Stochastic Scheduling on Parallel Machines

Richard Weber
Statistical Laboratory, University of Cambridge

A talk to accompany Lecture 9

Stochastic Scheduling

Jobs $1, \ldots, n$ are to be processed on a single machine.

- Processing times are $X_1, \ldots, X_n$, which are *ex ante* distributed as independent exponential random variables, $X_i \sim \mathcal{E}(\lambda_i)$ and $EX_i = 1/\lambda_i$, where $\lambda_1, \ldots, \lambda_n$ are known.
- If jobs are processed in order $1, 2, \ldots, n$, then they are finished in expected time $1/\lambda_1 + \cdots + 1/\lambda_n$.
  So the order of processing does not matter.

Stochastic Scheduling

Jobs $1, \dots, n$ are to be processed on a single machine.

- Processing times are $X_1, \dots, X_n$, which are *ex ante* distributed as independent exponential random variables, $X_i \sim \mathcal{E}(\lambda_i)$ and $EX_i = 1/\lambda_i$, where $\lambda_1, \dots, \lambda_n$ are known.
- If jobs are processed in order $1, 2, \dots, n$, then they are finished in expected time $1/\lambda_1 + \cdots + 1/\lambda_n$.
  So the order of processing does not matter.

But now suppose there are m $(2 \leq m < n)$ identical machines working in parallel. Let C_i be the **completion time** of job i.

- $\max_i C_i$ is called the **makespan** (time when all complete).
- $\sum_i C_i$ is called the **flow time** (sum of completion times).

Suppose we wish to minimize the expected makespan.
We can find the optimal order of processing by stochastic dynamic programming. But now we are in continuous time, $t \geq 0$.
So we need the important facts:

(i) $\min(X_i, X_j) \sim \mathcal{E}(\lambda_i + \lambda_j)$;

(ii) $P(X_i < X_j \mid \min(X_i, X_j) = t) = \lambda_i / (\lambda_i + \lambda_j)$.

Suppose we wish to minimize the expected makespan.
We can find the optimal order of processing by stochastic dynamic programming. But now we are in continuous time, $t \geq 0$.
So we need the important facts:

(i) $\min(X_i, X_j) \sim \mathcal{E}(\lambda_i + \lambda_j)$;

(ii) $P(X_i < X_j \mid \min(X_i, X_j) = t) = \lambda_i/(\lambda_i + \lambda_j)$.

Suppose $m = 2$. The optimality equations are

$$
\begin{aligned}
F(\{i\}) &= \frac{1}{\lambda_i} \\
F(\{i,j\}) &= \frac{1}{\lambda_i + \lambda_j}[1 + \lambda_i F(\{j\}) + \lambda_j F(\{i\})] \\
F(S) &= \min_{i,j \in S} \frac{1}{\lambda_i + \lambda_j}[1 + \lambda_i F(S^i) + \lambda_j F(S^j)],
\end{aligned}
$$

where S is a set of uncompleted jobs, and we use the abbreviated notation $S^i = S \setminus \{i\}$.

Let's rewrite the optimality equation. Let $\Lambda = \sum_i \lambda_i$. Then

$$\begin{aligned} F(S) &= \min_{i,j \in S} \frac{1}{\Lambda} \left[1 + \lambda_i F(S^i) + \lambda_j F(S^j) + \sum_{k \neq i,j} \lambda_k F(S) \right] \\ &= \min_{\substack{u_i \in [0,1], i \in S, \\ \sum_i u_i \leq 2}} \frac{1}{\Lambda} \left[1 + \Lambda F(S) + \sum_i u_i \lambda_i (F(S^i) - F(S)) \right] \end{aligned}$$

In all equations there is now the same divisor, Λ.

An event occurs after a time that is exponentially distributed with parameter Λ, but with probability λ_k/Λ this is a 'dummy event' if $k \neq i, j$. This trick is known as **uniformization**.

Let's rewrite the optimality equation. Let $\Lambda = \sum_i \lambda_i$. Then

$$F(S) = \min_{i,j \in S} \frac{1}{\Lambda}\left[1 + \lambda_i F(S^i) + \lambda_j F(S^j) + \sum_{k \neq i,j} \lambda_k F(S)\right]$$

$$= \min_{\substack{u_i \in [0,1], i \in S, \\ \sum_i u_i \leq 2}} \frac{1}{\Lambda}\left[1 + \Lambda F(S) + \sum_i u_i \lambda_i (F(S^i) - F(S))\right]$$

In all equations there is now the same divisor, Λ.

An event occurs after a time that is exponentially distributed with parameter Λ, but with probability λ_k/Λ this is a 'dummy event' if $k \neq i, j$. This trick is known as **uniformization**.

We should start with two jobs of least $\delta_i(S) = \lambda_i(F(S^i) - F(S))$.

The policy of always processing the m jobs of smallest λ_i is called the lowest hazard rate first policy, LHR.
Similarly, we define the highest hazard rate first policy, HHR.

Theorem.

(a) Expected makespan is minimized by LHR.

The policy of always processing the m jobs of smallest λ_i is called the lowest hazard rate first policy, LHR.
Similarly, we define the highest hazard rate first policy, HHR.

Theorem.

(a) Expected makespan is minimized by LHR.

(b) Expected flow time is minimized by HHR.

The policy of always processing the m jobs of smallest λ_i is called the lowest hazard rate first policy, LHR.
Similarly, we define the highest hazard rate first policy, HHR.

Theorem.

(a) Expected makespan is minimized by LHR.

