machines have finished processing, it follows that

$$C = \max\{B_1, B_2, \ldots, B_m\}.$$

Thus

$$F_C(t) = P(C \leq t) = P(B_1 \leq t, B_2 \leq t, \ldots, B_m \leq t) = F_{B_1}(t)F_{B_2}(t)\cdots F_{B_m}(t)$$

(since the processing times at the machines are assumed to be independent), and

$$E(C) = \int_{t=0}^{\infty}(1 - F_C(t))dt = \int_{t=0}^{\infty}(1 - F_{B_1}(t)F_{B_2}(t)\cdots F_{B_m}(t))dt.$$

The throughput TH follows from

$$TH = \frac{1}{E(C)}.$$

Example 12.1 If the processing times are all uniformly distributed on $(0, 1)$, i.e.,

$$F_{B_i}(t) = t, \qquad 0 \leq t \leq 1,$$

then we have

$$E(C) = \int_{t=0}^{1}(1 - t^m)dt = 1 - \frac{1}{m+1}.$$

Hence the throughput is always greater than 1, and for large m nearly equal to 1.

Example 12.2 If the processing times are all exponentially distributed with mean $1/2$, then it follows from the memoryless property of exponentials that

$$E(C) = E(\max(B_1, B_2, \ldots, B_m)) = \frac{1}{2}\left(\frac{1}{m} + \frac{1}{m-1} + \cdots + \frac{1}{2} + 1\right).$$

Hence, as m tends to infinity, then $E(C)$ also grows to infinity (as $\log(m)/2$), and thus the throughput tends to 0.

12.2 Paced synchronuous lines

In paced lines we have a *fixed cycle time* c. An important performance measure is the throughput, but also the *quality* of the jobs. Let $Q(c)$ denote the probability that a job at the end of the line has no defect, i.e., each machine in the line succeeded in completing the processing of the job. Thus

$$Q(c) = F_{B_1}(c)F_{B_2}(c)\cdots F_{B_m}(c).$$

m	c
5	5.51
10	6.21
20	6.90

Table 1: Minimal cycle time c to meet $Q = 0.98$ when all processing times are exponential with mean 1

In table 1 below we list the minimal cycle time c to meet $Q = 0.98$, where the number of machines m ranges from 5 to 20; all processing times B_i are exponentially distributed with mean 1.

The throughput of *good jobs* (i.e., jobs without defects) is

$$TH = \frac{Q(c)}{c}.$$

Clearly there is a trade-off between the throughput and the quality of the output $Q(c)$. If the cycle time is too small, the time available to perform the job is too small and hence the quality will be low, whereas if the cycle time is too big, the throughput will be low. The maximal throughput TH will be achieved when

$$\frac{d}{dc}\frac{Q(c)}{c} = \frac{Q'(c)c - Q(c)}{c^2} = 0,$$

or

$$Q'(c) = \frac{Q(c)}{c}.$$

Denote the cycle time maximizing TH by c^*. When the cycle time c is increased beyond c^*, then the quality will improve, and the throughput will decrease. But if the cycle time is decreased below c^*, then both the quality and the throughput will decrease. Hence, the cycle time c should never be set to be less than c^*; beyond c^* a tradeoff has to be made between quality and throughput. In table 2 we list c^* and the corresponding quality and throughput for lines of 5, 10 and 20 machines, when all processing times are exponential with mean 1.

m	c^*	$Q(c^*)$	TH
5	2.55	0.66	0.26
10	3.60	0.76	0.21
20	4.50	0.80	0.18

Table 2: Cycle time c^* for which TH is maximal, when all processing times are exponential with mean 1

12.3 Asynchronuous lines with exponential processing times

In this section we consider an asynchronuous production line with m machines in series; the machines are numbered $1, 2, \ldots, m$. The processing times at machine i are exponentially distributed with parameter μ_i, and jobs arrive at the first machine according to a Poisson stream with rate λ. Each machine has a buffer with infinite capacity (i.e., there is always room for jobs). The system is depicted in figure 1. We now want to determine the mean total number of jobs in the system and the mean production lead time.

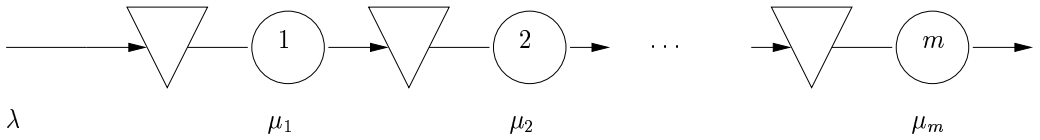

Figure 1: Production line with infinite buffers, exponential processing times and Poisson arrivals

To guarantee that the line can handle the amount of work offered per time unit, we have to require that $\lambda < \mu_i$ for each machine i. From the property that the output process of an $M/M/1$ system is again Poisson, it follows that the inflow at each machine is a Poisson stream with rate λ. Hence, each machine in the line can be modelled as an $M/M/1$ with arrival rate λ and service rate μ_i. Thus denoting the number of jobs at machine i by L_i, we find

$$E(L_i) = \frac{\rho_i}{1-\rho_i}, \qquad i = 1, 2, \ldots, m,$$

where $\rho_i = \lambda/\mu_i$, and for L, the total number of jobs in the system,

$$E(L) = \sum_{i=1}^{m} E(L_i) = \sum_{i=1}^{m} \frac{\rho_i}{1-\rho_i}.$$

By Little's law the mean production lead time $E(S)$ is given by

$$E(S) = \frac{E(L)}{\lambda} = \sum_{i=1}^{m} \frac{1/\mu_i}{1-\rho_i}.$$

Example 12.3 Let us consider the following *work load allocation problem.* The mean total work load for each job is W, and we have to allocate the work load among the m machines. Let w_i denote the mean work load allocated to machine i; we assume that the work load allocated to machine i is exponentially distributed. The problem is to find a work load allocation minimizing the mean total number of jobs in the system (or equivalently, minimizing the mean production lead time). Of course, it should hold that $m \geq \lambda \cdot W$; otherwise there is no feasible work load allocation (cf. (1)). Thus we have to solve

$$\min \sum_{i=1}^{m} \frac{\lambda w_i}{1-\lambda w_i}$$

$$
\begin{aligned}
&\text{subject to} \\
&\sum_{i=1}^{m} w_i = W, \\
&0 \leq \lambda w_i \leq 1, \quad i = 1, 2, \ldots, m.
\end{aligned}
$$

It can be shown that the optimal solution is given by

$$w_i = \frac{W}{m}, \qquad i = 1, 2, \ldots, m.$$

Hence, it is optimal to *balance the line*.

In the previous example we have seen that it is optimal to balance the line. But what is the impact of unbalance on the mean total number of jobs in the system? This is illustrated in the following example.

Example 12.4 We consider a line with 4 machines. The arrival rate at the first machine is 1 job per time unit. The mean processing time at machine i is denoted by w_i. Further, $\rho_i = \lambda w_i$ and ρ is the average work load per machine, defined as

$$\rho = \frac{1}{4}(\rho_1 + \rho_2 + \rho_3 + \rho_4).$$

In table 3 we list the mean total number of jobs, $E(L)$, for various values of ρ and different work load allocations. Clearly, in heavy load situations, slight unbalance may have a strong impact on the mean total number of jobs in the system (and thus also on the mean production lead time).

ρ	w_1	w_2	w_3	w_4	$E(L_1)$	$E(L_2)$	$E(L_3)$	$E(L_4)$	$E(L)$
0.80	0.85	0.65	0.90	0.80	5.7	1.9	9.0	4.0	20.5
0.80	0.80	0.80	0.80	0.80	4.0	4.0	4.0	4.0	16.0
0.90	0.95	0.83	0.97	0.85	19.0	4.9	32.3	5.7	61.9
0.90	0.90	0.90	0.90	0.90	9.0	9.0	9.0	9.0	36.0
0.95	0.96	0.93	0.97	0.94	24.0	13.3	32.3	15.7	85.3
0.95	0.95	0.95	0.95	0.95	19.0	19.0	19.0	19.0	76.0

Table 3: Mean total number of jobs in the system for different work load allocations

Remark 12.5 In this section we considered the situation where at each production stage there is exactly one machine available. The analysis is very similar in case production stage i, $i = 1, 2, \ldots, m$, is performed by a group of c_i parallel and identical machines. For stability we then have to require that $\lambda < c_i \mu_i$ for each i. Since the output process of an $M/M/c$ system is also Poisson, it follows that production stage i can be modelled as an $M/M/c_i$ with arrival rate λ and service rate μ_i.

