Framework for Solving Scheduling Problems Project
Group - 16 Report

IE 439

Machine Scheduling

Course Instructor

Utku Koç

Prepared by

Elif Cengiz (042003057)

Elif İlayda Güntürk (042003028)

Turgut Berk Karahasanoğlu (042003037)

1. INTRODUCTION

1.1. Overview of the Project

This project is mainly focused on solving scheduling issues in single and parallel machines. The objective is to build a system with a scheduling function that can compute several objectives and present the outcomes in the form of a Gantt chart. Some of the dispatching rules used in the project are the Shortest Processing Time (SPT), the Earliest Due Date (EDD) and the Longest Processing Time (LPT) and the project extends the above to parallel machines for complex scenarios. Secondly, local search is used to find better solutions.

1.2. Key Objectives and Goals

The main goal of this project is to design a scheduling system for the single machine and the parallel machines. This tool should be capable of:

- Calculating Various Scheduling Objectives: Several goals of scheduling like makespan, total completion time, total tardiness and weighted completion time will be shown as an output.
- Visualizing Schedules: The tool will automatically create Gantt charts to represent the job schedules with respect to which scheduling decisions can be easily analysed and modified.
- Applying Dispatching Rules: It shall use various dispatching rules of Shortest Processing Time (SPT), Earliest Due Date (EDD), Longest Processing time (LPT) for job sequencing effectively. They will be applied to various job sets to measure the effects on scheduling.
- Extending to Parallel Machine Setup: The tool will be expanded in the sense that it will be able to handle parallel machines where some jobs can be scheduled to run at the same time but under the selected rules.
- Optimizing Schedules Using Meta-Heuristics: The last of these is to apply meta-heuristic optimization methods, including local search procedures, to the generated schedules.

2. FRAMEWORK FOR SINGLE MACHINE SCHEDULING

2.1. Objectives for Scheduling Optimization

The goal of single machine scheduling is therefore to schedule performances towards particular goals. These include:

- Makespan: The total amount of time spent on all jobs should be the shortest time possible.
- Total Completion Time: Minimise the total aggregate of completion time across all jobs.
- Total Tardiness: Reduce the total amount by which job due dates are exceeded.
- Total Weighted Completion Time: Minimize the weighted completion time, with regards to job urgency.
- Maximum Lateness: The objective for the tardiness measure is to seek the case where the maximum deviation of a job's completion time from its due date is minimal.

2.2. Input Parameters and Sequence Handling

The scheduling system is designed to accept several input parameters for job sequences, which include:

- Job Number: Every job is reflected in a number assigned to it.
- Process Time: The number of hours estimated to be taken to finish every set job.
- Due Date: The idea of the client's preferred time for completion of each of the jobs.
- Weight: A value attached to each job that is used to do the weighted completion time allocation.
- Release Time: The opportunity at which the job can be processed through the system.

The framework takes as input a prescribed order of jobs and proceeds with the processing of these jobs based on the input parameters. There is also an option to define one's own sequence, this means that in the issue of job scheduling flexibility is concerned. The sequence handling function further enhances proper orders of jobs and; the jobs are executed following the indicated rule, or the sequence user-determined.

	job_number	process_time	due_date	weight	release
0	1	10	4	14	0
1	2	10	2	12	0
2	3	13	1	1	0
3	4	4	12	12	0

Figure 1. Job Scheduling Data Overview

2.3. Calculation of Completion Times and Evaluation Metrics

After the jobs have been sequenced, the framework estimates completion time for each job in reference to the sequence. The following steps are taken:

1. Completion Time Calculation: Each job is processed according to its processing time and the sum of this time with the previous job's time given for the release time.
2. Objective Evaluation: Once completion times of activities are computed the schedule constraints are assessed through achievement of the selected objectives (such as makespan, total time etc.)

Each of these measures is automatically computed and reviewed by the system and users to determine the effectiveness of the present job sequence.

2.4. Deliverable: Flexible Framework for Single Machine Scheduling

The output for this component is a working structure that allows users to input job sequences and consider more than a single scheduling objective. It shall infer the time to completion based on the job order within the framework and then infers the objective function values of interest. This tool will be able to turn any sequence of jobs, which is typical of a specific operation, into the kind of scheduling criteria that would allow the users to contemplate changes in the framework to suit the uniqueness of an operation.

