CS F364: Design & Analysis of Algorithm

15 Flow Shop Scheduling

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

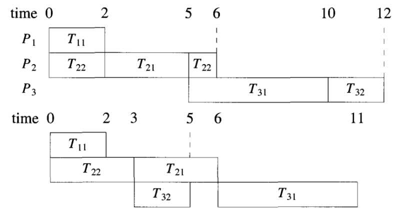
Feb 19, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/algo

Introduction

- Consider n jobs, each having m tasks
- Task can be done on specific shop only (or processor)

$$\begin{bmatrix} 2 & 0 \\ 3 & 3 \\ 5 & 2 \end{bmatrix}$$



Scheduling
Premptive/
nonpremptive

We have n jobs, each requiring m tasks $T_{1i}, T_{2i}, ..., T_{mi}$ for $1 \le i \le n$ where T_{1i} can execute on processor p_i only

Processor could not execute two task at a time. Also T_{2i} cannot execute before T_{1i}

Scheduling

- Finish time of the job i according to schedule S be $f_i(S)$
- Finish time of a schedule S be

$$F(S) = \max_{1 \le i \le n} \{f_i(S)\}$$

We want to get optimal nonpremptive schedule having minimum $F(S)$

- It is difficult to solve for $m > 2$
- Let use solve for the special case where $m = 2$
- Let us simplify the notation by using a_i for t_{1i} and b_i for t_{2i} and

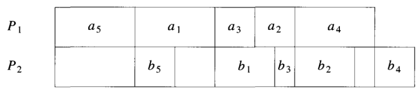
$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}$$

- Let S=(5,1,3,2,4)

Scheduling

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}$$

- Let S=(5,1,3,2,4)



- For $m = 2$, nothing could be gained by using different processing order on $P2$
- Optimal permutation (schedule) has the property that given the first job in the permutation, remaining permutation is optimal

Problem formulation

- $g(S, t)$: length of optimal schedule for the subset of jobs S under the assumption that processor 2 is not available until time t
- We want $g(\{1, 2, 3, ..., n\}, 0)$

$$g(\{1,2,3,...,n\},0) = \min_{1 \le i \le n} \{a_i + g(\{1,2,3,...,n\} - \{i\}, b_i)\}$$

- Here we assume $g(\phi, t) = t$ and $a_i \neq 0$
- Same could be generalized as

$$g(S, 0) = \min_{i \in S} \{a_i + g(S - \{i\}, b_i + \max\{t - a_i, 0\})\}$$

If finish time of the two processors are f_1 and f_2 then $f_2 - f_1 = b_i + max\{a_i, t\} - a_i = b_i + max\{0, t - a_i\}$

Problem formulation

$$g(S, 0) = \min_{i \in S} \{a_i + g(S - \{i\}, b_i + \max\{t - a_i, 0\})\}$$

- If i and j be the first two jobs in the schedule and $P2$ is not available for time t

$$\begin{aligned} g(S, t) &= a_i + g(S - \{i\}, b_i + \max\{t - a_i, 0\}) \\ &= a_i + a_j + g(S - \{i, j\}, b_j + \max\{b_i + \max\{t - a_i, 0\} - a_j, 0\}) \end{aligned}$$

See

$$\begin{aligned} t_{ij} &= b_j + \max\{b_i + \max\{t - a_i, 0\} - a_j, 0\} \\ &= b_j + b_i - a_j + \max\{\max\{t - a_i, 0\}, a_j - b_i\} \\ &= b_j + b_i - a_j + \max\{t - a_i, 0, a_j - b_i\} \\ &= b_j + b_i - a_j - a_i + \max\{t, a_i, a_j + a_i - b_i\} \end{aligned}$$

Problem formulation

$$
\begin{aligned}
t_{ij} &= b_j + b_i - a_j - a_i + \max\{t, a_i, a_j + a_i - b_i\} \\
t_{ji} &= b_j + b_i - a_j - a_i + \max\{t, a_j, a_j + a_i - b_j\}
\end{aligned}
$$

$$g(S, t) = a_i + a_j + g(S - \{i, j\}, t_{ij}\})$$

- If i and j are exchanged then the finish time is

$$g'(S, t) = a_i + a_j + g(S - \{i, j\}, t_{ji}\})$$

Comparing $g(S, t)$ and $g'(S, t)$ we see if

$$\max\{t, a_i, a_j + a_i - b_i\} \leq \max\{t, a_j, a_j + a_i - b_j\}$$

then $g(S, t) < g'(S, t)$

Problem formulation

For $g(S, t) < g'(S, t)$

$$\max\{t, a_i, a_j + a_i - b_i\} \leq \max\{t, a_j, a_j + a_i - b_j\}$$

should hold for all values of t so

$$\max\{a_i, a_j + a_i - b_i\} \leq \max\{a_j, a_j + a_i - b_j\}$$

$$a_i + a_j + \max\{-a_j, -b_i\} \leq a_i + a_j + \max\{-a_i, -b_j\}$$

$$\min\{a_j, b_i\} \geq \min\{a_i, b_j\}$$

Schedule

If $\min\{a_1, a_2, ..., a_n, b_1, b_2, ..., b_n\}$ is a_i then i should be the first job
If $\min\{a_1, a_2, ..., a_n, b_1, b_2, ..., b_n\}$ is b_j then j should be the last job

Example: find schedule for

$$\begin{bmatrix} 3 & 4 & 8 & 10 \\ 6 & 2 & 9 & 15 \end{bmatrix}$$

Thank You!

Thank you very much for your attention! (Reference[1])

Queries ?

[1] [1] Book - *Introduction to Algorithm*, By THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN