July 11th, 2018

4541

Recall:

R||Cmax: R - flowshop - Each job j consists of m operations: operation 1 has to be completed on m/c 1, operation 2 has to be completed on m/c 2, etc.

F2||Cmax: There is always an optimal schedule that is a permutation schedule

(Def'n - Permutation schedule - All jobs are processed in the same ~~order~~ order on all m/c's)

We want to develop an algorithm for R2||Cmax.

Observations:

1) If we decrease all P_{ij}'s by Δ, where $P_{ij} - \Delta \geq 0 \ \forall i,j$, then the ~~set~~ of optimal permutation schedules is unchanged.

Why?:

- Let S be a permutation schedule corresponding to ~~per~~ permutation $1, \ldots, n$. ~~Define $C_{i,j}^S$~~

Define $C_{i,j}^S$ = completion time of i^{th} operation of job j in S, where S does not have any unforced idle time

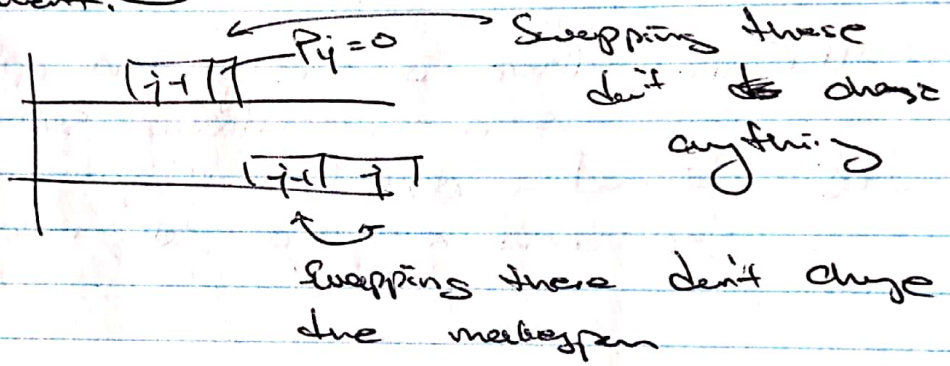
So, $C_{1,j}^S = \sum_{k=1}^{j} P_{1k} \ \forall j$, and in general:

$$C_{i,j}^S = \max\left(C_{i,j-1}^S, C_{i-1,j}^S\right) + P_{ij} \quad \forall i,j$$

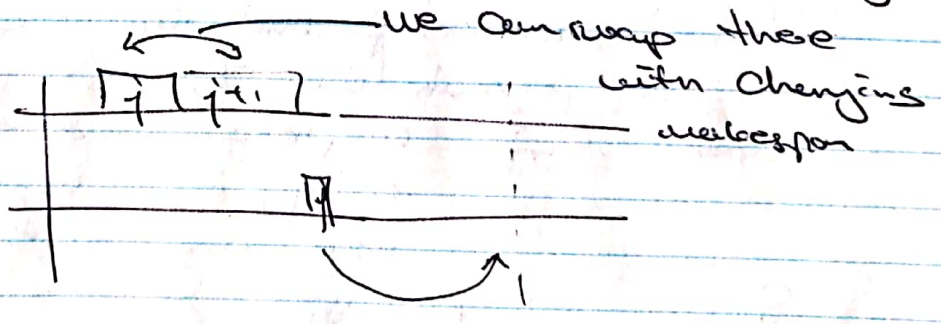
Can't

Observations (cont'd):

2) If $P_{ij} = 0$, then there is always an optimal permutation schedule where j is scheduled ~~first~~ as the first job.
This is easy to see w/ an interchange argument:



3) Analogously, if $P_{ji} = 0$ for some j, then there is ~~one~~ always an optimal permutation schedule where j is scheduled last. Again, we can see this via an interchange argument.



Algorithm:

1) Decrease all p_{ij}'s by $\Delta_i = \min_j p_{ij}$ to get $\hat{p}_{i,j}$'s ($\hat{p}_{ij} = p_{ij} - \Delta \; \forall i,j$)

2a) If some $\hat{p}_{1j} = 0$, then schedule j first and recurse on remaining jobs

2b) If some $\hat{p}_{2j} = 0$, then schedule j last and recurse on remaining jobs.

Algorithm (restated) ⇒ Johnson's Algorithm

Let $J_1 = \{j : p_{1j} \leq p_{2j}\}$ → jobs for which 2a will apply

$J_2 = \{j : p_{2j} < p_{1j}\}$ → jobs for which 2b will apply

(We schedule J_1 first in increasing p_{1j} order as this is the order in which $\hat{p}_{1j}$ hits 0. Then we schedule J_2 in decreasing p_{2j} order, as for these jobs $\hat{p}_{2j}$ hits 0 in opposite order.

Ex:

	Jobs					
	1	2	3	4	5	6
M1	3	2	1	3	2	1
M2	2	2	4	5	3	1	3

$J_1 = \{2, 3, 4, 6\}$

$J_2 = \{1, 5\}$

So, our permutation is $\underbrace{3, 6, 2, 4}_{J_1}, \underbrace{1, 5}_{J_2}$.