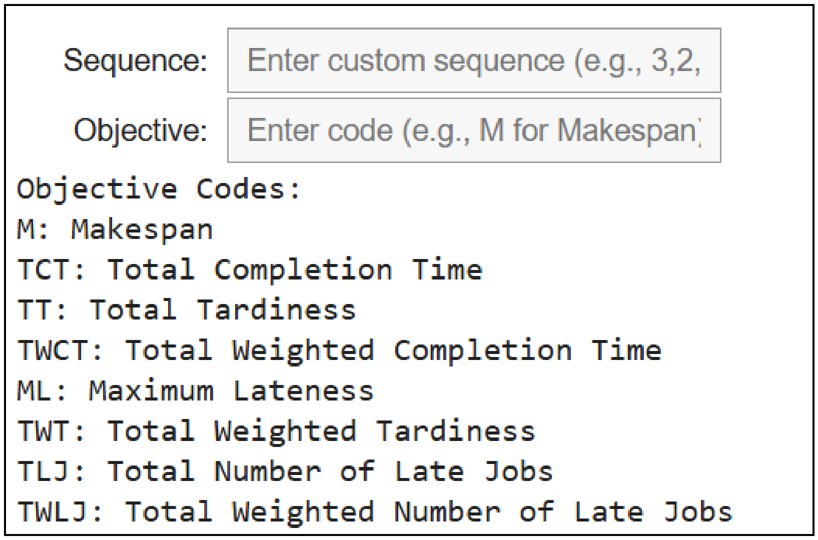

Figure 2. User Input for Job Sequencing and Objective Selection

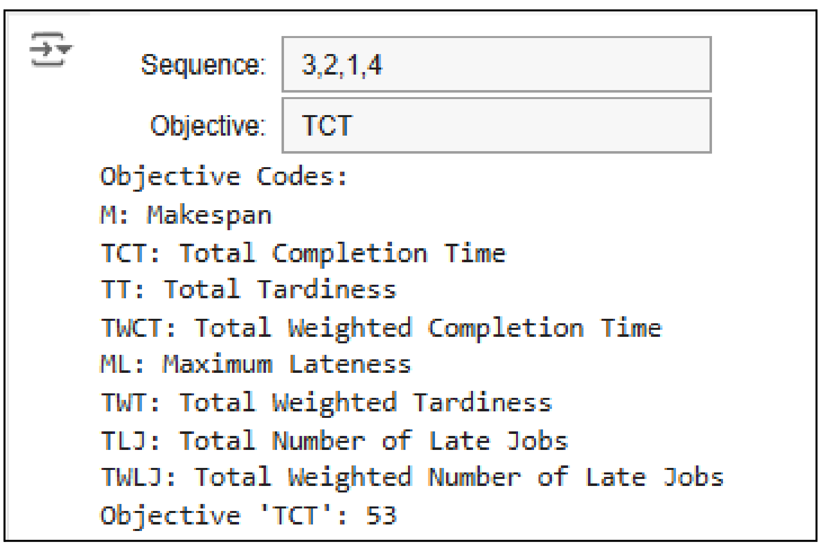

Figure 3. Objective Calculation for Selected Sequence

3. GANTT CHART GENERATOR

3.1. Overview of Visualization Tools for Scheduling

The Gantt chart generator in this project is used to take the job scheduling data and turn it into a format that is easy to understand so that various scheduling rules may be compared and the effects on performance variables analyzed.

3.2. Design of the Gantt Chart Input Format

The Gantt chart generator takes job scheduling data as input, including the following key parameters:

- Job ID: A reference number for each job.
- Start Time: The period when a job first touches on a machine.
- End Time: This is the time when a job stops processing.
- Machine ID: The specific equipment used in processing a particular job.
- Processing Time: The time it takes to do each of the jobs envisaged.
- Job Details: Other kinds of information like due dates, weight of the job which can be used further for analysis or for labeling of charts.

3.3. Development of Chart Structure and Labeling

The structure of the Gantt chart development means placing job data on a timeline section. Each job is represented by a horizontal bar, where:

- Length of the bar represents the time which takes to process a certain job.
- The position of the bar on the x-axis is the time at which the job begins, according to the data.

Labels are included within the chart for clarity, showing key information such as:

- Job ID: They are also placed side by side with the corresponding bar graph to distinguish the differences.
- Start and End Time: Shown above or below the bar depending on the correct timing.
- Machine ID: Shown on top of y-axis to depict on which machine the job is processed.
- Processing Time: Also written at the same line of bar for convenience.

