

CSE 3512 – Algorithm Design and Analysis

Lab 6

Job Sequencing with Deadlines (Branch and Bound)

Problem Statement 1

Given a set of n jobs, where each job i has:

- Profit $p[i]$
- Deadline $d[i]$

Each job requires exactly one unit of processing time, and only one job can be scheduled at a time. A job must be completed on or before its deadline. The objective is to maximize total profit. You are required to solve this problem using the Branch and Bound technique.

Task 1: Implementation of job sequencing with deadlines using Branch and Bound

1. Design a state-space tree where each node represents a partial selection of jobs for a job sequencing with deadlines problem with number of jobs: $n = 5$; Profit: $(p_1, p_2, p_3, p_4, p_5) = (6, 3, 4, 8, 5)$; Deadline: $(d_1, d_2, d_3, d_4, d_5) = (3, 1, 4, 2, 4)$. Each job requires one unit of processing time.
2. Define:
 - a. Branching rule (selecting or rejecting a job)
 - b. Bounding function to compute an upper bound on achievable profit
3. Use the bound to prune non-promising nodes.
4. Implement the algorithm in C++.

Input Format:

```
n  
p[1..n] // profits  
d[1..n] // deadlines
```

Output Format:

Maximum Profit
Selected Job Sequence

Sample Input

```
5 //instances  
6 3 4 8 5 //profits  
3 1 4 2 4 //deadlines
```

Sample Output

Maximum Profit = 23
Selected Jobs (Time Slot : Job ID)
Slot 1 : Job 3
Slot 2 : Job 4
Slot 3 : Job 1
Slot 4 : Job 5

Task 2: Input Size vs Time Analysis

1. Generate job instances of different sizes (for example: $n = 10, 20, 30, 40, 50$) with corresponding profit and deadline vectors.
2. Measure the execution time of your Branch and Bound algorithm for each input size.
3. Tabulate the results showing:
 - a. Number of jobs (n)
 - b. Execution time
4. Plot and analyze how execution time grows with increasing input size.

Discussion (for report writing):

- Why does Branch and Bound perform better than brute force?
- How does the bounding strategy affect runtime?
- Practical limitations of Branch and Bound for large inputs.

Problem Statement 2:

A cloud service provider receives multiple computational tasks from clients. Each task:

- requires 1 unit of processing time
- must be completed before a given deadline
- provides a certain revenue upon successful completion

Due to limited resources, the server can execute only one task at a time. The objective is to maximize total revenue while meeting all deadlines.

Task 3: Apply your Branch and Bound implementation from **Problem 1** to this scenario and display: Selected tasks, Execution order, and Total revenue earned.