

Computational Methods Assignment

Task 5

The job-scheduling problem summarised is finding an optimised sequence of jobs to resources to complete an objective.

There are a certain number of jobs that has a deadline, and each job has a profit if it is done before the deadline.

Every job only has a single unit of time, meaning the minimum possible deadline for any job is 1. How do we maximise the total profit if only one job can be scheduled at a time?
(*Job Sequencing Problem, 2017*)

Due to its nature of being an NP complete problem, we can't find the best and fastest solution as there is no known efficient algorithm to solve this in polynomial time, however, we can use known algorithms, such as the greedy strategy, to generate an approximate, good enough solution.

The job-scheduling problem is significant because in the real world it can help optimise the allocation of limited resources by improving efficiency, reducing costs, maximising profit and making sure that tasks are completed on time in a range of industries such as industrial, engineering, computing and management.

When applying the greedy strategy to the job-scheduling problem, we can follow the following steps:

1. Set your overall deadline from the job with the largest deadline.
2. Sort your jobs array by descending order of profits.
3. Attempt to schedule each job as late as possible
 - a. If the job can't be done without missing the deadline, skip the job
4. Output the new set of jobs

(*Job Sequencing Problem, 2017; Job Sequencing with Deadline*)

When we have the following set of jobs, first we need to sort them based on profit, then:

JobId	A	B	C	D	E	Sorted Into →	JobId	A	C	D	B	E
Deadline	2	1	2	1	3		Deadline	2	2	1	1	3
Profit	100	19	27	25	15		Profit	100	27	25	19	15

Once we have the jobs sorted by profit, we can begin working our way through the algorithm assigning jobs into slots based on their deadlines.

	↓									
JobId	A	C	D	B	E			1s	2s	3s
Deadline	2	2	1	1	3	→	Answer:		A	
Profit	100	27	25	19	15					

	↓									
JobId	A	C	D	B	E			1s	2s	3s
Deadline	2	2	1	1	3	→	Answer:	C	A	
Profit	100	27	25	19	15					

	↓									
JobId	A	C	D	B	E			1s	2s	3s
Deadline	2	2	1	1	3	→	Answer:	C	A	E
Profit	100	27	25	19	15					

1. We can complete A as late as possible, at our 2 second deadline.
2. We can't complete C as late as possible, as A is already taking that slot, so we complete C before A.
3. The next two jobs, D and B must be completed before our 1s deadline, however since C is occupying this deadline, we skip them.
4. Our last job E has a deadline of 3s, since this slot is free, we can allocate the job.

When assuming the array of jobs is already sorted by profit, the previous example will run in $O(n^2)$ time.

Bibliography

GeeksforGeeks (2017) Job Sequencing Problem (Greedy Algorithm). 21st February 2017. Available at: <https://www.youtube.com/watch?v=R6Skj4bT1HE> (Accessed: 26th November 2024)

tutorialspoint (no date) Job Sequencing with Deadline. Available at: https://www.tutorialspoint.com/data_structures_algorithms/job_sequencing_with_deadline.htm (Accessed: 26 November 2024).