Interval Scheduling Problem on m machines (m-ISP): Schedule a set of intervals $\{I_1, I_2, \cdots, I_n\}$ on m machines such that no two intervals scheduled on the same machine intersect. Note that each interval I_i has a start time s_i and a finish time f_i. This problem is an extension of the standard Interval Scheduling Problem discussed in the lecture.

An optimal algorithm

Algorithm 1: Best Fit EFT (an extension of the standard EFT algorithm)

Sort intervals such that $f_1 \leqslant f_2 \leqslant \ldots \leqslant f_n$
for $k = 1$ *to* m **do**
  $e_k = 0$ `//` e_k `is the latest finish time of intervals on machine` k.
for $i = 1$ *to* n **do**
  Let $k = \begin{cases} \arg\min_l (s_i - e_l \geqslant 0) & \text{if such } l \text{ exists} \\ 0 & \text{if such } l \text{ does not exist} \end{cases}$
  $\sigma(i) = k$ `//` $\sigma(i)$ `specifies on which machine Interval` I_i `is scheduled.` $\sigma(i) = 0$ `means that` I_i `is not scheduled.`
  $e_k = f_i$

Proof of optimality: The exchange proof method.
Idea: Let $S_0, S_1, ..., S_n$ be the partial solutions constructed by the algorithm at the end of each iteration. The solution S_i contains the scheduling for intervals $I_1, \cdots, I_i$.
Prove each S_i can be *completed (extended)* to reach an optimal solution (just by scheduling $I_{i+1}, \cdots, I_n$). Call that optimal solution S'_i. The scheduling for all intervals $I_1, \cdots, I_i$ are the same in both S_i and S'_i.
If S'_i exists, we say S_i is *promising*.
Note: S'_i may not be unique (there may be more than one way to achieve optimal).
Prove that S_i *is promising* by induction in i (number of iterations).

Proof:

- Base case: $S_0 = \{\}$: any optimal solution S'_0 extends S_0 just by scheduling the intervals in $\{I_1, ..., I_n\}$.
- Ind. Hyp.: Suppose $i \geqslant 0$ and optimal S'_i extends S_i by scheduling only the intervals in $\{I_{i+1}, ..., I_n\}$.
- Ind. Step (To prove): S_{i+1} is promising w.r.t. $\{I_{i+2}, ..., I_n\}$.

  Let's see what happens in iteration $i + 1$. There are two cases.

  1. The algorithm sets $\sigma(i + 1) = 0$
     It means that I_{i+1} conflicts with all machines according to the S_i scheduling. Thus, in S'_i we should have $\sigma_{S'_i}(i + 1) = 0$ (otherwise, S'_i has a conflict and it is not a solution). Set $S'_{i+1} = S'_i$. Thus, S_{i+1} is promising.
     Note: $\sigma_{S'_i}(i + 1)$ is the scheduling for interval I_{i+1} in S'_i.
  2. The algorithm sets $\sigma(i + 1) = k$ $(k \neq 0)$
     Three cases may happen:
     (a) $\sigma_{S'_i}(i + 1) = k$
     Set $S'_{i+1} = S'_i$. Thus, S_{i+1} is promising.
     (b) $\sigma_{S'_i}(i + 1) = 0$
     It means that there is an interval I_j scheduled by S'_i on machine k that conflicts with I_{i+1}; otherwise we can change $\sigma_{S'_i}(i + 1)$ to k (schedule I_{i+1} on machine k) and get a better solution. It means

that S'_i is not optimal that is a contradiction!
Moreover, $j > i+1$ and also I_j is unique. Why? If there are two intervals I_{j_1} and I_{j_2}, since $f_{i+1} \leqslant f_{j_1}$ and $f_{i+1} \leqslant f_{j_2}$, they should conflict. Hence they cannot be part of a solution.
Therefore if we set $\sigma_{S'_i}(i+1) = k$ and $\sigma_{S'_i}(j) = 0$, the updated scheduling S'_i still extends S_i and is optimal.
Set S'_{i+1} to this updated S'_i. Hence, S_{i+1} is promising.

(c) $\sigma_{S'_i}(i+1) = k'$ ($k' \neq k$, $k' \neq 0$)
Look at machines k and k'. First we know that $s_{i+1} - e_k \geqslant 0$. Thus, $s_{i+1} \geqslant e_k$.
Second, $s_{i+1} - e_k$ has the minimum positive value among all machines. Thus, $e_{k'} \leqslant e_k$.
Substitute all jobs after e_k on machine k with all jobs after $e_{k'}$ on machine k'. Note that the number of scheduled intervals remain the same and there is no conflict. why?
In the new scheduling, I_{i+1} is scheduled on machine k. This scheduling can be utilized to extend S_{i+1}. Hence, S_{i+1} is promising.

Thus, S_n is promising. It means that S_n is optimal.