1 Sequencing and Scheduling

The sequencing and scheduling literature was first derived in terms of jobs and machines, but it applies to other areas.

Notation:

- n jobs J_i $i = 1, \ldots, n$.
- m machines M_j $j = 1, \ldots, m$.
- O_{ij} operation of job i on machine j.
- p_{ij} processing time of O_{ij}.

Technological constraints. Each job should be processes through the machines in a particular order.

Sequencing is the order in which jobs are processes through machines.

A *schedule* is derived from a sequence by determining when to start each operation, i.e., by *timbetabling* the sequence.

The general job shop problem is to find a sequence and a time table wich is compatible with the technological constraints and is optimal with respect to some measure of performance.

Why is scheduling important?

Need to utilize resources to effectively process or service a variety of different job or customers. As an example consider a print shop, where each job has to go through typesetting, printing, binding and packaging, or aircraft queueing for landing, or cars waiting for repair in a garage.

Why scheduling can be difficult? Suppose 35 students need signatures of nine professors. This can be done in $(35!)^9 = 1.34 \times 10^{360}$.

Additional notation:

- d_i due date.
- r_i ready date.
- $a_i = d_i - r_i$ allowance.
- W_{ik} waiting time of J_i preeceding k^{th} operation.
- $W_i = \sum_{k=1}^{m} W_{ik}$ waiting time of J_i.
- $C_i = r_i + \sum_j p_{ij} + W_i$ completion time.
- $F_i = C_i - r_i$ flow time.
- $L_i = C_i - d_i$ lateness.
- $T_i = L_i^+$ tardiness.
- $E_i = L_i^-$ earliness.
- $\bar{F} = \frac{1}{n} F_i$ average flow time.
- $C_{\max} = \max(C_1, \ldots, C_n)$ makespan.
- $I_j = C_{max} - \sum_i p_{ij}$ idle time.

– $N_w(t)$ number of jobs waiting at time t.

– $N_p(t)$ number of jobs processed at time t.

– $N_c(t)$ number of jobs completed at time t.

– $N_u(t)$ number of jobs unfinished at time t.

Notice that $n = N_w(t) + N_p(t) + N_c(t)$, $N_u(t) = n - N_c(t)$, and $N_u(0) = n$, $N_u(C_{max}) = 0$.
Assumptions:

– Each job is an entity

– No pre-emptions allowed.

– Each job has m distinct operations, one on each machine.

– No cancellations.

– The processign times are independent of the schedule.

– In-process inventory is allowed.

– Ther is only one of each type of machine.

– Machines may be idle.

– No machine may process more than one operation at a time.

– Machines never breakdown and are available throughout the scheduling period.

– The technological constaints are known in advance.

– Ther is no randomness.

Performance measures:

– $F_{\max}, C_{\max}, \bar{F}, \bar{C}$.

– $L_{\max}, T_{\max}, \bar{L}, \bar{T}$, and n_T.

– $\bar{N}_w$

– $\bar{N}_p$.

Classification Sytem $n/m/A/B$ where n and m stand respectively for the number of jobs and the number of machines, A is the type of shop (F for flowshop, i.e., the machine order for all jobs is the same; and G for general shop where there are no restrictions on the form of technological constraints.) and B is the performance measure. For example, $n/2/F/C_max$ is the n job, 2 machine, flow-shop problem where the aim is to minimize make-span.

Regular measures. R is a *regular* measuer if $C \leq C'$ implies $R(C) \leq R(C')$ where C and C' are the vector of competion times under schedules S and S'. The following are known to be regular measures: $\bar{C}, C_{\max}, \bar{F}, F_{\max}, \bar{L}, L_{\max}, \bar{T}, T_{\max}, n_T$. For instance, consider the performance measure $C_{\max}$. Here we have $R(C) = \max(C_1, \ldots, C_n)$. If $C \leq C'$ then clearly

$$R(C) = \max(C_1, \ldots, C_n) \leq \max(C'_1, \ldots, C'_n) = R(C'),$$

so $C_{\max}$ is a regular measure of performance.