3.4. Advanced Features and Testing

The Gantt chart generator can be enhanced with additional features such as:

- Interactive Elements: Occupations remain the same as in the previous design; users can view detailed job information by halting the pointer over bars or choose jobs according to the filters such as priority or machine.
- Customizable Appearance: The chart can be modified to fit in a number of requirements including the color of the bars, label formats and axes if needs be.
- Error Handling and Validation: It validates the input parameters of the job by scanning for bidirectional processing time numbers and negative start times and other similar issues.

Several job sets and scheduling sequences are actually performed to ensure that the Gantt chart reflects the schedules of the jobs correctly and all the necessary details are well captured in the chart.

3.5. Deliverable: Gantt Chart Generator for Job Scheduling

The deliverable for this component is a fully functional Gantt chart generator that:

- It receives scheduling data for jobs and forms a schedule graphic from it.
- Enables users to define labels for jobs and format of charts.
- Supplements include features such as interactivity in revealing further job information.
- Should be cross validated against datasets of different employment categories to gain specification and reliability of the graphic interface used for scheduling.

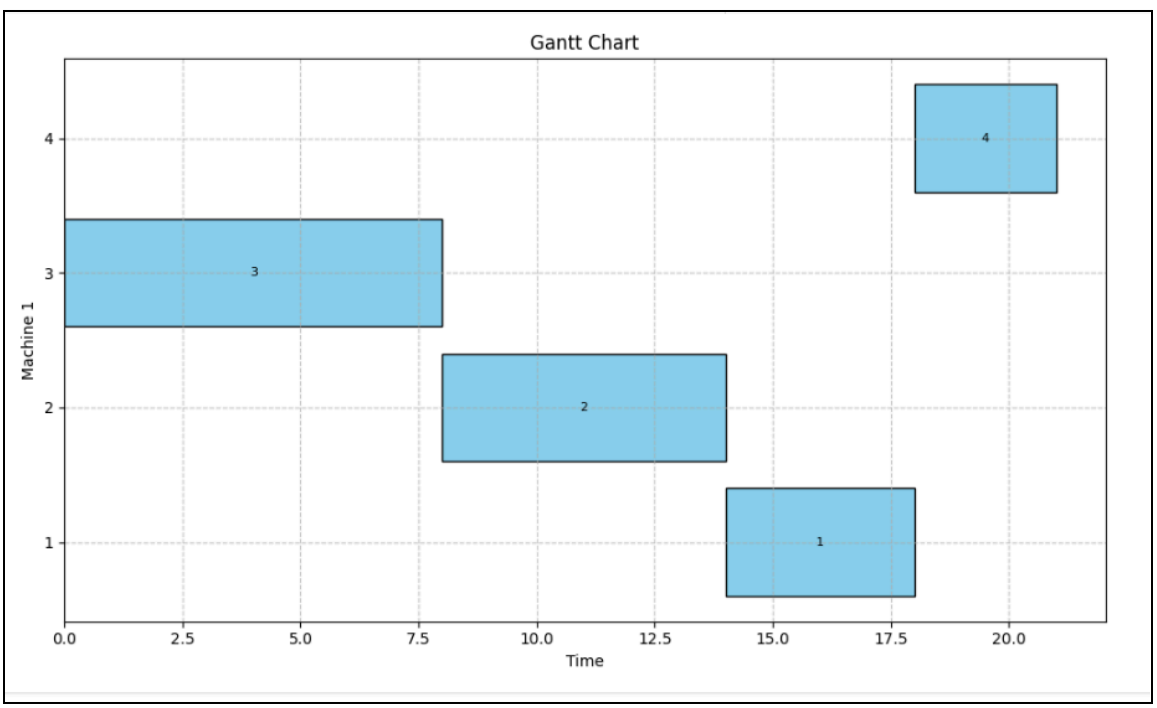

Figure 4. Job Scheduling Gantt Chart for Sequence 3,2,1,4

4. APPLICATION OF DISPATCHING RULES

4.1. Dispatching Rules for Job Sequencing

Dispatching rules refer to the procedures used in developing the sequence of jobs that are to be processed in a particular machine. These rules assist in decisions concerning the prioritization of jobs through factors such as duration in processing the job, the due dates among other factors. In fact, selection of the appropriate dispatching rule also has a major impact on the scheduling performance, in terms of makespan, tardiness, or machine utilization.

