Activity Selection Problem using Greedy Method

Activity Selection Problem

Greedy Algorithms

We consider optimisation problems. Algorithms for optimization problems typically go through a sequence of steps, with a set of choices at each step.

> A *greedy algorithm* is a process that always makes the choice that looks best at the moment.

Greedy algorithms are natural, and in few cases solve optimally the given problem.

We will look at one simple example and we will try to understand why does it work.

We will consider more examples later.

Activity-selection problem

We are given a set of proposed activities $S = \{A_1, A_2, \ldots, A_n\}$ that wish to use a resource, which can be used by only one activity at a time. Each activity is defined by a pair consisting of a *start time* s_i and a *finish time* f_i, with $0 \leq s_i < f_i < +\infty$. If selected, activity A_i takes place during the time interval $[s_i, f_i)$. Two activities A_i and A_j are *compatible* if $s_i \geq f_j$ or $s_j \geq f_i$. The *activity-selection problem* is to select the maximum number of mutually compatible activities.

Example

Activity	St	Ft
L1	0	6
L2	3	5
L3	1	4
L4	3	8
L5	6	10
L6	5	7
L7	5	9
L8	2	13
L9	12	14
L10	8	11
L11	8	12

Solution using Greedy Method

Activity	St	Ft
L3	1	4
L2	3	5
L1	0	6
L6	5	7
L4	3	8
L7	5	9
L5	6	10
L10	8	11
L11	8	12
L8	2	13
L9	12	14

Solution using Greedy Method

- Solution Vector S = Ø

  S = {L3, L6, L10, L9}

Job Scheduling Problem/Job Sequencing Problem using Greedy Method

Job Scheduling Problem

- For a given set of n jobs there are n-deadlines and n-profits to be earned for any i^{th} Job, there is deadline di > 0 and profit Pi > 0
- Profit Pi is earned if job is completed within its deadline
- Assume that each job requires one time unit for completion and there exist a single machine.

Feasible Solution:

A set of jobs such that each job that can be completed by its deadline and value of feasible solution is sum of profits.

Example

Jobs	J1	J2	J3	J4
Profits	100	10	15	27
Deadline	2	1	2	1

Jobs can put in descending order according to profit

Jobs	J1	J4	J3	J2
Profits	100	27	15	10
Deadline	2	1	2	1

Final Solution

Deadline	1	2
Jobs	J4	J1
Profit	27	100

J = {J4, J1}

J = 27 + 100 = 127

Another Example

Jobs	J1	J2	J3	J4	J5	J6
Profits	**20**	**10**	**7**	**5**	**15**	**3**
Deadline	**2**	**1**	**3**	**1**	**1**	**3**

Final Solution

Deadline	1	2	3
Jobs	**J5**	**J1**	**J3**
Profit	**15**	**20**	**7**

J = {J5, J1, J3}

J = 15 + 20 + 7 = 42

GTU Paper Example

- Using greedy algorithm find an optimal schedule for following jobs with n=7

  profits: (P1,P2,P3,P4,P5,P6,P7) = (3,5,18,20,6,1,38) and

  deadline (d1,d2,d3,d4,d5,d6,d7) = (1,3,3,4,1,2,1)

Answer is : (J7,J6,J3,J4) --- (38+1+18+20) = 77