We still need to prove observation 1.

Lemma 1:
Let I be an $F2||C_{max}$ instance with p_{ij} operation lengths. Let $\hat{I}$ be the instance obtained from I with operation lengths $\hat{p}_{ij} = p_{ij} - \Delta \geq 0 \quad \forall i,j$. Then S is an optimal permutation schedule for I iff it is an optimal permutation schedule for $\hat{I}$. Moreover, if S is optimal, then $C^S_{max} = OPT(I) = OPT(\hat{I}) + (n+1)\Delta$. (where n is the # of jobs)

[Proof].
Let S be an optimal permutation schedule for I corresponding to permutation $1, \ldots, n$.
Let C^S_{1j}, C^S_{2j} be the completion times of operation 1 and operation 2 of j under S for instance I, where S does not have any unforced idle time.

So, $C^S_{max} = OPT(I) = C^S_{2,n}$.
Now schedule jobs in order $1, \ldots, n$ for instance $\hat{I}$ as follows:

- On m/c 1, schedule j so that it completes at time $\hat{C}^S_{1j} = \sum_{k=1}^{j} \hat{p}_{1k} = \sum_{k=1}^{j} (p_{1k} - \Delta) = C^S_{1j} - j\Delta$.

- On m/c 2, schedule j so that it completes at time: $C^S_{2j} - (j+1)\Delta$. (i.e. Start j on m/c 2 at time $C^S_{2j} - (j+1)\Delta - \hat{p}_{2j}$)

$\subseteq_{\neq}$

Proof (Cont'd)

We claim:
$\hat{C}_{ij}$'s form completion times of a valid flow shop schedule. (i.e. At every time, at most one operation of a job is processed; on any m/c, at any time $\leq I$ operation is processed)

We need to check that: (a) $\hat{C}^S_{ij} - \hat{C}^S_{2,j-1} \geq \hat{P}_{2j}$
(b) $\hat{C}^S_{2,1} - \hat{C}^S_{1,j} \geq \hat{P}_{2,1}$

(a) $\hat{C}^S_{2,1} - \hat{C}^S_{2,j-1} = \underbrace{C^S_{2,1} - C^S_{2,j-1}} - \Delta$

$\geq P_{2j}$ (since $C^S_{2,j}$'s form valid completion times for a valid schedule of I)

$\geq P_{2j} - \Delta = \hat{P}_{2j}$

(b) $\hat{C}^S_{2,j} - \hat{C}^S_{1,j} = (C^S_{2,j} - (j+1)\Delta) - (C^S_{1,j} - j\Delta)$

$= \underbrace{C^S_{2,j} - C^S_{1,j}} - \Delta$

$\geq P_{2j}$ (same reasoning as above)

$\geq P_{2j} - \Delta = \hat{P}_{2,j}$

$\Rightarrow$ S gives a permutation schedule for $\hat{I}$ of makespan $\hat{C}^S_{2,n} = C^S_{2,n} - (n+1)\Delta$

$= OPT(I) - (n+1)\Delta$ (*)

$\Rightarrow OPT(\hat{I}) \leq OPT(I) - (n+1)\Delta$

[Proof] (cont.)

Conversely, if S is an optimal perm. schedule for $\hat{I}$, then S leads to a permutation schedule for I of makespan $\leq OPT(\hat{I}) + (m+1)\Delta$ (Exercise)
(*) (**)

(*) and (**) show
- $OPT(I) = OPT(\hat{I}) + (m+1)\Delta$
- S is an opt. perm. schedule for I iff ——— " ——— $\hat{I}$.

□

$R||C_{max}$:

Goal: Devise a polytime algorithm (permutation) with guarantee: It returns a schedule S with makespan $\leq OPT + (m-1)\cdot p_{max}$, where $p_{max} := \max_{i,j} p_{ij}$.

Notation:
Let $\Pi_i = \sum_{j=1}^{n} p_{ij}$ and $\Pi_{max} = \max_{i=1,\dots,m} \Pi_i$.

Clearly, $OPT \geq \Pi_{max}$.
Suppose we have a permutation schedule S that schedules jobs in the order:
$\sigma(1), \sigma(2), \dots, \sigma(n)$, where $\sigma: \{1,\dots,n\} \to \{1,\dots,n\}$ is a permutation, s.t. there is no unforced idle time.

As usual, $C_{ij}^S :=$ Completion time of ith operation of job j under schedule S.
And, we know $C_{ij}^S =$

Co

July 07th, 2018

And,

$$C^S_{i,\sigma(j)} = \sum_{k=1}^{j} P_{i,\sigma(k)}, \text{ and}$$

$$C^S_{i,\sigma(j)} = \max\{C^S_{i,\sigma(j-1)}, C^S_{i-1,\sigma(j)}\} + P_{i,\sigma(j)}$$

So, $C^S_{max} = C^S_{m,\sigma(n)} = \sum_{j=1}^{n} P_{m,\sigma(j)}$ + (Total idle time on m/c m in $[0, C^S_{max}]$).