12.4 Asynchronuous lines with general processing times

We now consider an asynchronuous production line with m machines in series and generally distributed processing times. The mean processing time at machine i is $E(B_i)$ and the squared coefficient of variation is $c^2_{B_i}$. Jobs arrive according to a stream with generally distributed independent interarrival times with mean $1/\lambda$ and squared coefficient of variation c^2_A. For stability we have to assume that $\rho_i = \lambda E(B_i) < 1$ for each machine i.

In the exponential case we could exactly model each production stage as an $M/M/1$ system. In the general case we will use an approximation by modelling each production stage as a $G/G/1$ system. The arrival process at production stage i can be approximated as follows. The arrival rate at stage i is λ (by conservation of flow), and since arrivals at stage i are departures from stage $i-1$, we may approximate the squared coefficient of variation $c^2_{A_i}$ of the interarrival times at stage i by

$$c^2_{A_i} = (1-\rho^2_{i-1})c^2_{A_{i-1}} + \rho^2_{i-1}c^2_{B_{i-1}}, \qquad i = 2,3,\ldots,m.$$

Of course, at machine 1 we have $c^2_{A_1} = c^2_A$. Denoting the production lead time at stage i by the random variable S_i, we get as approximation

$$E(S_i) = \frac{\rho_i}{1-\rho_i} \cdot \frac{c^2_{A_i} + c^2_{B_i}}{2} \cdot E(B_i) + E(B_i), \qquad i = 1,2,\ldots,m, \tag{2}$$

and $E(L_i)$ may be obtained by application of Little's law.

Let us suppose that the line is *balanced*, so $E(B_1) = \cdots = E(B_m)$. But the variation in the processing times may be different at the production stages. Let us also assume that the production stages may be rearranged in any order. Then, what is the best order of the production stages, i.e., which order minimizes the mean total production lead time? Based on approximation (2) for the production lead time, it can be shown that the machines with the best processing reliability should be placed at the beginning of the line (cf. [1]). More formally, if $(\pi_1, \pi_2, \ldots, \pi_m)$ is the order of the machines in the production line (so machine π_1 is the first one, machine π_2 the second and so on), then the order minimizing the mean total production lead time should satisfy

$$c^2_{B_{\pi_1}} \le c^2_{B_{\pi_2}} \le \cdots \le c^2_{B_{\pi_m}}.$$

In the following example we explore the impact of other machine orderings on the performance of the production line.

Example 12.6 We consider a line with 3 machines, numbered $0, 1, 2$. The inflow is Poisson with a rate of λ jobs per time unit. The mean processing time at each machine is 1 (so the line is balanced); the squared coefficient of variation of the processing time at machine i is i. So machine 0 has the most reliable processing times (i.e., deterministic), and machine 2 has the least reliable one. In table 4 we demonstrate the difference in performance by arranging the machines in increasing, respectively decreasing order of processing reliability.

Machine order			λ	$E(S)$
2	1	0	0.80	12.8
0	1	2	0.80	10
2	1	0	0.90	30
0	1	2	0.90	22.3
2	1	0	0.95	64.8
0	1	2	0.95	47.2

Table 4: Mean total production lead time as a function of the machine order

Remark 12.7 The (approximative) analysis of production lines with general interarrival times and general processing times, where each stage is performed by a group of parallel and identical machines, proceeds along the same line: then each machine group is modelled as a $G/G/c$.

