Each job j=1,...,n is to be processed by exactly one machine i=1,...,m. Scheduling job j on machine i has time $p_{ij}$ and cost $c_{ij.}$

If jobs could be assigned to machines fractionally, we could use an LP to find a schedule in which fraction $x_{ij}$ of job j is done on machine i. This pseudo-schedule would have cost C and makespan (latest completion time of a job) T.

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \le C$$

$$\sum_{i=1}^{m} x_{ij} = 1 \text{ for } j=1,...,n$$

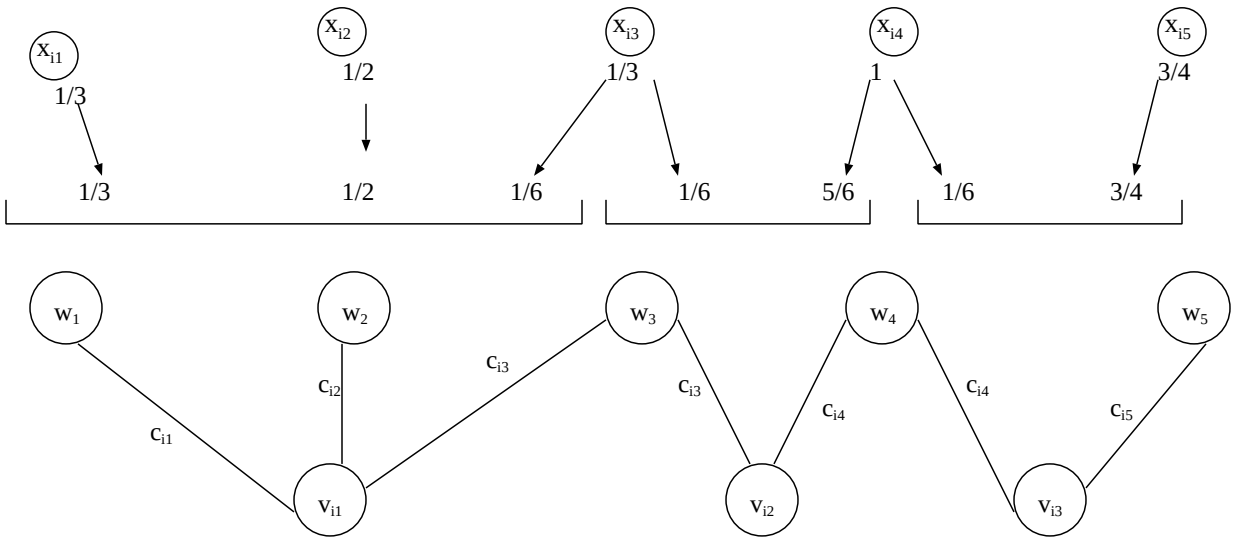
$$\sum_{j=1}^{n} p_{ij}\, x_{ij} \le T \text{ for } i=1,...,m$$

$$x_{ij} \ge 0$$

Given a fractional schedule of cost C and makespan T, we can find a schedule of cost at most C and makespan at most 2T.
We convert fractional solution x to a schedule where each job is entirely processed by exactly one machine.
First, we construct a bipartite graph B(x) = (V,W,E) with a value x'(v,w) for each edge.
W consists of job nodes $w_j$ with j=1,...,n. V is more complicated.

For a particular machine i,
find $\sum_{i=1}^{m} x_{ij}$ and then find $k_i = \lceil \sum_{i=1}^{m} x_{ij} \rceil$.
Machine i will have $k_i$ nodes in V, labeled $v_{is}$ with s=1,..,$k_i$.

