2 Lecture 2: Scheduling and Queueing

- Scheduling theory
  - general setup
  - single machine scheduling
  - machines in parallel and in series
  - the job shop scheduling problem
- Queueing
  - single server queue
  - schedeuling the single server queue
  - single server queue in heavy traffic
- Appendix
  - fluid and diffusion approximation to single server queue in heavy traffic
  - Diffusion Processes, Brownian Motion and Reflected Brownian Motion

2.1 Scheduling Theory

I will only cover some very selected topics from scheduling theory, which are relevant to our type of manufacturting environment. It is not meant to address the whole theory and indeed may paint a distorted picture of the whole theory which is much richer, more sophisticated and far more practical then will be indicated here. In particular, the main feature I will stress in our context is that while optimal solution of our scheduling porblems is completely impractical, special features of our manufacturing systems allow us to approximate the optimal solution extremely well by very simple heuristics. This finesses most of the theory of scheduling.

2.1.1 General setup

Scheduling problems have a machine environment, job characterictics, and objective function. This classifies a rich collection of problems, under the 3-field notation:

1st Field describes the machine environment:

$1/$ / Single machine

$Pm/$ / Parallel machines

$Qm/$ / Uniform machines

$Rm/$ / Unrelated machines

$Om/$ / Open shop

$Fm/$ / Flowshop

Jm/ / Jobshop

2nd Field describes the Job Environment

$/X_j = 1$ / Unit processing times

$/r_j$ / Release dates

$/D_j$ / Deadlines

/prec / , /chain / , /intree / , /serpar / , Precedence constraints, of various types

/pmtn / Preemptions

3nd Field describes the Optimality Criteria

/ / $f_{\max}$ General Minmax

/ / $\sum f_j$ General Minsum

/ / $C_{\max}$ Makespan

/ / $\sum C_j$ Flowtime

/ / $\sum w_j C_j$ Weighted flowtime

/ / $L_{\max}$ Max lateness

/ / $T_{\max}$ Max tardiness

/ / $\sum T_j$ Sum of tardiness

/ / $\sum w_j T_j$ Weighted tardiness

/ / $\sum U_j$ Number of late jobs

Examples

(1) A 3 machine flowshop, minimize sum of completion times (flowtime), allowing preemptions: F3/pmtn/$\sum$Cj

(2) Schedule jobs with precedence constraints and release dates, on parallel machines (the number of machines is part of the data, and we need an algorithm to handle any number of them) so as to minimize the weighted sum of late jobs: P/r_j,prec/$\sum w_j U_j$

Within this classificaiton, Lawler, Lenstra, Rinnoy-Kan and Shmoys [] in their monumental survey consider a total of 4536 problems. Polynomial time algorithms were devised to solve 416 of these. 3582 of them were shown to be NP-hard.

Problems which are NP-hard cannot be solved efficiently, and so one needs to devise efficient algorithms to approximate the optimal solution. There are two ways to measure the performacnce of such heuristics.

In a worst case analysis one estimates the performance on the worst case problem, and ususally one proves results of the form

$$\max_{\text{all problem instances}} \frac{V^{heuristic}}{V^{Optimal}} \leq c$$

If we can get polynomial algorithms which achieve, at calculation time polynomial in $1/\epsilon$, a performance bound $c = 1+\epsilon$, we say we have a fully polynomial approximation scheme to solve the problem. Existense of such may also be an NP-hard problem (e.g. jobshop scheduling).

In a probabilistic analysis we assume that problems are drawn randomly from some problem population, e.g. assume that the processing times $X_1, \ldots, X_n$ of a problem instance are independently drawn from a distribution F. One then evaluates the performance of the heuristic by studying the distribution of the random variable $G = V^{heuristic} - V^{Optimal}$ or $R = \frac{V^{heuristic}}{V^{Optimal}}$.

Scheduling theorists (I believe also computer scientists) prefer by far to obtain worst case guarantees for their results. An even more pessimistic approach allows an adversary to choose the problem instance. I prefer the probabilistic approach. In particular, one gets a much more optimistic view of the algorithm. Often the behavior of an algorithm in practice refelects the probabilistic analysis rather than the worst case analysis.

In probabilistic analysis we can often show that an algorithm is asymptotically optimal, by showing that for problems of size N the performance ratio $R(N)$ satisfies:

$$E(R(N)) \to 1 \text{ or } P(R(N) > 1+\epsilon) \to 0 \text{ as } N \to \infty$$

or even more explicitly, find h, g such that:

$$P(R(N) > 1 + h(N)) \leq g(N) \text{ where } h(N), g(N) \to 0 \text{ as } N \to \infty.$$

2.1.2 Single machine scheduling

Minimize flowtime — S(E)PT

Perhaps the clearest statement of scheduling theory comes from scheduling a batch of N jobs to minimize the flowtime, i.e. the sum (or average) of the sojourn (or waiting) times of all the jobs. this is denoted $1/\ /\sum C_j$. Let $X_1, \ldots, X_N$ be the processing times (this is all the data here), and let $D_1, \ldots, D_n$ be their departure times (schedulers use the term completion times, hence the C_j notation, I will use D_j).

Proposition 2.1 *SPT - shortest processing time first minimizes the flowtime on a single machine.*

Proof. If jobs are started in the order $1, \ldots, n$ without idling, the flowtime can be expressed in three ways:

$$\begin{aligned} \sum_{j=1}^{n} D_j &= \sum_{j=1}^{n}\sum_{k=1}^{j} X_k && (2.1)\\ &= \sum_{j=1}^{n}(n-j+1)X_j && (2.2)\\ &= \sum_{j=1}^{n} X_j + \sum_{j=1}^{n}\sum_{k\neq j}\delta_{k,j}X_k && (2.3) \end{aligned}$$

Where

$$\delta_{k,j} = \begin{cases} 1 & \text{job } k \text{ starts before job } j \\ 0 & \text{otherwise} \end{cases}$$

this gives 3 proofs:

- Pairwise interchange of $j, j+1$ in (2.1) changes the objective by $X_j - X_{j+1}$, so any schedule can be improved by a sequence of such interchanges to get SPT.
- In (2.2) you pay at rates $n, n-1, \ldots$ when you work on 1st 2nd etc. job. The proposition follows from the Hardy-Littlewood-Polya inequality.
- In (2.3) for any pair of jobs j, k, exactly one waits for the other. SPT minimizes this wait simultaneously for all pairs by making the long job wait for the short job rather than the short wait for the long, i.e. we choose the shorter wait for every pair.

■

The result extends immediately to stochastic processing times: With EX_j the predicted processing times, SEPT (shortest expected processing time first) will minimize the expected flowtime.

How much do we gain from this policy? If we do not know anything about the jobs, or if we have to schedule them in a previously assigned order, in particular if we use an "on-line" schedule, how much do we lose? If we ask this question in a worst case scenario, we find that the worst case performance ratio R is unbounded.

We now perform a probabilistic analysis to gage how good SEPT is: Assume an underlying distribution F and assume that each problem instance is chosen as a random sample from it. Let $X_1, \ldots, X_N$ be the (random) processing times. We can schedule the jobs by SPT (we assume then that the values drawn for this instance are known to the scheduler), or we can use an "on-line" policy, where we assume the scheduler has no knowledge at all of the instance processing times. More generally we can assume that the scheduler has some data about the problem instance, in the form of some measurement T_j of the processing time X_j, and he uses the data to predict X_j, as $E(X_j|T_j)$. The scheduler then scuedules optimally according to SEPT. This is summarized by:

Proposition 2.2 *Assume job processing times X_j are drawn independently from a distribution F with mean m_1, variance σ_F^2, and information about them is summarized by i.i.d T_j. Let $Y_j = E(X_j|T_j)$ be the expected processing times, let G denote the distribution of Y_j (they are i.i.d.). Let $\rho = Corr(X_j, Y_j)$ be the correlation coefficient between X_j and Y_j. The expected value of the flowtime of a batch of n jobs, scheduled by SEPT, is*

$$E(\sum_{j=1}^{n} C_j |\, SEPT) = nm_1 + \frac{n(n-1)}{2} m_1 (1 - \rho \frac{\sigma_F}{m_1} d_G) \tag{2.4}$$

Here $m_1 = EX_j = EY_j$, $\sigma_F = \sigma(X_j)$, $\rho = \rho(X_j, Y_j)$, $\sigma_G = \sigma(Y_j) = \rho\sigma(X_j)$, $m_{1:2}^G = E\min\{Y_1, Y_2\}$ and $d_G = \frac{m_1 - m_{1:2}^G}{\sigma_G}$. Typically $d_F \approx d_G \approx 1/2$.

The probabilistic analysis shows a performance ratio:

$$R = \frac{E\sum D_j|SEPT}{E\sum D_j|RAND} = 1 - \rho\,\frac{\sigma}{m}\,d$$

Minimize weighted flowtime — Smith's ($c\mu$) Rule

Proposition 2.3 *To minimize weighted flowtime, $E\sum W_j D_j$, schedule jobs with highest $E(W_j)/E(X_j)$ first.*

Same 3 proofs work, and we also have the picture (figure 1). In fact, the picture suggests

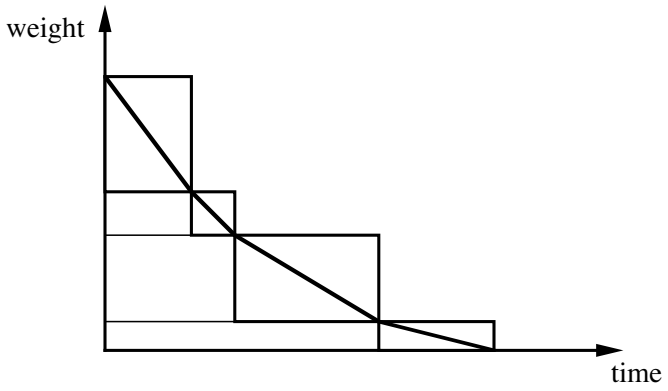

Figure 1: Minimizing weighted flowtime with Smith's rule

a slightly modified objective which will reappear later in the course: *Sum of weighted half completion times* $E\sum W_j(D_j - \frac{1}{2}X_j)$. (Note, the name $c\mu$ refers to cost c and processing rate μ).

Note: If the weights are proportional to the processing times, all schedules have the same weighted flowtime. This is often the case (approximately) in manufacturing.

Scheduling for due date adherence

Here is another pearl of scheduling theory: Let h_j be a monotone non-decreasing cost function for job j. We wish to minimize $h_{\max} = \max h_j(D_j)$ subject to some precedence constraints (e.g. job k needs to complete before start of job l). For this problem, 1/prec $/h_{\max}$, Lawler's algorithm is:

- **Initialize:** scheduled jobs: $J = \emptyset$, unscheduled jobs $J^C = \{1, \ldots, n\}$, unscheduled jobs with no successor: $J' \subseteq J^C$.
- **Calculate:** $j^* = \arg\max\{h_j(\sum_{k\in J^c} X_k) : j \in J'\}$. Move j^* form J^C to J.
- **Terminate:** When $J = \{1, \ldots, N\}$ schedule jobs in the opposite order of their entrance to J.

For the special case of minimizing the maximum lateness, $h_j = L_j = D_j - d_j$ where d_j is the due date of job j, $1/\text{prec}\ /\max L_j$:

Proposition 2.4 *To minimize maximum lateness,* $\max D_j - d_j$*, use Jacksons's rule, schedule jobs with earliest due date first (EDD).*

NP hard problems

While single machine scheduling exhibits a large number of polynomially solvable problems, still more problems, including some seemingly simple problems are NP-hard. For example, both minimum flowtime and minimum maximal lateness become NP-hard when one includes non-zero arrivas: $1/r_j\ /\sum C_j$ and $1/r_j\ /\max L_j$ are NP-hard.

2.1.3 Parallel machine scheduling

Makespan - how hard is NP-hard?

Consider positive integers $a_1, \ldots, a_N$, can you partition them into S, S^C, such that $\sum_S a_j = \sum_{S^C} a_j$? This is called the 2-partition problem. It can be solved in 2^N steps, but is (binary) NP-complete, so we do not know how to answer it efficiently. Hence scheduling a set of jobs on two machines to minimize makespan is (binary) NP-hard (3-partition is unary NP-hard).

However: (i) the worst schedule you could use has $R \leq 3/2$, (ii) if you use longest processing time first you get $R \leq 13/12$, (iii) and Karp and Karmarkar found a fully polynomial ϵ-approximation scheme for it.

Curiously, in the stochastic version of Pm/ $/C_{\max}$, if processing times are exponentially distributed, LEPT minimizes $EC_{\max}$.

But more important, a probabilistic analysis of this problem (rather than worst case) shows it is trivial: Consider scheduling a set of jobs with processing times drawn as a sample $X_1, \ldots, X_N$ from some distribution F. Just schedule them in any order, this is "on-line" scheduling.

Proposition 2.5 *Let* $T^* = \sum X_j/m$*. Let* T^{Opt} *and* T^{Rand} *denote the (random) oprimal makespan, and the makeapan obtained from an online arbitrary schedule. Then* $T^* \leq T^{Opt} \leq T^{Rand} = T^* + D$ *where* D *is the maximum of* $m-1$ *forward recurrecne times.*

Hence: For small N we can solve this problem optimally, for moderate N we have fully polynomial approximation schemes, and for large N we can schedule on-line and be almost optimal.

Flowtime - how complicated do you want to be?

Proposition 2.6 *SPT is optimal for Pm/* $/\sum C_j$.

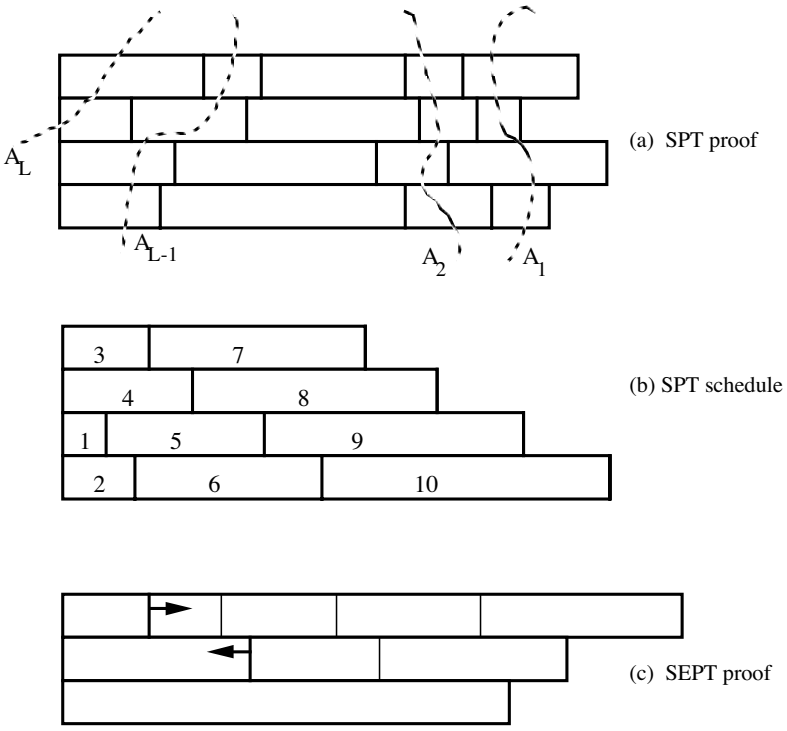

Figure 2: Minimizing flowtime on parallel machines

Proof. We rewrite (2.2). Let A_l, $l = 1, \ldots, L \leq N$ be the set of jobs which are scheduled in position l from the end on their machine. Then

$$\sum D_j = \sum_{l=1}^{L} \sum_{j \in A_l} l X_j,$$

and so it is optimal to place the m longest jobs last on their machines, the next longest m jobs one before last etc. But this is SPT. Figure 2(a,b) tells the story. ■

When processing times are not known to the scheduler but are random X_j SEPT is not always optimal as some simple though somewhat contrived examples show. However, if the distributions of processing times of jobs can be stochastically compared ($X \geq_{ST} Y \Leftrightarrow \mathbb{P}(X > x) \geq \mathbb{P}(Y > x)$ for all x), then Weber, Varaiya and Walrand [] showed that SEPT minimizes flowtime (in distribution, not just in expectation). Figure 2(c) indicates their basic idea.