12.5 Asynchronuous lines with finite buffers

We consider an asynchronuous production line with $m+1$ machines in series; the machines are numbered $0, 1, 2, \ldots, m$ (see figure 2). The processing times at machine i are exponentially distributed with parameter μ_i. In between machine $i-1$ and machine i there is a buffer of size $N_i - 1$, $i = 1, 2, \ldots, m$.

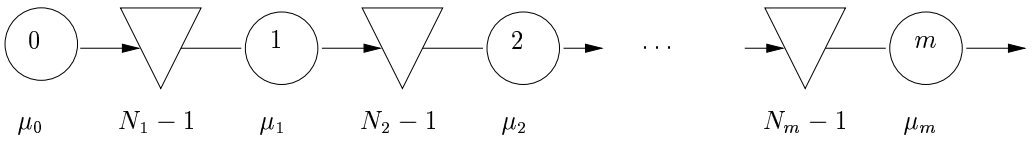

Figure 2: Production line with finite buffers and exponential processing times

Because the buffers are finite, machines may become blocked. We will consider two types of blocking:

- *Production blocking*: The machine will always process a job as long as there is one available. But when the machine has completed a job, it may have to wait to transfer the job until there is room in the downstream buffer.
- *Communication blocking*: The machine starts processing a job only if there is one available and there is room in the downstream buffer.

The first type of blocking is more common in production environments, The first type of blocking is more common in production environments, and it also referred to as *Blocking After Service*; the second type is *Blocking Before Service*. The arrival process of jobs is modelled differently from what we have seen for the infinite buffer system. In fact, machine 0 acts as arrival source; we assume that there are always new jobs (i.e., raw

material) available at machine 0, thus machine 0 will always process a job as long as it is not blocked. Relevant performance characteristics are, e.g., mean buffer levels, machine efficiency and the throughput of the line. In particular, to design production lines, it is important to know how the buffer sizes affect the throughput.

When the machines act according to the communication blocking protocol (the analysis of production blocking is similar), the system can be described by a Markov process with states $(n_1, n_2, \ldots, n_m)$ where n_m is the number of jobs at machine i (waiting or being processed); so $0 \leq n_i \leq N_i$. Hence the number of states of this Markov process is finite. The equilibrium probabilities $p(n_1, n_2, \ldots, n_m)$ can be solved (numerically) from the (finitely many) equilibrium equations, and performance characteristics such as the mean number of jobs at machine i and the throughput TH of the production line can be expressed in terms of these probabilities, i.e.,

$$E(L_i) = \sum_{(n_1,\ldots,n_m)} n_i p(n_1, \ldots, n_m), \qquad TH = \sum_{(n_1,\ldots,n_m):n_m>0} p(n_1, \ldots, n_m)\mu_m.$$

Note that, by conservation of flow, the throughput of each machine in the line is the same. The number of states of the Markov process is equal to

$$\Pi_{i=1}^{m}(N_i + 1).$$

Thus, although finite, the number of states may be very large for already small values of the buffer sizes N_i, so numerical solution of the equilibrium equations will be unpractical. Therefore we usually look for efficient approximations for estimating performance characteristics such as, e.g., the throughput of the line. There is a rich literature available on approximations for production lines with finite buffers; see, e.g., [3, 2, 4]. To illustrate some of the ideas we briefly describe a simple method for approximating the throughput of a three machine line.

Let us consider a production line with three machines, operating under the communication blocking protocol. To develop an approximation for the throughput we first *decompose* the line into two submodels; see figure 3. In the first submodel we replace the downstream machines 1 and 2 by a single *aggregate* machine; label this machine by d and let μ_d be its processing rate. The rate μ_d should be determined such that machine d properly describes the behavior of the machines 1 and 2. In the second submodel we aggregate the upstream machines 0 and 1 into a single machine, labeled by u and with processing rate μ_u. The throughput of each submodel may serve as an approximation for the throughput of the three machine line. Since the submodels consist of only two machines, their throughput is easy to determine (once the rates μ_u and μ_d are known).