4.2. Key Dispatching Rules: SPT, EDD, ERD, and LPT

The following key dispatching rules are implemented in the project to sequence jobs:

- Shortest Processing Time (SPT): This rule is also known as SJF and as its connotes it involves giving priority to the jobs that take the least time to be processed. SPT prov ides the benefit of reducing both the total completion time and makespan.
- Earliest Due Date (EDD): Scheduling of jobs is in order where jobs with the shortest due date are given priority over the longer ones. Of all the measures discussed above, EDD is especially effective concerning reduction of actual tardiness.
- Earliest Release Date (ERD): Work is prioritized according to its listing date; thus, jobs are handled based on the time the work becomes accessible. This rule ensures that no work is carried out before that work is ordered or requested by its owner.
- Longest Processing Time (LPT): This rule gives preference to those tasks that take longer time to be processed so that they are attended to first. With LPT, one can exclude the number of jobs that can lead to a possible break in that schedule.’

All of these rules are geared towards achieving different scheduling aims and the correct rule can thus be selected based on the concrete optimization target.

4.3. Implementation and Evaluation of Scheduling Sequences

The said rules are provided so as to produce the schedule of the given set of jobs in the context of the developed framework. Overall each sequenced event is dispatched according to the specified rule and is assessed with respect to key performance measures involving makespan, total time of completion, and tardiness. This means that the user can compare the results of different rules when implemented under different circumstances.

Each time a sequence is created using a specified dispatching rule, the system evaluates the objective function of the resulting schedule. These values are used in decision making in order to determine which scheduling rule gives the best performance for a given set of jobs.

4.4. Deliverable: Tool for Applying Dispatching Rules

The product of this type of learning materials section is an output tool that includes the chosen dispatching rules assigned to a set of jobs and produces the schedule for each job. The tool also considers the performance metrics of the different schedules for purposes of comparison before presenting the findings. This enables users to make proper comparison and fast on which rule can best suit their scheduling needs.

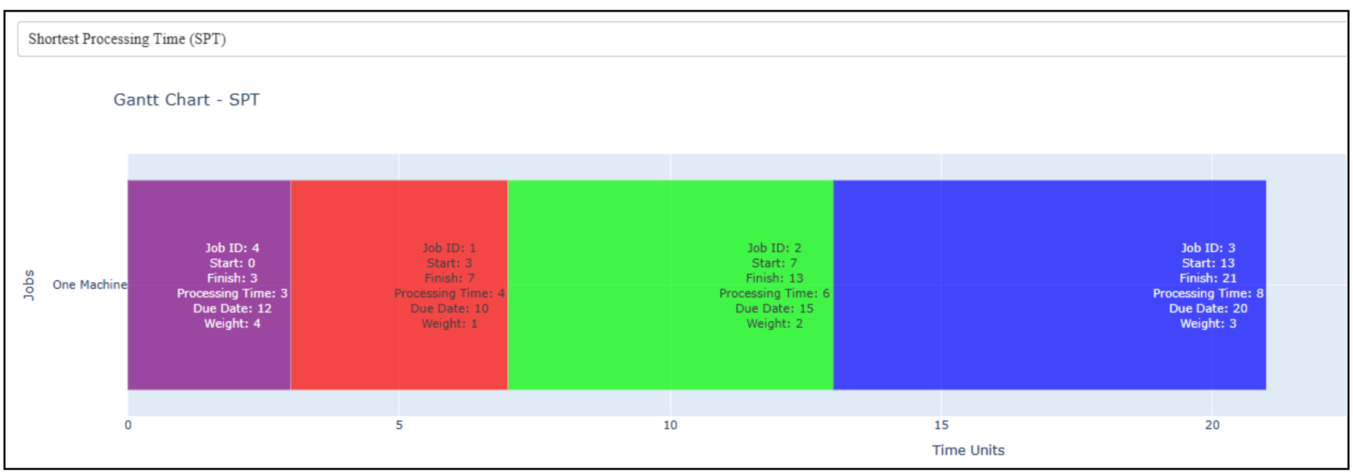

Figure 5. Gantt Chart for Job Scheduling using Shortest Processing Time (SPT)