Two performance measures are *equivalent* if a schedule which is optimal with respect to one measure is optimal with respect to the other measure and viceversa.The following are equivalent measures of performance:

- $\bar{C}, \bar{F}, \bar{W}$, and $\bar{L}$.
- C_{max}, $\bar{I}$, and $-\bar{N}_p$
- $\bar{N}_u$ and $\bar{C}/C_{\max}$
- $\bar{N}_w$ and $\bar{W}/C_{\max}$

For example, from $L_i = F_i - a_i = C_i - r_i - a_i = C_i - d_i = \sum_j p_{ij} + W_i - a_i$, we can see that

$$\bar{L} = \bar{F} - \bar{a} = \bar{C} - \bar{d} = \bar{W} + \frac{1}{n}\sum_i \sum_j p_{ij} - \bar{a},$$

so $\bar{L}$, $\bar{C}$, $\bar{F}$, and $\bar{W}$ are equivalent measures.

1.1 Single Machine Scheduling

Let p_i processing time of job i. We assume that all the jobs are ready for processing at the beginning of the processing period.

1.1.1 SPT scheduling:

Here $F_i = p_i + W_i$. Consequently, $\bar{F} = \bar{W} + \bar{p}$, so it is sufficient to minimize $\bar{W}$. Now, if the jobs are processed in the order $1, 2, \ldots, n$, then job i has to wait $\sum_{j<i} p_j$ units of time, so

$$\begin{aligned} \bar{W} &= \frac{1}{n}\sum_i \sum_{j<i} p_j \\ &= \frac{1}{n}\sum_i (n-i)p_i \end{aligned}$$

The Hardy-Littlewood Theorem says that this sum is minimized by sequencing the jobs in the order:

$$p_{(1)} \leq \ldots \leq p_{(n)},$$

i.e., to sequence the jobs by shortest processing time.

Example: $n = 6, p = (6, 4, 8, 3, 2, 7, 1)$. With SPT $\bar{W} = 8.43$ FCFS $\bar{W} = 15.43$ and LPT $\bar{W} = 18.14$.

1.1.2 EDD scheduling:

For $n/1//L_{\max}$, the Earliest Due Date (EDD) scheduling

$$d_{(1)} < \ldots < d_{(n)}$$

is optimal.

Let S be a non EDD schedule there exists k such that $d_{(k)} > d_{(k+1)}$ do pairwise interchange and show the schedule is at least as good.

Example: $p = (1, 1, 3, 4, 1, 3)$ and $d = (7, 3, 8, 12, 9, 3)$. EDD is $L_{\max} = 1$.

1.1.3 Moore's Algorithm

Moore's algorithm is designed for $n/1//n_T$ problem. Example: $d = (15, 6, 9, 23, 20, 30)$ and $p = (10, 3, 4, 8, 10, 6)$.

1. Sequence by EDO

2. Find first tardy job, if no such job exists fo to step 4
3. Reject job with largest processing time. Go to 2.
4. End.

An optimal sequence is given by $(2, 3, 4, 6, 1, 5)$.

1.1.4 Lawler's Algorithm

Lawler's algorithm for $n/1//\max_i g_i(C_i)$ under precedence constraints where g_i is and increasing function for each i

Let V set of jobs that can be done last and let $\tau = \sum_i p_i$. Let J_k be the job in V with the smallest $g_k(\tau)$. Then there is an optimal schedule in wich J_k is scheduled last.

Example: $p = (2, 3, 4, 3, 2, 1)$ and $d = (3, 6, 9, 7, 11, 7)$ with $6/1//L_{\max}$. Assume that $J_1 \to J_2 \to J_3$ and $J_4 \to J_5$ and $J_4 \to J_6$.

$V = \{J_3, J_5, J_6\},\ \tau = 15,\ L_3 = 6, L_5 = 4$, and $L_6 = 8$ so J_5 is last. We remove J_5 from the set.

$V = \{J_3, J_6\},\ \tau = 13,\ L_3 = 4, L_6 = 6$, so schedule J_3 last. The procedure is repeated after deleting J_3. An optimal schedule is given by $(J_1, J_2, J_4, J_6, J_3, J_5)$.