It remains to determine the processing rates μ_d and μ_u. For these rates we have the following equations:

$$\frac{1}{\mu_d} = P(B) \cdot \frac{1}{\mu_2} + \frac{1}{\mu_1}, \tag{3}$$

$$\frac{1}{\mu_u} = P(S) \cdot \frac{1}{\mu_0} + \frac{1}{\mu_1}, \tag{4}$$

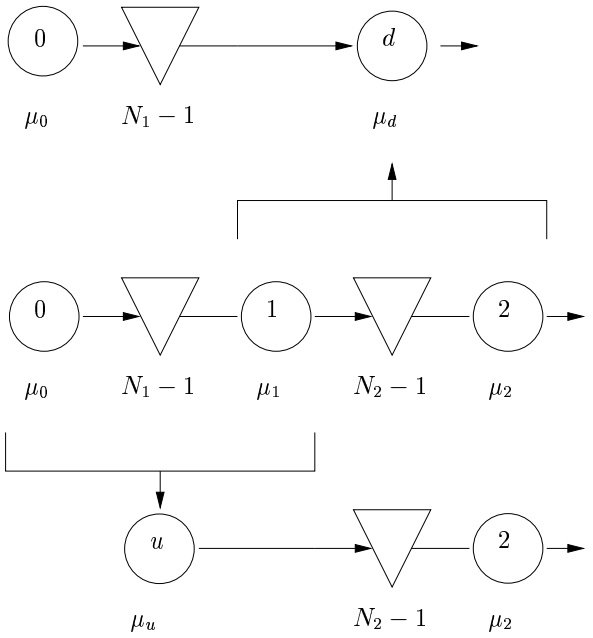

Figure 3: Decomposition of 3 machine line into two submodels, one with a *downstream* machine and one with an *upstream* machine

where $P(B)$ is the probability that machine 1 is blocked after job completion, and $P(S)$ is the probability that machine 1 is starved after job completion. The probability $P(B)$ can be estimated from the submodel with machine u.

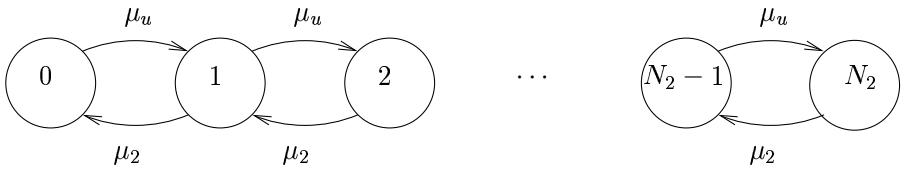

Figure 4: Flow diagram of the model with the upstream machine

Let p_k denote the probability there are k jobs in this submodel. From the flow diagram in figure 4 it is easily seen that

$$p_k = \left(\frac{\mu_u}{\mu_2}\right)^k p_0, \qquad k = 0, 1, \ldots, N_2.$$

Since $P(B)$ is equal to the probability that machine u is blocked after completion of a job, we get

$$P(B) = \frac{p_{N_2-1}\mu_u}{p_0\mu_u + p_1\mu_u + \cdots + p_{N_2-1}\mu_u} = \left(\frac{\mu_u}{\mu_2}\right)^{N_2-1} \frac{1 - \mu_u/\mu_2}{1 - (\mu_u/\mu_2)^{N_2}}.$$

Similarly, $P(S)$ follows from the submodel with machine d, yielding

$$P(S) = \frac{1 - \mu_0/\mu_d}{1 - (\mu_0/\mu_d)^{N_1}}.$$

Hence, the equations (3)-(4) form two (nonlinear) equations for μ_d and μ_u. These equations may be solved by successive substitutions starting with $\mu_d = \mu_u = \mu_1$. It can be shown that for the solution of (3)-(4), the throughtputs of the two submodels are exactly the same. Thus either throughput may be used as an approximation for the throughput of the three machine line.