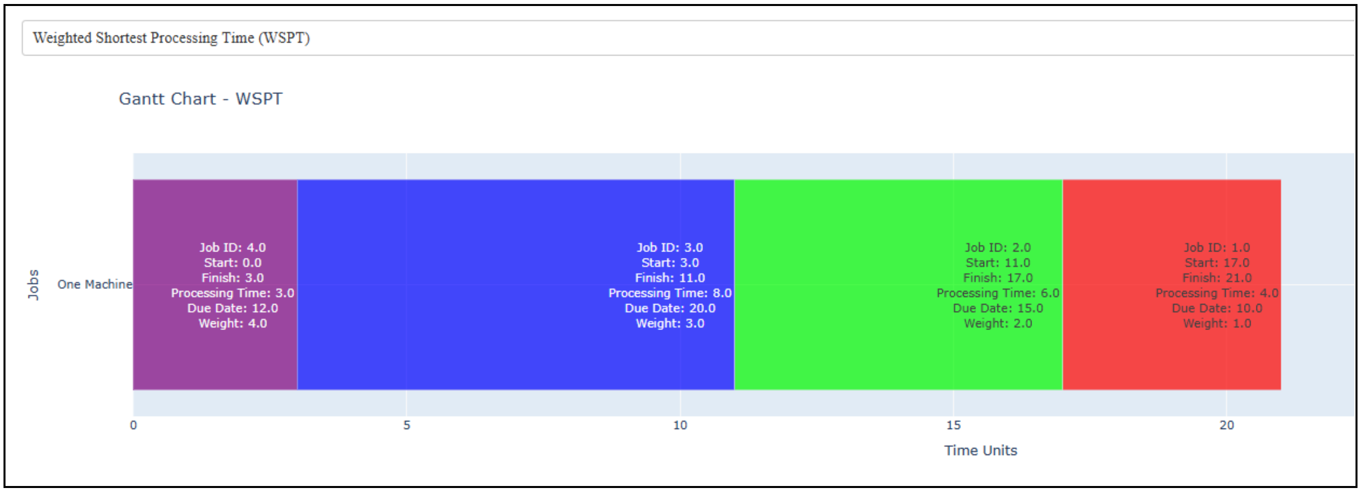

Figure 6. Gantt Chart for Job Scheduling using Weighted Shortest Processing Time (WSPT)

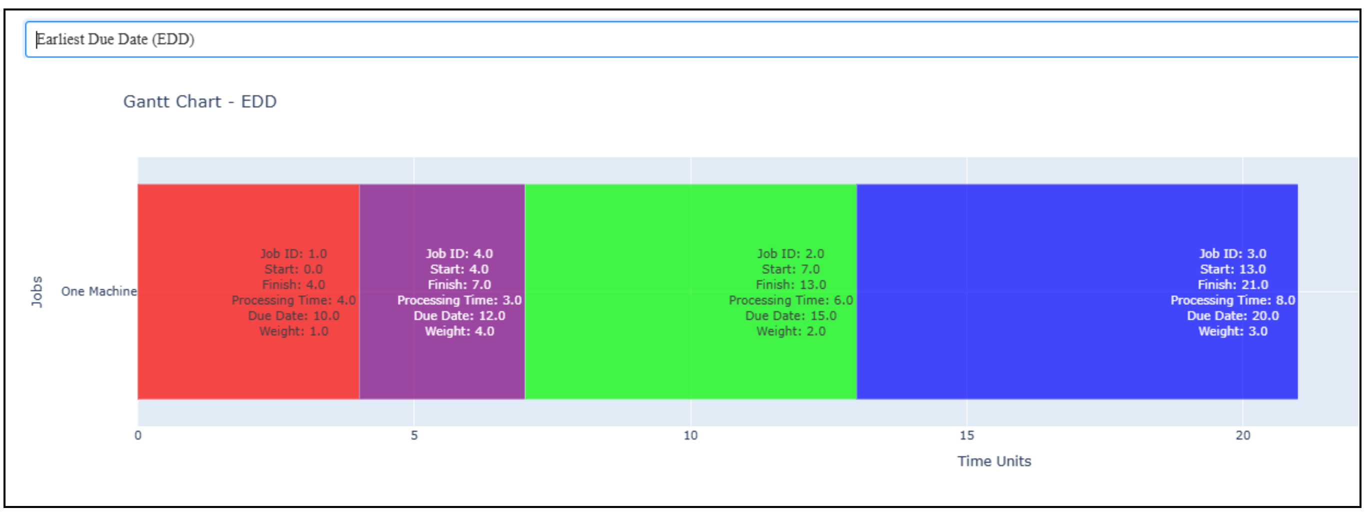

Figure 7. Gantt Chart for Job Scheduling using Earliest Due Date (EDD)

