8 The $M/G/1$ priority system

Usually not all jobs have the same urgency. Some jobs are supposed to be ready within a day or a week, while other jobs have a delivery date of 4 to 6 weeks from now. Further some customers are regular ones with contracts specifying short throughput times, others are occasional and receive a delivery date according to the present amount of work in process.

This makes it quite natural to study a simplified production system in which arriving jobs belong to different job classes and these job classes have different throughput time requirements. To further simplify this we say that the job classes have different priorities. If we number the priority classes from 1 upto r, then class 1 is top priority, class 2 has the second highest priority, etc. Further the job classes may have different processing time characteristics. We denote the processing time of class i by B_i, with mean $E(B_i)$ and mean residual processing time $E(R_i)$ with $E(R_i) = E(B_i^2)/2E(B_i)$. Class i jobs arrive according to a Poisson process with rate λ_i.

We will consider two variants of the priority rule. In the first one a job that has started cannot be interrupted; in the second one the processing of a job can be interrupted by newly arrived jobs of higher priority classes. If all higher priority jobs are served, the servicing of the job is resumed where it was preempted, i.e., no work is lost. The first type of priority is called *non-preemptive*, the second type is called *preemptive-resume*. If we think of the situation in which all production is done on one machine, non-preemptive priorities are far more natural, since an interrupt might lead to extra setup time or even to destruction of the product. If however the production capacity is mainly labour, then switching from one job to another might be fairly easy.

8.1 The non-preemptive priority system

The analysis is again based on the mean value approach, and it is a more or less straightforward extension of what we have seen for the $M/G/1$ system in chapter 7. For class i jobs, the quantities of interest are the mean waiting time $E(W_i)$, the mean number of jobs waiting in the queue $E(L_i^q)$, the mean throughput time $E(S_i)$ and the mean number of jobs in the system $E(L_i)$. Let us denote by $\rho_i = \lambda_i E(B_i)$ the utilization by class i jobs. Then, according to the PASTA property, an arriving job finds with probability ρ_i a class i job in service. Further, upon arrival the job finds on the average $E(L_i^q)$ jobs of class i in the queue.

Let us first look at a job of class 1. This job has to wait for jobs of its own class that arrived before, and also on the job (if any) on the machine. So,

$$E(W_1) = E(L_1^q)E(B_1) + \sum_{i=1}^{r} \rho_i E(R_i).$$

Defining,

$$\rho = \sum_{i=1}^{r} \rho_i$$

and

$$E(R) = \sum_{i=1}^{r} \frac{\rho_i}{\rho} E(R_i),$$

this becomes

$$E(W_1) = E(L_1^q)E(B_1) + \rho E(R), \tag{1}$$

where the term $\rho E(R)$ can be interpreted as the expected remaining amount of work currently present at the machine. Further, we have Little's formula again, stating

$$E(L_1^q) = \lambda_1 E(W_1).$$

Combining (1) and (8.1) yields

$$E(W_1) = \frac{\rho E(R)}{1 - \rho_1}.$$

Thus

$$E(S_1) = \frac{\rho E(R)}{1 - \rho_1} + E(B_1), \qquad E(L_1^q) = \frac{\lambda_1 \rho E(R)}{1 - \rho_1}, \qquad E(L_1) = \frac{\lambda_1 \rho E(R)}{1 - \rho_1} + \rho_1.$$

For the job classes $i = 2, ..., r$ the situation is more complicated. Apart from the amount of work found upon arrival that a job has to wait for, a job also has to wait for higher priority jobs that arrive later while it is waiting in the queue. Now let us consider a job of class i. According to the reasoning above we get intuitively

$$E(W_i) = \sum_{j=1}^{i} E(L_j^q)E(B_j) + \rho E(R) + E(W_i) \sum_{j=1}^{i-1} \rho_j.$$

One may formally show the correctness of the third term, but we will not do this here; it is the amount of higher priority work that arrives while the job is waiting. Moving the third term to the right we have

$$E(W_i)(1 - \sum_{j=1}^{i-1} \rho_j) = \sum_{j=1}^{i} E(L_j^q)E(B_j) + \rho E(R). \tag{2}$$

Using Little's law

$$E(L_i^q) = \lambda_i E(W_i),$$

we can rewrite (2) as

$$E(W_i)(1 - \sum_{j=1}^{i} \rho_j) = \sum_{j=1}^{i-1} E(L_j^q)E(B_j) + \rho E(R).$$