Weighted flowtime and Smith's rule

Minimizing weighted flowtime on parallel machines, Pm/ $/\sum w_j C_j$, is NP-hard. However, Smith's rule works quite well for such problems. In fact, it seems that Smith rule is optimal except for some end effects at the end of the schedule, where less than m machines remain busy, see Figure 3

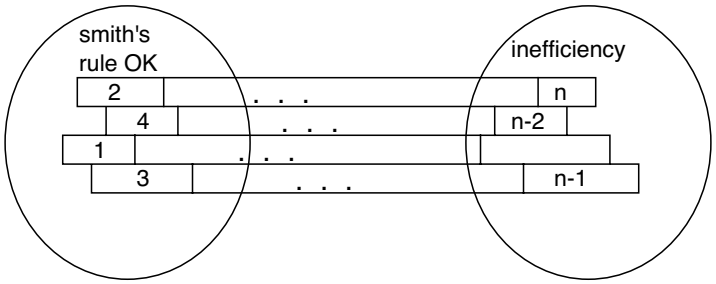

Figure 3: Applying Smith's rule to parallel machines

A worst case performance analysis shows that for Smith's rule $R < \frac{1}{2} + \sqrt{\frac{1}{2}} \approx 1.2$. This gap is only achieved by a very contrived example, consisting of many very short and few very long jobs, with weights proportional to job length (i.e. all schedules are optimal on a single machine). One can bound the end effect at the end of the schedule, for X_j deterministic or stochastic, and see that under most conditions the gap $V^{SmithRule} - V^{Opt}$ is small.

2.1.4 Flowshps — machines in series

Makespan of the two machine flowshop — Johnson's rule

The only general flowshop scheduling problem which is not NP-hard is F2/ $/C_{\max}$. For jobs $j = 1, \ldots, N$ let a_j, b_j be the processing times on machines 1 and 2 respectively. Johnson' rule says: Partition the jobs into two sets, according to $a_j \leq b_j$ and $a_j > b_j$, then schedule the first set according to SPT of machine 1, followed by the second set according to LPT of machine 2.

Proposition 2.7 *Johnson's rule minimizes makespan on the two mahcine flowshop.*

Proof. If jobs are scheduled in the order $1, \ldots, N$ the makespan on the two machine flowshop equals

$$C_{\max} = \max_k \sum_{j=1}^{k} a_j + \sum_{j=k}^{N} b_j$$

and the optimality of Johnson's rule follows by pairwise interchange. ■

It is obvious from the proof that the optimal schedule is far from unique. This is typical of the makespan objective — it is often the case that many very different schedules achieve the optimum, and furthermore, the objective is very flat around the optimum, so that an even larger set of schedules are nearly optimal (this can be regarded either as a weakness or as a strength, depending on your viewpoint).

In fact Johnson's rule is pretty sutpid. Here is its rationale: To minimize makespan you want to prevent machine 2 from idling (starvation). So you send it work as fast as you can and you and you keep it there for as long as you can, to get minimum makespan. In doing so you create a huge amount of WIP between the two machines, which is bad for flowtime and is

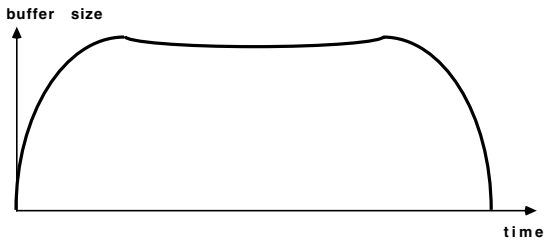

Figure 4: Evolution of WIP between machines in Johnson's rule

generally undesired in manufacturing. If jobs are chosen randomly, about 1/4 of the jobs will be in this WIP throughout most of the makespan, see Figure 4.

Assume jobs have processing times drawn independently, $a_j \sim A$, $b_j \sim B$, with means and standard deviations $m_A, m_B, \sigma_A, \sigma_B$. The machines are balanced if $m = m_A = m_B, \sigma = \sigma_A = \sigma_B$. The expected makespan in the balanced case is

$$\begin{aligned} EC_{\max}| \text{ Johnson's rule} &= mN + 0.56\sigma\sqrt{N} \\ EC_{\max}| \text{ Random order} &= mN + 1.12\sigma\sqrt{N} \end{aligned}$$

In the unbalanced case the makespan under Johnson's rule as well as under a random schedule will be:

$$EC_{\max} = \max\{m_A, m_B\}N + O(\log N).$$

This indicates that minimizing the makespan is not an issue in flowshops.

Minimizing makespan for more than 2 machines is NP-hard. For 4 or more machines the optimal order of jobs on successive machines may be different.

Minimizing flowtime or weighted flowtime is NP-hard for 2 or more machines. We shall re-examine flowshops as tandem queues in future lectures.

2.1.5 The job shop scheduling problem

A job shop consists of:

- Machines $i = 1, \ldots, I$
- Routes (production processes) $r = 1, \ldots, R$, where route r consists of steps $o = 1, \ldots, K_r$, denoted r, o, and each step requires one of the machines, $i = \sigma(r, o)$, and we let $C_i = \{r, o : \sigma(r, o) = i\}$.
- Jobs $j = 1, \ldots, N$, where each job has a release time $A(j)$, a route $r = r(j)$, and processing times along the route $X_{r,1}(j), \ldots, X_{r,K_r}(j)$.

Given this data, a schedule will consist of assigning start and finish times $s_{r,o}(j), t_{r,o}(j)$ to each operation of each job with the obvious constraints. The objective may be makespan, flowtime or weighted flowtime, or a due date related objective.

Job shop problems are of course NP-hard. Furthermore, even small instances are very hard to solve. Most of the extensive work on job shops in scheduling theory assumes makespan objective and 0 release times. A famous job shop scheduling problem is the 10×10 problem. This was randomly generated, with 10 jobs, each requiring 10 operations, ordered as a permutation of 10 machines (see Figure 5). This problem was taken on as a chellenge by the entire world wide scheduling community and took 10 years to solve. We shall study job-shop scheduling in detail, and refer back to the 10×10 example later in the course.

Step#	job	1	job	2	job	3	job	4	job	5
1	1	29	1	43	2	91	2	81	3	14
2	2	78	3	90	1	85	3	95	1	6
3	3	9	5	75	4	39	1	71	2	22
4	4	36	10	11	3	74	5	99	6	61
5	5	49	4	69	9	90	7	9	4	26
6	6	11	2	28	6	10	9	52	5	69
7	7	62	7	46	8	12	8	85	9	21
8	8	56	6	46	7	89	4	98	8	49
9	9	44	8	72	10	45	10	22	10	72
10	10	21	9	30	5	33	6	43	7	53

Step#	job	6	job	7	job	8	job	9	job	10
1	3	84	2	46	3	31	1	76	2	85
2	2	2	1	37	1	86	2	69	1	13
3	6	52	4	61	2	46	4	76	3	81
4	4	95	3	13	6	74	6	51	7	7
5	9	48	7	32	5	32	3	85	9	64
6	10	72	6	21	7	88	10	11	10	76
7	1	47	10	32	9	19	7	40	6	47
8	7	65	9	89	10	48	8	89	4	52
9	5	6	8	30	8	36	5	26	5	90
10	8	25	5	55	4	79	9	74	8	45

First Appeared in 1963 book of Muth and Thompson.
Solution found by Carlier and Pinson, 1988,
Solution is 930 Lower bound 631.