(b) Expected flow time is minimized by HHR.

(c) $E[C_{(n-m+1)}]$ (expected time there is first an idle machine) is maximized by LHR.

The policy of always processing the m jobs of smallest λ_i is called the lowest hazard rate first policy, LHR.
Similarly, we define the highest hazard rate first policy, HHR.

Theorem.

(a) Expected makespan is minimized by LHR.

(b) Expected flow time is minimized by HHR.

(c) $E[C_{(n-m+1)}]$ (expected time there is first an idle machine) is maximized by LHR.

(c) is the **Lady's nylon stocking problem**.

We think of a lady (having $m = 2$ legs) who starts with n stockings, wears two at a time, each of which may fail, and she wishes to maximize the expected time until she has only one good stocking left to wear.

Proof. We prove (a). For ease assume $m = 2$ and $\lambda_1 < \cdots < \lambda_n$. Recall that with $\delta_i(S) = \lambda_i(F(S^i) - F(S))$,

$$F(S) = \min_{\substack{u_i \in [0,1], i \in S, \\ \sum_i u_i \leq 2}} \frac{1}{\Lambda} \left[1 + \Lambda F(S) + \sum_i u_i \delta_i(S) \right].$$

We would like to prove conjecture C.

(C) For any $S \subseteq \{1, \ldots, n\}$, $i, j \in S$ and i, j not the two jobs of least hazards rates in S,

$$i < j \iff \delta_i(S) < \delta_j(S) \tag{1}$$

Truth of (C) would imply that jobs should be started in the order $1, 2, \ldots, n$.

Let π be LHR. Take an induction hypothesis that (C) is true and that $F(S) = F(\pi, S)$ when S is a strict subset of $\{1, \ldots, n\}$. Consider $S = \{1, \ldots, n\}$. Examine $F(\pi, S)$, and $\delta_i(\pi, S)$, under π.

Let S^k denote $S \setminus \{k\}$. For $i \geq 3$,

$$\begin{aligned} F(\pi, S) &= \frac{1}{\lambda_1 + \lambda_2}[1 + \lambda_1 F(S^1) + \lambda_2 F(S^2)] \\ F(\pi, S^i) &= \frac{1}{\lambda_1 + \lambda_2}[1 + \lambda_1 F(S^{1i}) + \lambda_2 F(S^{2i})] \\ \implies \delta_i(\pi, S) &= \frac{1}{\lambda_1 + \lambda_2}[\lambda_1 \delta_i(S^1) + \lambda_2 \delta_i(S^2)], \quad i \geq 3. \end{aligned} \tag{2}$$

Suppose $3 \leq i < j$. Then $\delta_i(S^1) \leq \delta_j(S^1)$ and $\delta_i(S^2) \leq \delta_j(S^2)$. So $\delta_i(\pi, S) \leq \delta_j(\pi, S)$.

Similarly, we can compute $\delta_1(\pi, S)$.

$$
\begin{aligned}
F(\pi, S) &= \frac{1}{\lambda_1+\lambda_2+\lambda_3}[1+\lambda_1 F(S^1)+\lambda_2 F(S^2)+\lambda_3 F(\pi, S)] \\
F(\pi, S^1) &= \frac{1}{\lambda_1+\lambda_2+\lambda_3}[1+\lambda_1 F(S^1)+\lambda_2 F(S^{12})+\lambda_3 F(S^{13})] \\
\implies \delta_1(\pi, S) &= \frac{1}{\lambda_1+\lambda_2+\lambda_3}[\lambda_2\delta_1(S^2)+\lambda_3\delta_1(\pi, S)+\lambda_1\delta_3(S^1)] \\
&= \frac{1}{\lambda_1+\lambda_2}[\lambda_1\delta_3(S^1)+\lambda_2\delta_1(S^2)]. \qquad (3)
\end{aligned}
$$

$$
\delta_i(\pi, S) = \frac{1}{\lambda_1+\lambda_2}[\lambda_1\delta_i(S^1)+\lambda_2\delta_i(S^2)], \quad i \geq 3. \qquad (2)
$$

By comparing (2) and (3) and using our inductive hypothesis, we see that $\delta_1(\pi, S) \leq \delta_i(\pi, S)$.

This completes a step of an inductive proof by showing that (C) is true for S, and that $F(S) = F(\pi, S)$.

We only need to check the base of the induction, when $S = \{1, 2\}$.

This is provided by the simple calculation

$$
\begin{aligned}
\delta_1(\{1,2\}) &= \lambda_1(F(\{2\}) - F(\{1,2\})) \\
&= \lambda_1 \left[\frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2} \left(1 + \frac{\lambda_1}{\lambda_2} + \frac{\lambda_2}{\lambda_1} \right) \right] \\
&= -\frac{\lambda_2}{\lambda_1 + \lambda_2} \\
&\leq -\frac{\lambda_1}{\lambda_1 + \lambda_2} \\
&= \delta_2(\{1,2\}).
\end{aligned}
$$

Lady's nylon stocking problem

The proof of (c) is also similar. The base of the induction is provided by $\delta_1(\{1,2\}) = \lambda_1(0 - 1/(\lambda_1 + \lambda_2))$.

Since we are seeking to maximize $EC_{(n-m+1)}$ we should process jobs for which δ_i is greatest, i.e., least λ_i.

Problem (c) is known as the **Lady's nylon stocking problem**.

We think of a lady (having $m = 2$ legs) who starts with n stockings, wears two at a time, each of which may fail, and she wishes to maximize the expected time until she has only one good stocking left to wear.