Algorithms Design and Analysis [ETCS-301]

Dr. A K Yadav
Amity School of Engineering and Technology
(affiliated to GGSIPU, Delhi)
akyadav1@amity.edu
akyadav@akyadav.in
www.akyadav.in
+91 9911375598

October 8, 2019

An activity-selection problem

- Suppose $S = \{a_1, a_2, \ldots, a_n\}$ is a set of n activities
- Each activity a_i have start time s_i and finish time f_i such that $0 \leq s_i < f_i < \infty$
- Two activity a_i and a_j are compatible if one start after other finishes that is either $s_i \geq f_j$ or $s_j \geq f_i$ or we can say time interval $[s_i, f_i)$ of a_i and $[s_j, f_j)$ of a_j is non overlapping.
- a_0 finishes at 0 and a_{n+1} starts after f_n that is $[-\infty, 0)$ for a_0 and $[f_n, \infty)$ for a_{n+1}

Dynamic programming Solution

- S_{ij} is the set of activities that start after activity a_i finishes and that finish before activity a_j starts
- A_{ij} is the set of maximum mutually compatible activities that start after activity a_i finishes and that finish before activity a_j starts and $Aij \subseteq Sij$
- We want to find A_{0n+1}
- Suppose $a_k \in A_{ij}$
- There is two sub-problems A_{ik} and A_{kj} where $A_{ik} = A_{ij} \cap S_{ik}$ and $A_{kj} = A_{ij} \cap S_{kj}$
- $A_{ij} = Aik \cup \{a_k\} \cup A_{kj}$
- $|A_{ij}| = |Aik| + |A_{kj}| + 1$
- If $c[i,j]$ is the size of an optimal solution for the set S_{ij} then $c[i,j] = c[i,k] + c[k,j] + 1$

$$c[i,j] = \begin{cases} 0 & \text{if } S_{ij} = \emptyset \\ \max\limits_{a_k \in S_{ij}} \{c[i,k] + c[k,j] + 1\} & \text{if } S_{ij} \neq \emptyset \end{cases}$$

Greedy choice Solution

- We should choose an activity that leaves the resource available for as many other activities as possible
- Choose the activity in S with the earliest finish time, so that maximum resource available for the following activities
- Since activities are in sorted increasing order of finishing time so select a_1
- After selecting a_1 only one sub-problem remains: selecting the activity from rest of the activities witch starts after a_1.
- Let $S_k = \{a_i \in S : s_i \geq f_k\}$ is the set of activities that start after activity a_k finishes
- So a_1 is finishing first, we have to find out S_1 only.

A recursive greedy algorithm

RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)

1. $m = k + 1$
2. while $m \leq n$ and $s_m < f_k$ // find the first activity in S_k to finish
3. $\quad m = m + 1$
4. if $m \leq n$
5. $\quad$ return $\{a_m\} \cup$RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
6. else
7. $\quad$ return $\emptyset$

Let a_0 is a activity which finishes at $f_0 = 0$. We want to find out optimal solution $S_0 = S$, So first call we be RECURSIVE-ACTIVITY-SELECTOR(s, f, 0, n)

An iterative greedy algorithm

GREEDY-ACTIVITY-SELECTOR(s, f, n)

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4.   if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. return A

a_k is the most recent addition to A and activities are arranged in monotonically increasing finish time, so f_k is always the maximum finish time of any activity in A i.e. $f_k = \max\{f_i : a_i \in A\}$ and we are searching the next activity which starts after a_k finishes.

Correctness of the algorithm I

Is the greedy choice in which we choose the first activity to finish is always part of some optimal solution?

- Statement: Consider any nonempty subproblem S_k. let a_m be an activity in S_k with the earliest finish time. Then a_m is included in some maximum-size subset of mutually compatible activities of S_k
- Let A_k is a maximum-size subset of mutually compatible activities in S_k so $A_k \subseteq S_k$.
- let a_j is the activity in A_k with the earliest finish time
- Now there are two cases : either $a_j = a_m$ or we can replace a_j with a_m in A_k without affecting the solution.
- if $a_j = a_m$ then nothing to do and Statement is correct.
- if $a_j \neq a_m$, then Let another set $A_k^{'} = A_k - \{a_j\} \cup \{a_m\}$ that is substitute a_j with a_m.

Correctness of the algorithm II

- Now we have to check that is still $A_k^{'}$ is a maximum-size subset of mutually compatible activities.
- For that we have to check the new activity a_m is compatible with the rest of activities in $A_k^{'}$.
- As a_m is the first activity to finish in S_k and a_j is the first activity to finish in A_k and $A_k \subseteq S_k$ so either $a_j = a_m$ or $f_m \leq f_j$
- It means we can replace a_j with a_m and still set A_k will remain maximum sized non overlapping set of activities.
- So a_m will be the part of optimal solution.

Thank you

Please send your feedback or any queries to akyadav1@amity.edu, akyadav@akyadav.in or contact me on +91 9911375598