Figure 5: The 10×10 jobshop scheduling problem

2.2 Queueing Theory

We again just skim the theory. In particular we mainly get results about averages, and consider just a small selection of models.

2.2.1 The single server queue

Customers $j = 1, 2 \ldots$ arrive at a server at times $0 < A_1 < \cdots < A_j < \cdots$ and require from the server service times $X_1, \ldots, X_j, \ldots$. After waiting for service and after being served they

depart at times $D_1, \ldots, D_j, \ldots$. Let $\mathcal{A}(t), \mathcal{D}(t)$ denote the cumulative number of arrivals and departures in $(0, t]$, then the queue at time t is $Q(t) = \mathcal{A}(t) - \mathcal{D}(t)$.

Service of customers is according to a service policy, e.g. FIFO (FCFS), LIFO (LCFS), Priority, EDD, SPT, SRPT. We shall assume that the server is working whenever there is work, these are called (not a very good name) work conserving policies. $\mathcal{A}(\cdot), X_{(\cdot)}$, and the serivce policy determine the departure and queue processes. A good exercise at this point is to simulate such queing processes (say with a spreadsheet).

Sojourn of customer j is $W_j = D_j - A_j$, and it is composed of delay V_j and service X_j. We let $\mathcal{W}(t)$ denote the work in the system at time t, i.e. if arrivals are switched off, how long before the system is empty. Under work conserving policies, $\mathcal{W}(t)$ increases by X_j at A_j, and decreases at rate 1 whenever it is > 0. Hence, $\mathcal{W}(t)$ is invariant under all (work conserving) policies (hence the name?). $\mathcal{W}(t)$ is called the virtual work load process.

To study this as a system we need to make some assumptions on the ensemble of arrivals and service times. We shall assume at least existence of rates: Arrival rate $\lambda = \lim_{t\to\infty} \frac{\mathcal{A}(t)}{t}$, and processing rate $\mu = 1/m$ where $m = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} X_j$. More than that, we shall mostly assume that inter arrivals $A_j - A_{j-1}$ and processing times X_j form independent i.i.d. sequences.

Clearly the departure rate of the queue cannot exceed λ, μ and therfore Q can remain bounded only is $\lambda \leq \mu$. In that case, queueing is the result of variability in the arrival and service times: If arrivals occur at every hour on the clock, and service takes exactly 54 minutes then nobody ever waits, each arrival finds an empty system, even though the system is busy .9 of the time. The average queue length (averaged over a long time) is .9 customers. If on the other hand, with the same averages, arrivals are Poisson and services exponential, then the system will still be busy .9 of the time, but there will be 9 customers in the system on the average, and an arrival will find an average of 9 customers in the system.

Relations between sequences of values at a sequence of time points, and values over continuous times, in particular between time averages and customer averages, are very informative.

Exercise 2.1 Assume arrivals and departures occur singly. Show that the average number of customers in the system before and after an arrival and before or after a departure can be very different from the average number of customers in the system averaged over time. What relationships can you write between these?

Proposition 2.8 (Little's law) *Let λ be the arrival rate into a system, and assume that the sojourn times are regular enough to have $\bar{W} = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} W_j$. Then the number of customers in the system will have a long-term average $L = \lim_{T\to\infty} \frac{1}{T} \int_0^T Q(t)dt$, and $L = \lambda \bar{W}$.*

Proof. we have (see Figure 6)

$$\sum_{j=1}^{\mathcal{D}(T)} W_j \leq \int_0^T Q(t)dt \leq \sum_{j=1}^{\mathcal{A}(T)} W_j$$

Divide by T, and multiply and divide by $\mathcal{D}(t), \mathcal{A}(t)$ the l.h.s and r.h.s. to get

$$\frac{\mathcal{D}(T)}{T} \frac{1}{\mathcal{D}(T)} \sum_{j=1}^{\mathcal{D}(T)} W_j \leq \int_0^T Q(t)dt \leq \frac{\mathcal{A}(T)}{T} \frac{1}{\mathcal{A}(T)} \sum_{j=1}^{\mathcal{A}(T)} W_j,$$

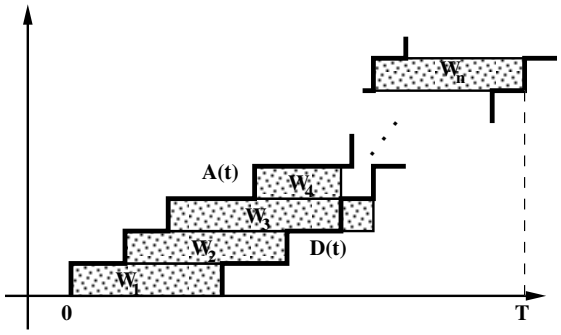

Figure 6: Little's law

and let $T \to \infty$. It follows from the existence of $\bar{W}$ that $\frac{\mathcal{D}(T)}{T}$ and $\frac{\mathcal{A}(T)}{T}$ both tend to λ and both r.h.s and l.h.s tend to $\lambda\bar{W}$. ■

A much subtler property of time and customer averages is PASTA

Proposition 2.9 (PASTA) *Let $Z(t)$ be a stochastic process, and $\Lambda(t)$ a Poisson process, with events at times $0 < T_1 < \cdots < T_j < \cdots$. Assume for all t that $\{\Lambda(s) - \Lambda(t) : s > t\}$ is independent of $\{Z(s), \Lambda(s) : s \le t\}$. Then the following (if they exist) are equal:*

$$\lim_{T\to\infty} \frac{1}{\Lambda(T)} \sum_{j=1}^{\Lambda(T)} Z(T_j) = \lim_{T\to\infty} \frac{1}{T} \int_0^T Z(t)dt$$

Here are 3 consequences of Little's law:

- The long term fraction of time the server is busy is $\rho = \frac{\lambda}{\mu}$.
- The long term average workload at the server is $\frac{1}{2}\lambda E(X^2)$
- The long term average virtual workload in the system is $\lambda\left(E(X)\bar{V} + \frac{1}{2}E(X^2)\right)$ where $\bar{V}$ is the average delay of a customer, before service.

Three single server models of interest are M/M/1, M/G/1, and GI/G/1, which we describe now, when we assume that $\rho < 1$. For ergodic arrival and service streams, $\rho < 1$ implies that the system will reach a steady state.

M/M/1 queue

If arrivals are Poisson and services exponential the queue length is a Markov process with countable state space, in fact a birth and death process. In steady state $P(Q(t) = x) = (1-\rho)\rho^x$, $x = 0, 1, \ldots$, and the sojourn time of a customer is $W_j \sim \exp(\mu - \lambda)$. The average queue length, average delay, and average sojourn time are: $L = \frac{\rho}{1-\rho}$, $\bar{V} = \frac{\rho}{\mu-\lambda}$. $\bar{W} = \frac{1}{\mu-\lambda}$.

M/G/1 queue

This system has an embedded Markov process at the times of service completions. By PASTA an arriving customer will on the average find the long term time average virtual workload in the system. If service is FIFO this gives an equation:

$$\bar{V} = \rho\bar{V} + \frac{1}{2}\lambda E(X^2) \tag{2.5}$$

from which we get:

$$\bar{V} = \frac{\frac{1}{2}\lambda E(X^2)}{1-\rho} = \frac{m}{1-\rho}\frac{1+c_s^2}{2}$$

where $c_s^2 = \frac{Var(X)}{E(X)^2}$ is the service time squared coefficient of variation (which is 1 for exponential service, but is usually much smaller, e.g. it is $1/3$ for $X \sim U(0,b)$, and ≈ 0.04 for $X \sim U(b,2b)$).