Edges E are created for $x_{ij} > 0$. For each $x_{ij} > 0$ there will be one or two edges in B(x), incident on node $w_j$ and on one or two nodes $v_{is}$.
The values of $x'(v_{is},w_j)$ will be assigned so that $x_{ij} = \sum_{s=1}^{k_i} x'(v_{is},w_j)$.
Creating edges:
For a particular machine i, sort jobs in order of non-increasing processing time $p_{ij}$, so that $p_{i1} \ge p_{i2} \ge \ldots \ge p_{in.}$
Consider the $x_{ij}$ in the same order as their respective $p_{ij}$.
Assign $x_{ij}$'s to machine nodes $v_{is}$ so that the sum of the $x_{ij}$'s assigned to each $v_{is}$ is equal to one (except perhaps for the last $v_{is}$).
example: $\sum_{i=1}^{m} x_{ij} = (35/12)$ so $k_i=3$




Assign weights to the edges. The weight of edge $(v_{is},w_j)$ is $c_{ij}$, the cost of doing job j on machine i.

Definitions: A vector on the edges of a graph means that a number is associated with each edge of the graph.
A non-negative vector z on the edges is a fractional matching if for each node v,
the sum of the components of z on the edges incident to v is at most one.
The fractional matching exactly matches a node u if the corresponding sum is exactly one.
A fractional matching z is a matching if each component of z is 0 or 1.

The vector x' of values x' $(v_{is},w_j)$ is a fractional matching in B(x) of cost at most C.

It exactly matches each job node $w_j$, j=1,...,n, and some of the machine nodes $v_{is}$, i=1,...,m, s=1,...,$k_i$-1.

This implies that there exists an integral matching M in B(x) of cost at most C which exactly matches all job nodes. Why?
Consider B(x) as a flow network for min-cost flow, with a supersource with supply n connected to the job nodes,

and a supersink with demand n connected to the machine nodes, unit capacities on all edges, and cost $c_{ij}$ on edge $(v_{is},w_j)$.

The fractional matching x' corresponds to a flow x' $(v_{is},w_j)$ of cost C. This flow has 1 coming out of each job node, so it is the max flow.
But finding an integral min-cost flow in this network is possible because all edge capacities are integral.
This min-cost flow would have the same total flow of value n and cost at most C.
This would correspond to an integral matching in B(x).

Find a minimum cost matching M in B(x) which exactly matches all job nodes. For each edge $(v_{is},w_j)$ in M, schedule job j on machine i.
The cost of this schedule is at most C.

Now that we have shown that how to find an integral schedule of cost at most C, the next thing is to show that the makespan is at most 2T:

Let $p_{is}^{max}$ be the maximum of processing times p corresponding to edges $(v_{is},w_j)$.

Let $p_{is}^{min}$ be the minimum of processing times p corresponding to edges $(v_{is},w_j)$.

Note that $p_{is}^{min} \geq p_{i,s+1}^{max}$ for i=1,...,m and s=1,...,$k_i$-1

At most $k_i$ jobs might be scheduled on machine i, for a total time of $\sum_{s=1}^{k_i} p_{is}^{max}$.

Note that $p_{i1}^{max} \leq T$.

The rest: $\sum_{s=2}^{k_i} p_{is}^{max} \leq \sum_{s=1}^{k_i-1} p_{is}^{min}$ because $p_{is}^{min} \geq p_{i,s+1}^{max}$ for i=1,...,m and s=1,...,$k_i$-1

$\leq \sum_{s=1}^{k_i-1} \sum_{j=1}^{n} p_{ij}\, x'(v_{is},w_j)$ because $p_{is}^{min}$ is the minimum value associated with $v_{is}$ and because the x' values are $\leq 1$

$\leq \sum_{s=1}^{k_i} \sum_{j=1}^{n} p_{ij}\, x'(v_{is},w_j)$ because that just includes one more term in the summation

$= \sum_{s=1}^{k_i-1} \sum_{j=1}^{n} p_{ij}\, x_{ij}$ because of how the x'$(v_{is},w_j)$ values were constructed

$\leq T$ because that is a constraint of the original LP

So $\sum_{s=1}^{k_i} p_{is}^{max} \leq 2T$