Lecture 2 (Job Scheduling)

1 Formal Statement

Given an array of n jobs, represented by their start and end times: $[s_i, e_i]$, $1 \leq i \leq n$, find the largest subsequence of non-interecting jobs.

2 Greedy Strategy

Out of the set of *not-chosen* intervals, choose the interval with the smallest end time and remove the interesecting intervals from the *not-chosen* set.

3 Algorithm

```
solution = []
while (!isEmpty(jobs)) {
    interval = min(jobs) // smallest with respect to end time
    solution.add(interval)
    for (job in jobs) {
        if (intersection(interval, job)) {
            jobs.remove(job)
        }
    }
}
```

4 Proof

Consider any optimal set, P. Now, we will prove that we can generate a set G using the greedy strategy, such that $|P| = |G| = n$.

Induction

Hypothesis: $G' = P \setminus \{P_1, P_2, \ldots, P_i\} \cup \{G_1, G_2, \ldots, G_i\}$ is a valid scheduling $\forall i \in \{1, 2, \ldots, n\}$

Base Case: $(i = 0)$ $G' = P$ is a valid scheduling by assumption.

Inductive Step:
Consider it to be true for $i = k$. Now, consider the set of *not-chosen* intervals as all those intervals which do not intersect with $\{G_1, G_2, \ldots, G_i\}$. Call this set A.
Now consider, P_{k+1}. It is easy to prove that this interval is present in A. Consider the interval with the smallest end time out of all elements in A and call it g. Thus, since P_{k+1} is in A, $endTime(P_{k+1}) \geq endTime(g)$.
Therefore, we can replace P_{k+1} with g, such that $G_{k+1} = g$. This completes the inductive step.

Completion: This completes the proof of correctness of the greedy strategy of the scheduling problem.

5 Followups

1. Find the condition for a unique optimal solution.
2. If instead of a single *server*, there are two *servers*, find the optimal scheduling.

5.1 Followup 1

Idea: Every interval not a part of the optimal scheduling, should intersect with > 1 interval of the optimal scheduling.