We note that any policy which cannot predict the processing times of jobs has the same average queue length and processing time (e.g. LIFO or EDD).

GI/G/1 queue

This has no convenient regeneration points (except starts of busy periods) but is positive Harris recurrent if we include remaining (or attained) processing time and remaining (or attained) interarrival time in the state description. There are no closed form formula for any GI/G/1 steady state characteristics. Indeed, quantities such as average queue length are functions of the whole service and interarrival distributions and are not expressible in terms of just a few moments. What is known (easy to derive) is:

$$\bar{V} \leq \frac{m}{1-\rho}\frac{c_a^2+c_s^2}{2}$$

and in heavy traffic (which is usually the case in manufacturing systems)

$$\bar{V} \approx \frac{m}{1-\rho}\frac{c_a^2+c_s^2}{2}, \quad \rho \approx 1$$

where we now have dependence on c_a^2, the interarrival time squared coefficient of variation

The dependence on ρ and explosive growth of congestion are depicted in figure 7.

busy period

In a process with Poisson arrivals the idle periods are $\sim \exp(\lambda)$, with expected duration $1/\lambda$. Since these alternate with busy periods, and the fraction of time that the server is busy is ρ, we have the equation

$$\frac{E(BP)}{E(BP)+1/\lambda} = \rho$$

from which we get:

$$E(BP) = \frac{m}{1-\rho}.$$

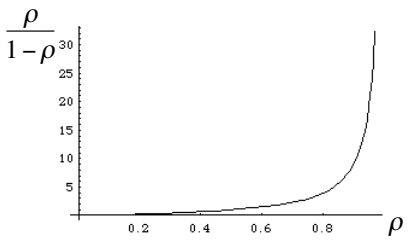

Figure 7: Growth of queue length, sojourn and busy period with ρ

If a busy period starts with a service of length x, we call it an exceptional first service busy period. At the completion of time x this allows $M \sim Poisson(\lambda\, x)$ arrival to be there, and the remainder of the busy period consists of M i.i.d ordinary busy periods for a total expected duration:

$$E(EFSBP) = \frac{x}{1-\rho}.$$

2.2.2 Scheduling the single server queue

The problem $1/r_j\,/\sum C_j$ is NP-hard, so for given arrival times and processing times, minimizing flowtime in the single server queue is NP-hard. The following results hold:

- If preemptions are allowed then SRPT, shortest remaining processing time first, will minimize the flowtime. The proof of this is beautiful but I will skip it (included in appendix?). Note that this policy is on-line: You need to examine your current queue, and determine the remaining processing times of all jobs present. It requires knowing the processing times exactly. However the optimal decision at time t is independent of jobs that have not yet arrived.

- If arrivals are Poisson, then using SPT or SEPT turns out to be optimal. This is Klimov's problem of control of multiclass M/G/1 queue. For weighted flowtime the $c\mu$ rule (Smith's rule) is optimal. There is a rich literature on this problem, starting with Cox's proof that SEPT is the best priority policy, on to Klimov's '76 paper, and solution using Gittins index, and more recently, achievable region approach and extension to show asymptotic optimality for parallel machines (in heavey traffic).

  We shall study priority scheduling presently.

- In heavy traffic, priority rules will cause a state space collapse: Virtually all of the queueing will be done by the customers of the lowest priority class.

  We shall study heavy traffic presently.

Note that to minimize flowtime in the case of arrivals the objective is to minimize the sum of the sojourn times, $\sum D_j - A_j$. If we do not control the arrival times, then this is equivalent to minimizing the sum of completion times, $\sum D_j$. However, if arrivals are regular (have a rate of arrival) then the sum of sojourn times of N jobs will be of order $O(N)$, while the sum of

departure times will be of order $O(N^2)$. Long term average only makes sense if we use sojourn time and not sum of completion times. In the finite horizon case we will also use the sojourn time, but a significant part of the jobs will be available right at the start.

M/G/1 queue under priority policies

Jobs are classified into classes $k = 1, \ldots, K$ with independent Poisson arrivals of rates λ_k and service rates μ_k; traffic intensities are $\rho_k = \frac{\lambda_k}{\mu_k}$, and they add up to $\rho = \sum_{k=1}^{K} \rho_k$.

We give priority to class 1 over 2 etc, over K. Under this priority policy, class k customers will satisfy (similar to 2.5) the equation:

$$\bar{V_k} = (\sum_{i=1}^{k} \rho_i \bar{V_i} + \lambda E(X^2)/2)/(1 - \sum_{i=1}^{k-1} \rho_i).$$

Here the wait of an arriving customer consists of the average work of his own or higher priority customers in the system (by PASTA) and since he will also need to wait for all higher priority customers that arrive while this original amount of work is processed, this average amount of work is blown up by a factor of $1/(1 - \sum_{i=1}^{k-1} \rho_i)$ (this is the formula for a busy period with exceptional first service). The expectation $E(X^2)$ is over the distribution of processing times of all customers. It is an easy induction to see that:

Proposition 2.10 *The average delay for a class k job, under a nonpreemptive priority policy is:*

$$\bar{V_k} = \frac{\lambda E(X^2)/2}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^{k} \rho_i)}$$

Similar formulas can be computed for the average delay in queue for preemptive priorities, and for non-preemptive SPT and preemptive SRPT, see e.g. Wolff [].

It is easy to see that SEPT will minimize the average sojourn time. In fact it is optimal among all possible policies, as shown by Klimov or Gittins. For weighted flowtime, with weight w_k for class k, the $c\mu$ priority rule will be optimal.

Multiclass GI/G/1 under heavy traffic

Under heavy traffic, the higher priority classes pass through the system much faster, so that most of the queueing is done by the lowest priority. Because the total virtual workload is the same under any policy, we have:

$$\frac{\text{Average sojourn| Priority Rule}}{\text{Average sojourn| No Priority}} = \frac{E(X_K)}{E(X)}.$$

2.3 Fluid and Diffusion Approximations to the Single Server Queue in Heavy Traffic

To analyze GI/G/1 in heavy traffic we consider a sequence of systems indexed by n, with initial states and parameters $Q^n(0), \lambda^n, \mu^n, c_A^2, c_S^2$, and let $n \to \infty$. We obtain results for scaled

versions of the queue length and busy time processes, $Q^n(t), B^n(t)$. The fluid scaling of a sequence $x^n(t)$ is $\bar{x}^n(t) = \frac{1}{n}x^n(nt)$. If $\bar{x}^n(t) \stackrel{a.s.}{\to} \bar{x}(t)$ then $\bar{x}(t)$ is the fluid limit. The diffusion scaling of a sequence $x^n(t)$, centered around its fluid limit, is $\hat{x}^n(t) = \frac{1}{\sqrt{n}}(x^n(nt) - \bar{x}(nt))$.

Fluid approximation

If $\frac{1}{n}Q^n(0) \to \bar{Q}(0)$, and $\lambda^n \to \lambda, \mu^n \to \mu$:

$$\begin{aligned}
\bar{Q}^n(t) &\stackrel{a.s.}{\to} \bar{Q}(t) = \max\{0, (\lambda - \mu)t\} \\
\bar{B}^n(t) &\stackrel{a.s.}{\to} \bar{B}(t) = \begin{cases} t & t \le \tau \\ \tau + \rho(t-\tau) & t \ge \tau \end{cases} \\
&\qquad \tau = \inf\{t : \bar{Q}(t) = 0\}
\end{aligned}$$

Diffusion approximation

We use diffusion scaling, $\bar{Q}^n(t) = \frac{1}{\sqrt{n}}Q^n(nt)$, $\bar{B}^n(t) = \frac{1}{n}B^n(nt)$.

