Sequencing Problems in Operations Research

Sequencing (or sequential) problems in operations research involve determining the best order to process a set of jobs on one or more machines to optimize objectives such as minimizing total completion time (makespan), machine idle time, or lateness.

Types of Sequencing Problems

1. n-job 1-machine Problem

- **Description:**
  n jobs are to be processed one after another on a single machine.
- **Objective:**
  Minimize total completion time, lateness, or tardiness.
- **Solution Approach:**
  Use rules such as Shortest Processing Time (SPT) or Earliest Due Date (EDD) to sequence jobs. Since only one machine is involved, the problem is straightforward and jobs are processed sequentially without overlap.

2. n-job 2-machine Problem

- **Description:**
  n jobs must each be processed on two machines in the same order (e.g., first on Machine A, then on Machine B).
- **Objective:**
  Minimize the total elapsed time (makespan).
- **Solution Approach:**
  Use **Johnson's Algorithm**:
  - Find the job with the smallest processing time among all jobs on both machines.
  - If the smallest is on the first machine, schedule it as early as possible; if on the second, schedule it as late as possible.
  - Remove the scheduled job and repeat until all jobs are sequenced.
    This ensures the same order on both machines and minimizes idle time on the second machine.

3. n-job 3-machine Problem

- **Description:**
  n jobs are processed in the same order on three machines (A, B, and C).
- **Objective:**
  Minimize the total elapsed time.
- **Solution Approach:**
  If certain conditions are met (e.g., the minimum processing time on the first or third machine is greater than or equal to the maximum on the second), the problem can be reduced to an equivalent two-machine problem by creating two "dummy" machines:
  - For each job, calculate:
    - Dummy G = processing time on A + B
    - Dummy H = processing time on B + C
  - Then, apply Johnson's Algorithm to this transformed problem.
    If these conditions are not met, more complex or computational methods may be needed.

4. n-job m-machine Problem

- **Description:**
  n jobs must be processed on m machines (where m > 2), each in the same order.
- **Objective:**
  Minimize total elapsed time.
- **Solution Approach:**
  Johnson's Rule does not directly apply for m > 2 unless the problem can be reduced as in the 3-machine case. For general m-machine problems, heuristic or computational algorithms are used. Sometimes, the machines are grouped into two "super-machines" to reduce the problem to a two-machine case if possible.

5. 2-job m-machine Problem

- **Description:**
  Only two jobs, each to be processed through m machines in the same order.
- **Objective:**
  Minimize total elapsed time.

- **Solution Approach:**
  Since there are only two jobs, the problem size is small and can be solved by enumeration or graphical methods. Each possible sequence is evaluated to find the one that minimizes the makespan.