The right-hand side in this expression is the one we also get if we are computing $E(W_{i-1})$. Thus

$$E(W_i)(1 - \sum_{j=1}^{i} \rho_j) = \sum_{j=1}^{i-1} E(L_j^q)E(B_j) + \rho E(R) = E(W_{i-1})(1 - \sum_{j=1}^{i-2} \rho_j).$$

From this and the expression for $E(W_1)$ we easily derive recursively

$$E(W_i) = \frac{\rho E(R)}{(1-\sum_{j=1}^{i}\rho_j)(1-\sum_{j=1}^{i-1}\rho_j)}, \qquad i = 1, 2, \ldots, r.$$

From this we directly find $E(S_i)$, $E(L_i^q)$ and $E(L_i)$ using

$$\begin{aligned} E(S_i) &= E(W_i) + E(B_i), \\ E(L_i^q) &= \lambda_i E(W_i) \end{aligned}$$

and

$$E(L_i) = E(L_i^q) + \rho_i.$$

8.2 The non-preemptive SPT rule

One of the simplest priority rules is the so-called SPT rule, where the next job to be processed is always the one with the Shortest Processing Time (SPT). It will be clear that if we have two jobs, job 1 with a processing time of a and job 2 with a processing time of b, with $b > a$, then the mean waiting time of the two jobs will be smaller if we process job 1 first. If we use this SPT rule in a production-to-order system, then there will be a gain in mean waiting time when we compare it to the FCFS rule. We want to investigate here is whether this difference is substantial.

The approach of the previous section can be easily used to treat the SPT system. Let us consider the situation where the processing times are continuously distributed with density $f_B(\cdot)$ and let $E(W(x))$ be the mean waiting time for a job with processing time x. For this job there are essentially only three classes. Class 1 is the set of higher priority jobs, so jobs with a processing time smaller than x, class 2 are the jobs with a processing time of exactly x and class 3 are jobs with processing times larger than x. So we can directly apply the expression for the mean waiting time of the previous section, for which we now need the quantities $E(R)$, ρ_1, ρ_2 and ρ_3.

Of course, for the mean residual service time it is irrelevant in which order the jobs are processed, so $E(R)$ is the same as in the FCFS system. Furthermore, the computation of ρ_i is also straightforward, yielding

$$\rho_1 = \lambda \int_0^x y f_B(y) dy, \qquad \rho_2 = 0, \qquad \rho_3 = \lambda \int_x^\infty y f_B(y) dy = \rho - \rho_1.$$

Here, $\rho_2 = 0$ follows from the fact that the processing times are continuously distributed, and hence there are no jobs of length exactly x (except, of course, the one job we are interested in). This leads to

$$E(W(x)) = E(W_2) = \frac{\rho E(R)}{(1-\rho_1)(1-\rho_1-\rho_2)} = \frac{\rho E(R)}{(1-\lambda\int_0^x y f_B(y)dy)^2}.$$

Hence, the overall mean waiting time $E(W)$ is given by

$$E(W) = \int_{x=0}^\infty E(W(x)) f_B(x) dx = \int_{x=0}^\infty \frac{\rho E(R)}{(1-\lambda\int_0^x y f_B(y)dy)^2} f_B(x) dx.$$

The results of the SPT rule for the $M/M/1$ system are displayed in the following table, where we assume that the mean processing time is always equal to 1.

x	$\rho = 0.5$	$\rho = 0.8$	$\rho = 0.9$	$\rho = 0.95$
0	0.50	0.80	0.90	0.95
1	0.66	1.29	1.55	1.69
2	1.01	2.90	4.16	5.00
3	1.39	6.20	11.54	16.60
4	1.68	10.71	27.05	50.62
5	1.85	14.82	48.38	121.55
10	2.00	19.92	89.20	372.89
20	2.00	20.00	90.00	380.00
$E(W)$	0.71	1.88	3.20	5.26

Table 1: Mean waiting times for the $M/M/1$ system with the non-preemptive SPT rule

From the table we see that the overall mean waiting time $E(W)$ is considerably smaller than in the FCFS case, for which $E(W)$ would be equal to 1, 4, 9 and 19 for $\rho = 0.5$, 0.8, 0.9 and 0.95 respectively. Also note that for $\rho = 0.95$ all jobs with a processing time less than 3 benefit from the SPT rule, that is, more than 95 percent of the jobs will have a reduced waiting time.