We let $\frac{1}{\sqrt{n}}Q^n(0) \to \hat{Q}(0)$ (so that $\bar{Q}(0) = 0$). We also let $\lambda^n, \mu^n \to \lambda$, so in the limit $\rho = 1$. To capture the degree of closeness to $\rho = 1$ we let $\sqrt{n}(\lambda^n - \mu^n) \to \theta$.

We let $\mathcal{X}$ denote the so called netput process.

$$\begin{aligned}
\hat{\mathcal{X}}^n(t) &= \frac{1}{\sqrt{n}}\left[Q^n(0) + (\lambda^n - \mu^n)nt + (\mathcal{A}^n(nt) - \lambda^n nt) - (\mathcal{S}^n(B^n(nt)) - \mu^n B^n(nt))\right] \\
&\stackrel{w}{\to} \hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)
\end{aligned}$$

The queue length process and the busy time then satisfy:

$$\begin{aligned}
\hat{Q}^n(t) = \frac{1}{\sqrt{n}}Q^n(nt) &\stackrel{w}{\to} \Psi\left(\hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\right) \\
\hat{B}^n(t) = \frac{1}{\sqrt{n}}B^n(nt) - t &\stackrel{w}{\to} -\frac{1}{\lambda}\Phi\left(\hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\right)
\end{aligned}$$

where Ψ, Φ denote the reflection and regulator operations in the reflection mapping of $\mathcal{X}$.

Strong approximation

By the FSAT (when moments of order $r > 2$ exist), we can approximate the queue length by:

$$\tilde{Q}(t) = RBM_{Q(0)}(t, \theta, \lambda(c_A^2 + c_S^2)) = \Psi\left(Q(0) + \theta + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\right)$$

which is a reflected Brownian motion starting at $Q(0)$, with drift θ and diffusion coefficient $\lambda(c_A^2 + c_S^2)$.

For $\theta < 0$, this will reach a steady state distribution $\tilde{Q}(t) \sim \exp(\frac{-2\theta}{\lambda(c_A^2+c_S^2)}) = \exp(\frac{2(1-\rho)}{c_A^2+c_S^2})$ with mean $\frac{c_A^2+c_S^2}{2(1-\rho)}$

2.4 Maximal queue length for GI/G/1 queue

It is instructive to examine the maximal queue length attined in the service of the first n customers. This is of interest when we want to study the transient behavior of the queue. Since we want to optimize the system for a given initial state and for a finite horizon, this is exactly the type of result we need.

In the following we need two assumption, one is that the GI/G/1 queue is stable ($\rho < 1$), the other assumption is that the interarrival and service distributions possess exponential moments (Laplace transform exists in a neighbothood of 0). As a result of that, the maximal queue length is of order $O(\log n)$.

Lemma 2.11 *Consider a GI/GI/1 queue. Let $\{u_i, i \geq 1\}$ be iid inter-arrival times and $\{v_i, i \geq 1\}$ be iid service times. Assume that for some $\theta > 0$,*

$$\mathbb{E}[e^{\theta(u_1+v_1)}] < \infty \tag{2.6}$$

and $\mathbb{E}[u_1] > \mathbb{E}[v_1]$. Let

$$\tau_n = u_1 + \ldots + u_n$$

be the arrival time of the nth job. Let $Z(t)$ be the queue length (including possibly the one being serviced) at time t. For any $\kappa \geq 1$, there exists a constant $c > 0$ such that for all $n \geq 2$,

$$\mathbb{P}\left(\sup_{0\leq t\leq \tau_n} Z(t) > c\,\log n\right) \leq 1/(\kappa\, n).$$

In particular, for all $n \geq 2$,

$$\mathbb{P}\left(\sup_{0\leq t\leq \tau_i} Z(t) > c\,\log n\right) \leq 1/(\kappa\, n) \quad \textit{for any } 1 \leq i \leq n.$$

Proof. For each $\theta > 0$, set $f(\theta) \equiv \mathbb{E}[e^{\theta(v_1-u_1)}]$. Since $\mathbb{E}[u_1] > \mathbb{E}[v_1]$ and (2.6) holds, there exists $\theta = \theta_0 > 0$ such that $f(\theta_0) < 1$. (See, for example, Shwartz and Weiss [**?**, Exercise 1.3].)

We first claim that for any $m \geq 0$ and $l \geq 0$, and $k \geq 1$,

$$\mathbb{P}(u_m + \ldots + u_{m+l} + \ldots + u_{m+l+k} < v_m + \ldots + v_{m+l}) \leq (f(\theta_0))^l \mathbb{E}[e^{-\theta_0 u_1}]^k.$$

To see this, for any $\theta > 0$,

$$\begin{aligned}
&\mathbb{P}(u_m + \ldots + u_{m+l} + \ldots + u_{m+l+k} < v_m + \ldots + v_{m+l}) \\
&= \mathbb{P}\Big(v_m - u_m + \ldots + v_{m+l} - u_{m+l} - u_{m+l+1} - \ldots - u_{m+l+k} > 0\Big) \\
&= \mathbb{P}\Big(\exp(\theta(v_m - u_m + \ldots + v_{m+l} - u_{m+l} - u_{m+l+1} - \ldots - u_{m+l+k}) > 1\Big) \\
&\leq \mathbb{E}\Big[\exp(\theta(v_m - u_m + \ldots + v_{m+l} - u_{m+l} - u_{m+l+1} - \ldots - u_{m+l+k})\Big] \\
&= \Big(\mathbb{E}[e^{\theta v_1}]\mathbb{E}[e^{-\theta u_1}]\Big)^l \mathbb{E}[e^{-\theta u_1}]^k,
\end{aligned}$$

where the inequality follows from Chebyshev's inequality (see, for example, Theorem A.113 in Shwartz and Weiss [**?**]). Next, we have for any $k \geq 1$,

$$\mathbb{P}\Big(\bigcup_{l,m=1}^{n} \{u_m + \ldots + u_{m+l} + \ldots + u_{m+l+k} < v_m + \ldots + v_{m+l}\} \Big) \leq n^2 \mathbb{E}[e^{-\theta_0 u_1}]^k.$$

To prove the lemma, we notice that in order for queue length to exceed some constant $c(n)$, it must do so in some busy period. Suppose that the busy period starts at the arrival of the mth job and that the queue length exceeds $c(n)$ for the first time when the $(\ell + m)$th job is in service. Then the arrival time of the $(m+\ell+c(n))$th job happens before the service completion of the $(m + \ell)$th job. Setting

$$c(n) = \lceil -3 \log(\kappa\, n) / \log \mathbb{E}[e^{-\theta_0 u_1}] \rceil,$$

one can check that

$$\begin{aligned}
&\mathbb{P}\Big(\sup_{0 \leq t \leq \tau_n} Z(t) > c(n) \Big) \\
&\quad \leq \quad \mathbb{P}\Big(\bigcup_{l,m=1}^{n} \{u_m + \ldots + u_{m+l} + \ldots + u_{m+l+c(n)} < v_m + \ldots + v_{m+l}\} \Big) \\
&\quad \leq \quad n^2 \mathbb{E}[e^{-\theta_0 u_1}]^{c(n)} \\
&\quad \leq \quad 1/(\kappa\, n).
\end{aligned}$$

The lemma is proved by choosing c such that $c \log n \geq c(n)$ for all $n \geq 2$. ■

A Appendix A

A.1 Fluid and Diffusion approximation of the single server queue

We now give a more leisurely coverage of the fluid and diffusion approximations to the single server queue.

This is done in a sequence of steps. We consider a GI/G/1 system, with renewal arrivals and i.i.d. services. Let $\mathcal{S}(t) = \max\{n : \sum_{j=1}^{n} X_j \leq t\}$.

