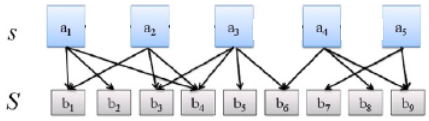
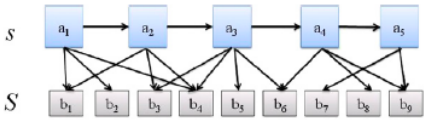
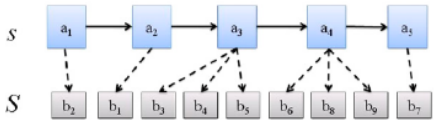
Single-Machine Scheduling with with Supporting Tasks

- Tasks $\{1,2,\ldots,m\}$ and Jobs $\{1,2,\ldots,n\}$ are to be processed on a single machine (So, we have $m+n$ positions on the machine)
- Task i is characterized by processing time t_i, Job j is characterized by processing time p_j and weight w_j
- For task i, $\mathcal{J}_i$ is the set of jobs that can start only when task j is finished
- For job j, $\mathcal{T}_j$ is the set of tasks that must precede the start of job j
- The supporting relation is presented as a bi-partite graph
- At any time, the machine can perform at most one task or one job
- A schedule is feasible if the precedence constraints are observed
- The problem is to determine a schedule that has the minimum $\sum_{j=1}^{n} w_j C_j$, where C_j is the completion time of job j



Tasks Sequence is Given





Reduce the size $O((m+n)!)$ of solution space to $O(m!)$

We can also reduce the size to $O(n!)$ by the technique of fixing a job sequence.