Applied Algorithms. Prof. Tami Tamir
Missing Proofs, lecture 3. July 3rd.

The Problem $R||\Sigma_j C_j$
Theorem: a minimum weight perfect matching corresponds to an optimal schedule.

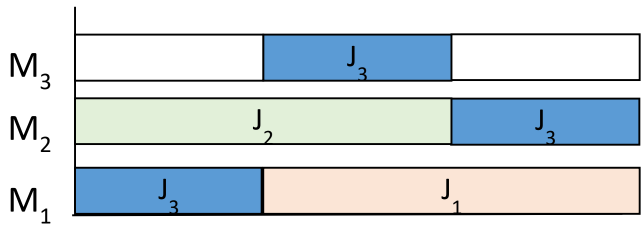
Proof: Let M be a min-weight perfect matching. For every machine i, it holds that for some k the vertices corresponding to the last k positions on Mi are matched. Otherwise, it is possible to reduce the matchings' cost. Specifically, if w_{ik} is not matched and $w_{i,k+1}$ is matched with vj, then the matching M'= M-{(vj, $w_{i,k+1}$)} $\cup$ {(vj, $w_{i,k}$)} is cheaper by p_{ij}.
This implies that a min-weight perfect matching corresponds to a feasible schedule.
By the construction, the matching has cost A iff $\Sigma_j C_j$=A. Thus min-weight iff min total flow time.

Hardness of Open-shop scheduling

Theorem: The problem $O3||C_{max}$ (Minimize makespan of an open-shop schedule on 3 machines) is NP-hard.

Proof: Reduction from *Partition*. The input for *Partition* is a set A of n integers $a_1,\dots,a_n$ whose total sum is 2B. The problem is to find a subset of these integers whose total sum is exactly B.
Given an instance A for *Partition* build the following instance for $O3||C_{max}$:
There are n+3 jobs. For the first job $p_{1,1}$=2B, $p_{2,1}$=$p_{3,1}$=0, for the second job, $p_{1,2}$=$p_{3,2}$=0, $p_{2,2}$=2B, for the third job $p_{1,3}$= $p_{2,3}$= $p_{3,3}$ =B. For the last n jobs, that is, for j=4,…,n+3, $p_{1,j}$= $p_{2,j}$=0, $p_{3,j}$=a_j.
(where $p_{i,j}$ is the length of the task of Job j on M_i)

Claim: C_{max} = 3B if and only if the set S has a partition.
The proof has two parts – one for each direction of the 'if and only if'
1. Assume there is a schedule with C_{max} = 3B, it must be that the sub-jobs of job J_3 are processed one after the other with no delay. Since J_1 and J_2 occupy M_1 and M_2 respectively for intervals of length 2B, J_3 must be processed on one of these machines (w.l.o.g M_1) during the interval [0,B], on M_2 during the interval [2B,3B], and on M_3 during the interval [B,2B] (see figure).



As a result, the processing of the n jobs j=4,…,n+3 on M_3 splits between the intervals [0,B] and [2B,3B], inducing a partition of these jobs into two sets of total processing time B.

2. Assume the set S has a partition into two sets U and W each with total size B. A valid schedule with C_{max} = 3B is the following: On M_1: J_3 followed by J_1. On M_2: J2 followed by J_3. On M3: the jobs originated from U, followed by J_3, followed by the jobs originated from W. Since the total processing time of job originated from U and from W is exactly B, M_3 is processed with no idle and the jobs from U and W exactly fill the B-segments before and after the processing of J_3.