A.1.1 FSLLN, FCLT and FSAT for renewal processes

The FSLLN for renewal processes

$$\frac{1}{n}\mathcal{A}(nt) \stackrel{a.s.}{\to} \lambda t \quad \text{u.o.c.}, \quad \frac{1}{n}\mathcal{S}(nt) \stackrel{a.s.}{\to} \mu t \quad \text{u.o.c.},$$

follows directly from the FSLLN for sums of i.i.d. r.v's (random walks), which follows directly from the SLLN.

The FCLT for renewal processes

$$\sqrt{n}(\frac{1}{n}\mathcal{A}(nt) - \lambda t) \stackrel{w}{\to} \lambda^{\frac{1}{2}} c_A \, BM(t), \quad \sqrt{n}(\frac{1}{n}\mathcal{S}(nt) - \mu t) \stackrel{w}{\to} \mu^{\frac{1}{2}} c_S \, BM(t),$$

$$\sqrt{n}\left(\frac{1}{n}\left(\mathcal{A}(nt) - \mathcal{S}(nt)\right) - (\lambda - \mu)t\right) \stackrel{w}{\to} \sqrt{\lambda c_A^2 + \mu c_S^2}\, BM(t)$$

follows directly from Donsker's theorem, on the BM limit for random walks.

When interarrivals and service times posess moments of order $r > 2$, the processes $\mathcal{A}(t), \mathcal{S}(t)$ have a Skorohod representation in a joint probability space with a related BM so that:

$$\sup_{0 \leq s \leq t} |(\mathcal{A}(s) - \mathcal{S}(s)) - (\lambda - \mu)s - \sqrt{\lambda c_A^2 + \mu c_S^2}\, BM(s)| \stackrel{a.s.}{=} o((t)^{1/r}) \quad \text{as } t \to \infty$$

A stronger assumption which I believe can often be made in a manufacturing environment is that interarrival and processing time distribution have an exponentially decaying tail. This mean that they have a Laplace transform defined in the neighborhook of 0 and is expessed as existence of exponential moments: $E(e^{\theta X}) < \infty$ for some $\theta > 0$. In that case the FSAT will read (see Glynn []):

$$\sup_{0 \leq s \leq t} |(\mathcal{A}(s) - \mathcal{S}(s)) - (\lambda - \mu)s - \sqrt{\lambda c_A^2 + \mu c_S^2}\, BM(s)| \stackrel{a.s.}{=} O(\log t) \quad \text{as } t \to \infty$$

A.1.2 Dynamics of the single server queue

The queue length process satisfy the equation:

$$Q(t) = Q(0) + \mathcal{A}(t) - \mathcal{S}(B(t)) \tag{1.1}$$

where $Q(0)$ is the initial queue length, to which are added the arrivals, and from which are subtracted the departures which occur as a result of the processing for a total duration of the busy time $B(t)$. The busy time satisfies:

$$B(t) = \int_0^t 1_{\{Q(s)>0\}} ds \tag{1.2}$$

that is it increases at rate 1 while the queue is not empty, and remains constant when the queue is empty.

Clearly the queue length process and the busy time satisfy (1.1,1.2). We shall next see that these equations determine Q, B uniquely. We shall also (this can only be done for the single queue) get explicit expressiong for Q, B, as function of $\mathcal{A}, \mathcal{S}$.

A.1.3 Reflection mapping

Consider a function $x(t)$ in D_0 the space of all RCLL functions on $[0, \infty)$ with $x(0) \geq 0$. Pose the problem of finding functions $z = \Psi(x)$, $y = \Phi(x)$ such that (here r is a positive constant):

$$z(t) = x(t) + ry(t),$$

with the properties

(i) $z(t) \geq 0$.

(ii) $y(0) = 0$ and $y(t)$ is non-decreasing.

(iii) $\int_0^t z(s)dy(s) = 0$.

This is called a Skorohod problem, z is called the reflection mapping of x and y is called the rgulator of x.

Proposition A.1 *x determines y, z uniqely, $y, z \in D$, (iii) can be replaced by the equivalent requirement the y is minimal (for all t), and the mapping $x \to \Psi(x), \Phi(x)$ is Lipshitz conitnuous in the metric of D.*

In fact, for this one dimensional case the explicit expression for y, z is:

$$y(t) = \frac{1}{r} \sup\{x(s)^- : 0 \leq s \leq t\}, \quad z(t) = x(t) \vee \sup\{x(t) - x(s) : 0 \leq s \leq t\}$$

Because the mapping is continuous, $x_n \to x$ implies $z_n, y_n \to z, y$.

A.1.4 Centering the queue balance equation

We rewrite (1.1):

$$\begin{aligned} Q(t) &= [Q(0) + (\lambda - \mu)t + (\mathcal{A}(t) - \lambda t) - (\mathcal{S}(B(t)) - \mu B(t))] + \mu\,[(t - B(t))] \\ &= \mathcal{X}(t) + \mu\mathcal{Y}(t) \end{aligned} \tag{1.3}$$

where we call $\mathcal{X}(t)$ the netput process, and $\mathcal{Y}(t)$ is the idle time of the system.

Here $Q(t) \geq 0$, $\mathcal{Y}(0) = 0$ and $\mathcal{Y}(t)$ is non-decresing, and (1.2) implies

$$\int_0^t Q(s)d\mathcal{Y}(s) = 0 \tag{1.4}$$

from which it is seen that $Q(t), \mathcal{Y}(t)$ are just the reflection mapping of the netput $\mathcal{X}(t)$ and uniquely determined by (1.3,1.4).

A.1.5 The need to consider a sequence of system

We are ready to put things together, by rescaling the netput process and using FSLLN, FCLT and FSAT for it. But we will not get what we want if we just scale $\mathcal{X}(t)$.

On the fluid scale, what is left of $\mathcal{X}(t)$ will be $(\lambda-\mu)t$, since $\frac{1}{n}Q(0) \to 0$, and for $\lambda > \mu$ this has linear reflection for $\lambda \leq \mu$ it has 0 reflection. We cannot distinguish $\rho < 1$ and $\rho = 1$, and we cannot see how the fluid behaves in a stable system which start positive. To do so we need to look at a sequence of systems: Theses may share the same arrival and service streams but will differ initial queue length, i.e. we have $Q^n(0)$. We will then consider these values when $\frac{1}{n}Q^n(0) \to \bar{Q}(0) > 0$.

On the diffusion scale, for $\rho > 0$ the system blows up and $Q = \mathcal{X}$, no surprises here. For $\rho < 0$ the system is stable, and the diffusion limit will be 0, so we lost all the information. For $\rho = 1$ the system is unstable, so the diffusion limit will be unstable. What we want is a model to approximate the behavior for a stable congested system, or for system that is close to stable, we want to see more of the behavior around $\rho = 1$. For that purpose we again look at a sequence of systems, with $\rho \to 1$. To accomodate different ρ we will use a sequence of systems with arrival and service rates λ^n, μ^n. We will want $\sqrt{n}(\lambda^n - \mu^n) \to \theta$, which implies $\rho^n \to 1$. The different systems can still share a common pair of sequences of mean 1 variates from which the interarrivals and sevices are obtained by dividing by λ^n, μ^n.

A.1.6 The fluid limit of GI/G/1 queue

We let $\bar{Q}(0) = \lim_{n\to\infty} \frac{1}{n}Q^n(0)$. We get as $n \to \infty$:

$$\begin{aligned}
\bar{\mathcal{X}}^n(t) &\overset{a.s.}{\to} \bar{\mathcal{X}}(t) = \bar{Q}(0) + (\lambda - \mu)t \\
\bar{Q}^n(t) &\overset{a.s.}{\to} \bar{Q}(t) = \max\{0, \bar{Q}(0) + (\lambda - \mu)t\} \\
\bar{B}^n(t) &\overset{a.s.}{\to} \bar{B}(t) = \begin{cases} t & t \leq \tau \\ \tau + \rho(t-\tau) & t \geq \tau \end{cases} \\
&\qquad \tau = \inf\{t : \bar{Q}(t) = 0\}
\end{aligned}$$

A.1.7 The diffusion limit of GI/G/1 queue

For as long as the fluid limit is > 0 the server is always busy and $\hat{Q}$ is Brownian motion. When the fluid limit reaches 0 and $\lambda < \mu$ the queue is stable and so $\hat{Q} = 0$, while $\hat{B}$ is Brownian motion, capturing the variability of processing times.