8.3 The preemptive-resume priority system

If jobs are preempted, everything becomes more complicated. We no longer have at most one residual processing time, but there may be several, although at most one per job class. This makes it difficult to follow the reasoning we used in the non-preemptive case. However, there is an easy way out by concentrating not on jobs, but on work in the system and noting that the performance of a job of class i is completely unaffected by the jobs of the classes $j > i$. The system only works on these jobs if there are no class i or higher priority jobs. So if we try to derive the performance of a class i job we can act as if $\lambda_j = 0$ for $j > i$. Now note that the mean time in the system, $E(S_i)$, for a class i job consists of three parts:

$$E(S_i) = E(W_i) + E(B_i) + E(I_i). \tag{3}$$

The first term, $E(W_i)$, is the mean time until the job goes into production for the first time. The second term, $E(B_i)$ is the job's own mean processing time and the third term, $E(I_i)$, is the mean total interruption time during its processing time.

So let us look at a system in which the classes $i + 1$ upto r do not exist. And let us compare the waiting time in this system for a lowest priority job, i.e. a class i job, in the non-preemptive and preemptive-resume systems. First note that the total amount of work in the system is not affected by any of these two priority rules. The rules are so-called *work conserving*, i.e., no capacity is lost. So, although the order in which jobs are treated

is different from the non-preemptive case, the total amount of work found upon arrival is the same. This initial waiting time grows, due to higher priority class arrivals, to the total waiting time in the non-preemptive system. Thus

$$E(W_i) = \frac{\sum_{j=1}^{i} \rho_j E(R_j)}{(1 - \sum_{j=1}^{i} \rho_j)(1 - \sum_{j=1}^{i-1} \rho_j)}.$$

The second term in (3) is the mean processing time $E(B_i)$ and the third term is the sum of the interruptions. The total interruption time consists of two parts, the sum of the processing times of the jobs that interrupt the job we are looking at, and the sum of processing times of the jobs that arrive during periods in which the job is already interrupted. Hence, we get

$$E(I_i) = E(B_i) \sum_{j=1}^{i-1} \lambda_j E(B_j) + E(I_i) \sum_{j=1}^{i-1} \lambda_j E(B_j).$$

So the second and third term in (3), the mean processing time and the mean total interruption time, add up to what one might call the *generalized processing time*

$$E(B_i) + E(I_i) = \frac{E(B_i)}{(1 - \sum_{j=1}^{i-1} \rho_j)}.$$

So the total throughput time becomes

$$E(S_i) = \frac{\sum_{j=1}^{i} \rho_j E(R_j)}{(1 - \sum_{j=1}^{i-1} \rho_j)(1 - \sum_{j=1}^{i} \rho_j)} + \frac{E(B_i)}{(1 - \sum_{j=1}^{i-1} \rho_j)}.$$

8.4 A conservation law

In this section we consider a system processing r classes of jobs, each class arriving according to a general arrival stream. Jobs are processed in an order independent of their processing times and they may not be interrupted while they are processed. So, for example, jobs may be processed according to FCFS, random or a non-preemptive priority rule. Below we derive a conservation law for the mean waiting times of the r classes of jobs, which expresses that a weighted sum of these mean waiting times is independent of the processing discipline. This implies that an improvement in the mean waiting of one job class owing to a processing discipline will always degrade the mean waiting time of another job class.

Let $E(V(P))$ and $E(L_i^q(P))$ denote the mean amount of work in the system and the mean number of class i jobs waiting in the queue, respectively, for discipline P. The mean amount of work in the system is given by

$$E(V(P)) = \sum_{i=1}^{r} E(L_i^q(P))E(B_i) + \sum_{i=1}^{r} \rho_i E(R_i). \tag{4}$$

The first sum at the right-hand side is the mean amount of work in the queue, and the second one is the mean amount of work at the machine. Clearly the latter does not depend on the discipline P.

The important observation is that the amount of work in the system does not depend on the order in which the jobs are processed. The amount of work decreases with one unit per unit of time independent of the job being served and when a new job arrives the amount of work is increased by the processing time of the new job. Hence, the amount of work does not depend on P. Thus from equation (4) and Little's law

$$E(L_i^q) = \lambda_i E(W_i(P)),$$

we obtain the following conservation law for the mean waiting times,

$$\sum_{i=1}^{r} \rho_i E(W_i(P)) = \text{constant with respect to processing discipline } P.$$