12.6 Production line with closed-loop material handling

In this section we consider a production line with m machines in series; see figure 5. The machines are numbered $1, 2, \ldots, m$. The processing times at machine i are exponential with parameter μ_i and in front of each machine there is a buffer of size $N_i - 1$. Jobs are circulating on *pallets*; they can keep jobs in a fixed orientation (required for high precision operations) and make jobs easier to handle for transportation. As soon as a job is finished at (the last) machine m, it is removed from the pallet and a new job (raw part) is immediately placed on the pallet, after which it returns to machine 1. The number of circulating pallets is n. Clearly, the number of circulating pallets affects the throughput of the production line. If this number is small (large), we expect the throughput will be low (high). Below we investigate, for a simple two machine line, whether increasing the number of pallets indeed leads to higher throughput.

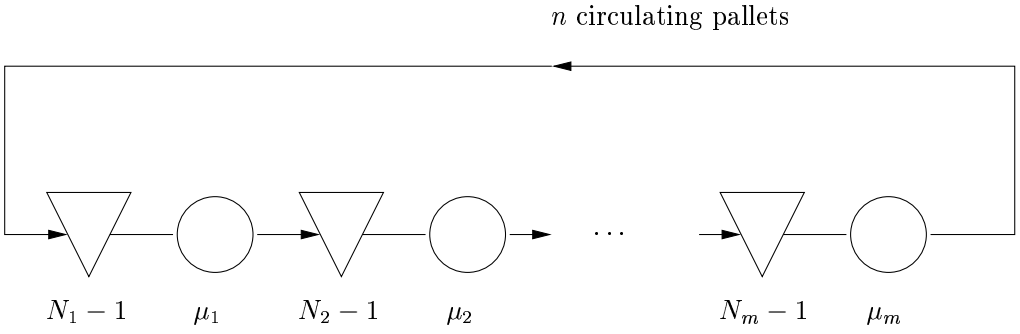

Figure 5: Production line with closed loop material handling

Let us look at a system with two machines, operating under the communication blocking protocol. Without loss of generality we assume that $N_1 \geq N_2$. The system can be described by a Markov process with states k, where k is the number of jobs at machine 1 (waiting in the buffer or being processed). Let p_k be the equilibrium probability of state k. To determine these probabilities we distinguish between several cases. If $n \leq N_2$, then there is no blocking at all; the flow diagram is shown in figure 6.

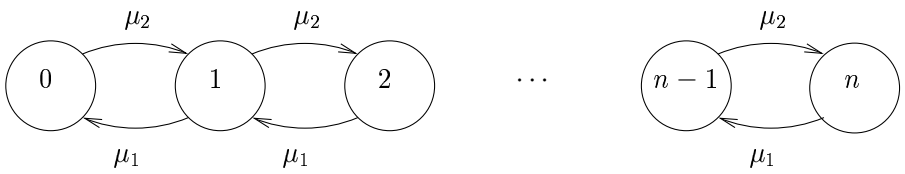

Figure 6: Flowdiagram for the two machine closed loop system with $n \leq N_2$

It readily follows that

$$p_k = \left(\frac{\mu_2}{\mu_1}\right)^k \frac{1-\mu_2/\mu_1}{1-(\mu_2/\mu_1)^{n+1}}, \qquad k = 0, 1, \ldots, n.$$

If $N_2 < n \leq N_1$, then machine 1 may be blocked because the buffer of machine 2 is full. The possible states are $n - N_2, n - N_2 + 1, \ldots, n$ and the probabilities satisfy

$$p_k = \left(\frac{\mu_2}{\mu_1}\right)^{k-(n+N_2)} \frac{1-\mu_2/\mu_1}{1-(\mu_2/\mu_1)^{N_2+1}}, \qquad k = n - N_2, n - N_2 + 1, \ldots, n.$$

Finally, if $N_1 < n \leq N_1 + N_2$, then the possible states are $n - N_2, n - N_2 + 1, \ldots, N_1$ and the probabilities are given by

$$p_k = \left(\frac{\mu_2}{\mu_1}\right)^{k-(n-N_2)} \frac{1-\mu_2/\mu_1}{1-(\mu_2/\mu_1)^{N_1-(n-N_2)+1}}, \qquad k = n - N_2, n - N_2 + 1, \ldots, N_1.$$

From the equilibrium probabilities p_k we can obtain the throughput of the line. Let $TH(n)$ denote the throughput for n circulating pallets. It then follows that

$$TH(n) = \begin{cases} (1-p_0)\mu_1, & n \leq N_2; \\ (1-p_{n-N_2})\mu_1 & n > N_2. \end{cases}$$

Note that the throughput is *symmetric*: $TH(n) = TH(N_1 + N_2 - n)$. In figure 7 we show the throughput $TH(n)$ as a function of n for a system $\mu_1 = 1$, $\mu_2 = 1.1$, $N_1 = 12$ and $N_2 = 8$. Observe that the throughput increases as we increase the number of pallets up until N_2; then it remains constant as long as n is between N_2 and N_1 and beyond this point the throughput decreases. Eventually, for $n = N_1 + N_2$ the throughput is 0; that is, the system will come to a *deadlock*.

References

[1] J.A. Buzacott, J.G. Shanthikumar, *Stochastic models of manufacturing systems*, Prentice Hall, Englewood Cliffs, 1993.

[2] Y. Dallery and S.B. Gershwin, *Manufacturing Flow Line Systems: A Review of Models and Analytical Results*, Queueing Systems Theory and Applications, 12 (1992), pp. 3–94.

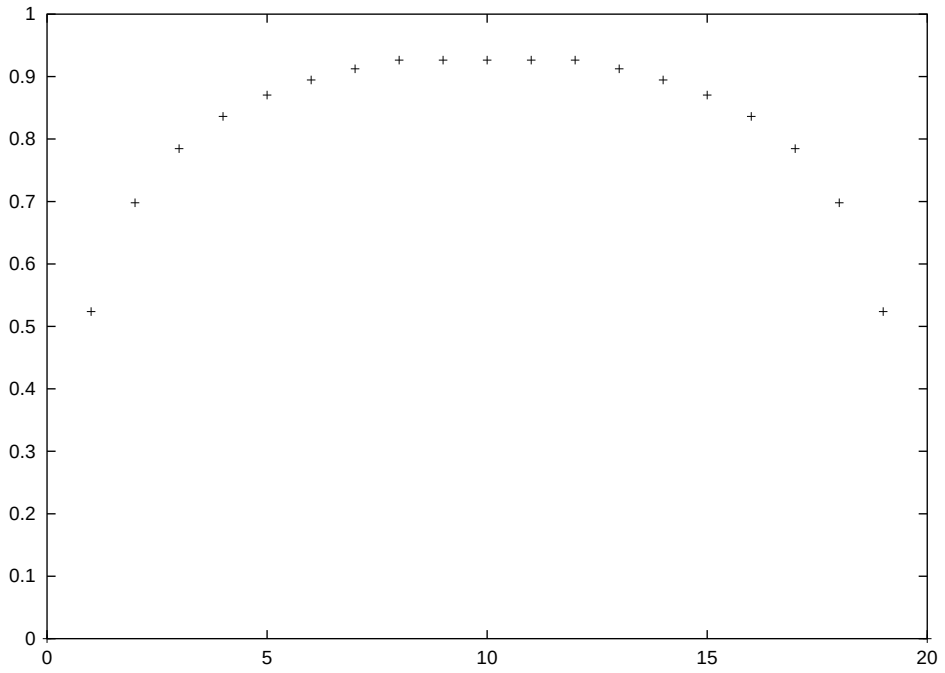

Figure 7: The throughput $TH(n)$ as a function of n for $\mu_1 = 1$, $\mu_2 = 1.1$, $N_1 = 12$ and $N_2 = 8$

[3] S.B. Gershwin, *An efficient decomposition method for the approximate evaluation of tandem queues with finite storage and blocking.* Oper. Res., 35 (1987), pp. 291–305.

[4] *Performance evaluation and optimization of production lines.* Papers from the International Workshop held on Samos Island, 1997. Eds. J. MacGregor Smith, S.B. gershwin and C.T. Papadopoulos. Ann. Oper. Res., 93 (2000), pp. 1–448.