The interesting case remains to look at a congested system in which queue length is moving more slowly than individual customers. To capture this we now let $\hat{Q}(0) = \lim_{n\to\infty} \frac{1}{\sqrt{n}} Q^n(0)$ (so that $\bar{Q}(0) = 0$). We also let $\lim_{n\to\infty} \lambda^n = \lim_{n\to\infty} \mu^n = \lambda$, so in the limit $\rho = 1$. This leaves us with $\bar{\mathcal{X}}(t) = \bar{Q}(t) = 0$, $\bar{B}(t) = t$. We let $\theta = \lim_{n\to\infty} \sqrt{n}(\lambda^n - \mu^n)$, which is the scaled deviation from $\rho = 1$.

Now the centered netput process will have a diffusion limit (use FCLT for the renewal processes, and continuous mapping theorem to deal with $B(nt) = n\bar{B}^n(t)$):

$$\begin{aligned}\hat{\mathcal{X}}^n(t) &= \frac{1}{\sqrt{n}}\left[Q^n(0) + (\lambda^n - \mu^n)nt + (\mathcal{A}^n(nt) - \lambda^n nt) - (\mathcal{S}^n(B(nt)) - \mu^n B(nt))\right] \\ &\xrightarrow{w} \hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\end{aligned}$$

The queue length process and the busy time then become:

$$\begin{aligned}\hat{Q}^n(t) = \frac{1}{\sqrt{n}} Q^n(nt) &\xrightarrow{w} \Psi\left(\hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\right) \\ \hat{B}^n(t) = \frac{1}{\sqrt{n}} B^n(nt) - t &\xrightarrow{w} -\frac{1}{\lambda}\Phi\left(\hat{Q}(0) + \theta t + \sqrt{\lambda(c_A^2 + c_S^2)}\, BM(t)\right)\end{aligned}$$

A.2 Diffusion Processes, Brownian Motion and Reflected Brownian Motion

A diffusion process is a Markov process with continuous sample paths. The basic diffusion process is the standard Brownian motion process $BM(t)$, defined by

(i) BM(t) has continuous sample paths

(ii) BM(t) has independent increments, that is $BM(t_1) - BM(t_0), \ldots, BM(t_n) - BM(t_{n-1})$ are independent random variables, for any $t_0 < t_1 < \cdots < t_n$.

(iii) BM(t) has a $N(0, t)$ distribution.

Wiener's theorem states that such a process exists and is unique, and it is in fact fully characterised by *(i) ,(ii)* alone.

Among the elementary properties of standard Brownian motion are its symmetry: $BM(t) =_w -BM(t)$ and its square root scaling: $aBM(t) =_w BM(a^2 t)$.

$BM(t)$ is strongly Markovian, that is to say: For any stopping time T, the process $BM^*(t) = BM(T + t) - BM(T)$ is itself a standard Brownian motion.

While the paths of the standard BM are continuous, they almost surely have infinite variation, that is $v_t(BM(\cdot, \omega)) = \sup\{\sum_{i=1}^n |BM(t_i, \omega) - BM(t_{i-1}, \omega)|\}$ where the supremum is over all finite partitions $0 < t_0 < t_1 < \cdots < t_n < t$, is infinite for all t for almost all ω. On the other hand the quadratic variation of standard Brownian motion, $q_t(BM(\cdot, \omega)) = \lim_{n\to\infty} \sum_{k=1}^{2^n - 1} [BM(\frac{(k+1)t}{2^n}, \omega) - BM(\frac{kt}{2^n}, \omega)]^2$ almost surely exists and is equal to t.

The general Brownian motion with initial state $x \geq 0$, drift m and diffusion coefficient σ^2 is defined as

$$BM_x(t; m, \sigma^2) =_w x + mt + \sigma BM(t)$$

Clearly, $BM_x(t; m, \sigma^2) \sim N(x + mt, \sigma^2 t)$.

A celebrated calculation, using the reflection principle shows that:

$$P(\sup_{0<s<t} BM(s) \leq y) = \Phi(yt^{-\frac{1}{2}}) - \Phi(-yt^{-\frac{1}{2}})$$

while another celebrated calculation, using a change of measure argument, shows that for a general Brownian motion:

$$P(\sup_{0<s<t} BM_x(s; m, \sigma^2) \leq y) = \Phi(\frac{y - x - mt}{\sigma t^{\frac{1}{2}}}) - e^{2my/\sigma^2}\Phi(\frac{-y + x - mt}{\sigma t^{\frac{1}{2}}})$$

where Φ denotes the standard normal distribution function.

Next we define reflected Brownian motion. Consider a general Brownian motion, $BM_x(\cdot; m, \sigma^2)$, and consider its reflection $Y(\cdot, \omega) = \phi_0(BM_x(\cdot, \omega; m, \sigma^2)$ and $\psi_0(BM_x(\cdot, \omega; m, \sigma^2)$, recall that $Y(t, \omega) = \sup_{0<s<t}\{BM_x(s, \omega; m, \sigma^2)^-\}$, and so $Y(t, \omega)$ is nonnegative with monotone increasing paths, with marginal distribution:

$$P(Y(t) \leq y) = \Phi(\frac{y + x + mt}{\sigma t^{\frac{1}{2}}}) - e^{-2my/\sigma^2}\Phi(\frac{-y - x + mt}{\sigma t^{\frac{1}{2}}}).$$

The process $\psi_0(BM_x(\cdot, \omega; m, \sigma^2)$ is called reflected Brownian motion, and we will denote it by $RBM_x(t; m, \sigma^2)$. We have:

$$RBM_x(t, \omega; m, \sigma^2) = BM_x(t, \omega; m, \sigma^2) + Y(t, \omega).$$

This process is nonegative, and while it is > 0 its sample paths behave like sample paths of the Brownian motion that generated it. However when it hits the value 0 it is 'reflected', or 'pushed' away towards the positive values. $Y(t, \omega)$ is the cumulative amount of pushing, and although it is continuous and monotone increasing it is not absolutely continuous, in other words it is not an integral of some pushing rate. Put more precisely, whenever $RBM_x(t, \omega; m, \sigma^2)$ hits 0, $Y(t, \omega)$ will have an uncountable number of points of increase in the neighborhood, but the Lebesgue measure of all the points of increase of $Y(t, \omega)$ is 0; for that reason $Y(t)$ is referred to as a singular control.

The marginal distribution of $RBM_x(t; m, \sigma^2)$ is calculated from that of Y by a time reversal argument. It is

$$P(RBM_x(t; m, \sigma^2) \leq z) = \Phi(\frac{z - x - mt}{\sigma t^{\frac{1}{2}}}) - e^{2mz/\sigma^2}\Phi(\frac{-z - x - mt}{\sigma t^{\frac{1}{2}}}).$$

These results can also be obtained from the Kolmogorov backwards equations asssociated with the diffusion process.

If $m > 0$ then $RBM_x(t; m, \sigma^2)$ is transient. If $m = 0$ it is null recurrent. If $m < 0$ then $RBM_x(t; m, \sigma^2)$ is positive Harris recurrent, and has a stationary distribution. This is given by

$$P(RBM(t \to \infty; m, \sigma^2) \leq z) = 1 - e^{2mz/\sigma^2},$$

in other words, the stationary distirbution is exponential with parameter $-2m/\sigma^2$ and with mean $-\sigma^2/2m$.