Job Scheduling Problem Write Up:

Theory:

1. The Job Scheduling Problem (JSP) is a well-known combinatorial optimization problem that deals with scheduling a set of jobs on available resources to optimize a given objective.
2. In its basic form, JSP involves determining the order in which the jobs should be processed on the available resources, considering constraints such as job durations, resource availability, and job dependencies.
3. The objective is usually to maximize the profit, within is the total time given to complete all jobs.
4. However, in the variant of JSP with deadlines and profit objectives, an additional dimension is introduced.
5. Each job is associated with a deadline, indicating the latest time by which the job must be completed, and a profit, indicating the value or revenue generated by completing the job.
6. The objective in this variant is to maximize the total profit obtained by scheduling the jobs in a way that meets their respective deadlines.

The problem can be mathematically formulated as follows:

- Given:
  - A set of n jobs, where each job j has a processing time or duration p_j.
  - A set of m resources available for scheduling the jobs.
  - For each job j, a deadline d_j indicating the latest time by which the job must be completed.
  - For each job j, a profit or revenue value v_j associated with completing the job.

- Objective:
  - Maximize the total profit obtained by scheduling the jobs while ensuring that each job is completed before its deadline.

- Constraints:
  - Each job can only be processed on one resource at a time.
  - The total processing time on each resource should not exceed the resource's capacity.
  - Each job must be completed before its respective deadline.

Input: *Four Jobs with following deadlines and profits*

JobID	Deadline	Profit
a	4	20
b	1	10
c	1	40
d	1	30

Output: *Following is maximum profit sequence of jobs: c, a*

Input: *Five Jobs with following deadlines and profits*

JobID	Deadline	Profit
a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

Output: *Following is maximum profit sequence of jobs: c, a, e*

Performance Analysis:

The time and space complexity of solving the Job Scheduling Problem (JSP) depends on the specific algorithm or approach used. Here, I will provide an overview of the complexities associated with two common approaches: the Job Scheduling algorithm and the Genetic Algorithm.

Job Scheduling Algorithm:

- The time complexity of the Job Scheduling algorithm for JSP is O(n^2), where n is the number of jobs. This is because the algorithm requires sorting the jobs based on their processing times, which has a time complexity of O(n log n) using efficient sorting algorithms like quicksort or merge sort. Additionally, the algorithm iterates over the sorted list of jobs, assigning each job to a resource and updating the schedule. Since there are n jobs and the algorithm iterates over each job once, the overall time complexity is O(n).
- The space complexity of the Job Scheduling algorithm is O(n), as it requires storing the schedule information for each job. Additionally, a separate data structure may be needed to keep track of the available time slots on the resources. Additionally, the performance of SSSP algorithms can be affected by

graph characteristics, such as the number of vertices, the number of edges, the density of edges, and the distribution of edge weights.

Conclusion:

In conclusion, the Job Scheduling Problem poses a significant challenge in optimizing job sequencing while considering deadlines and profit objectives. Various algorithms and heuristics have been proposed, each with its strengths and limitations. Further research is needed to develop efficient and effective approaches for real-world applications.