Single Machine Deterministic Models

Jobs: J_1, J_2, ..., J_n

Assumptions:
- The machine is always available throughout the scheduling period.
- The machine cannot process more than one job at a time.
- Each job must spend on the machine a prescribed length of time.

Requirements that may restrict the feasibility of schedules:
- precedence constraints
- no preemptions
- release dates
- deadlines

Whether some feasible schedule exist? NP hard

Objective function $f$ is used to compare schedules.

$f(S) < f(S')$ whenever schedule $S$ is considered to be better than $S'$

1. Completion Time Models

Due date related objectives:
2. Lateness Models
3. Tardiness Models
4. Sequence-Dependent Setup Problems

Completion Time Models

Contents
1. An algorithm which gives an optimal schedule with the minimum total weighted completion time
   $1 \parallel \sum w_j C_j$
2. An algorithm which gives an optimal schedule with the minimum total weighted completion time when the jobs are subject to precedence relationship that take the form of chains
   $1 | \text{chain} | \sum w_j C_j$

Literature:
Theorem. The weighted shortest processing time first rule (WSPT) is optimal for $1 \| \sum w_j C_j$

WSPT: jobs are ordered in decreasing order of $w_j/p_j$

The next follows trivially:

The problem $1 \| \sum C_j$ is solved by a sequence $S$ with jobs arranged in nondecreasing order of processing times.

Proof. By contradiction.

$S$ is a schedule, not WSPT, that is optimal.

$j$ and $k$ are two adjacent jobs such that $w_j/p_j < w_k/p_k$ which implies that $w_j p_k < w_k p_j$

$S$: \[ t + p_j + p_k \]

$S'$: \[ t + p_k + p_j \]

$S'$: the completion time for $S'$ < completion time for $S$ contradiction!

Lemma. If job $l^*$ determines $\rho(1,...,k)$, then there exists an optimal sequence that processes jobs $1,...,l^*$ one after another without interruption by jobs from other chains.

Algorithm

Whenever the machine is free, select among the remaining chains the one with the highest $\rho$ factor. Process this chain up to and including the job $l^*$ that determines its $\rho$ factor.

Example

chain 1: 1 → 2 → 3 → 4
chain 2: 5 → 6 → 7

 jobs 1 2 3 4 5 6 7
$w_j$ 6 18 12 8 8 17 18
$p_j$ 3 6 6 5 4 8 10

$\rho$ factor of chain 1 is determined by job 2: $(6+18)/(3+6)=2.67$
$\rho$ factor of chain 2 is determined by job 6: $(8+17)/(4+8)=2.08$

chain 1 is selected: jobs 1, 2

$\rho$ factor of the remaining part of chain 1 is determined by job 3: 2
$\rho$ factor of the remaining part of chain 2 is determined by job 7: 18/10=1.8

chain 1 is selected: job 3

job 4 is scheduled last

the final schedule: 1, 2, 5, 6, 3, 7, 4
1 \mid prec \mid \sum w_j C_j

- Polynomial time algorithms for the more complex precedence constraints than the simple chains are developed.
- The problems with arbitrary precedence relation are NP hard.
- \(1 \mid r_j, prmp \mid \sum w_j C_j\) preemptive version of the WSPT rule does not always lead to an optimal solution, the problem is NP hard
- \(1 \mid r_j \mid \sum C_j\) preemptive version of the SPT rule is optimal
- \(1 \mid r_j \mid \Sigma C_j\) is NP hard

Summary

- \(1 \parallel \sum w_j C_j\) WSPT rule
- \(1 \mid \text{chain} \mid \sum w_j C_j\) a polynomial time algorithm is given
- \(1 \mid prec \mid \sum w_j C_j\) with arbitrary precedence relation is NP hard
- \(1 \mid r_j, prmp \mid \sum w_j C_j\) the problem is NP hard
- \(1 \mid r_j, prmp \mid \Sigma C_j\) preemptive version of the SPT rule is optimal
- \(1 \mid r_j \mid \Sigma C_j\) is NP hard