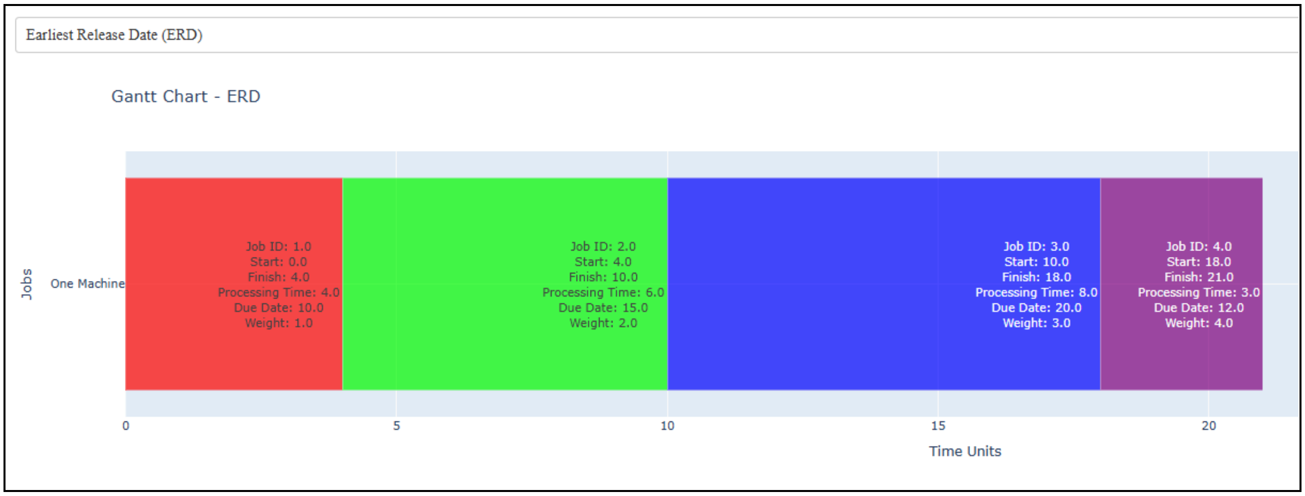

Figure 8. Gantt Chart for Job Scheduling using Earliest Release Date (ERD)

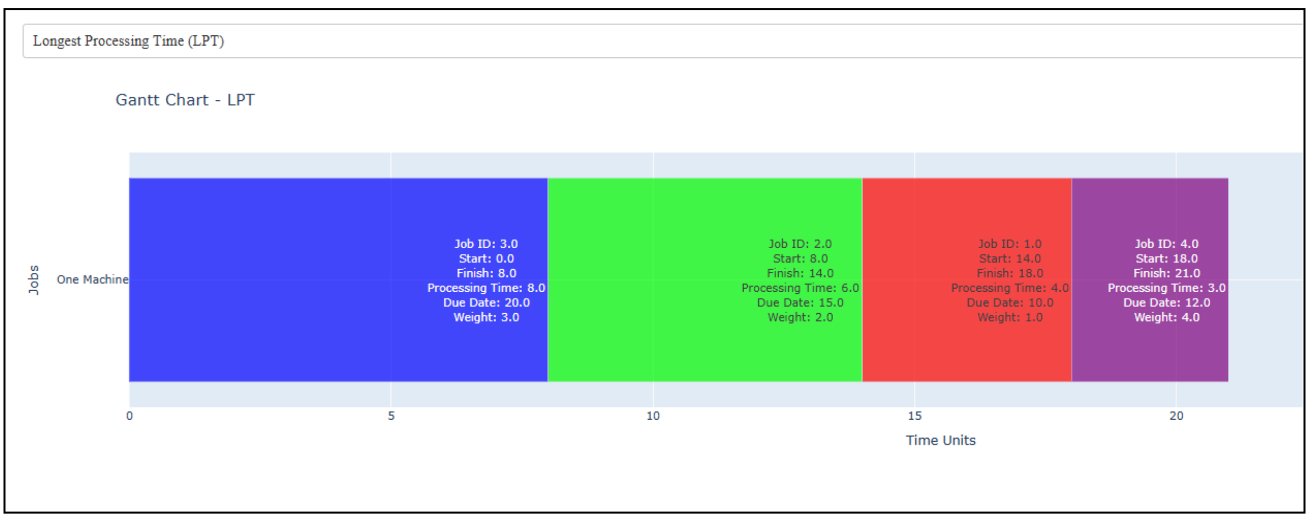

Figure 9. Gantt Chart for Job Scheduling using Longest Processing Time (LPT)

5. SCHEDULING FOR PARALLEL MACHINES

5.1. Parallel Machine Setup

Scheduling for parallel machines is not similar to scheduling for single machine scheduling since the jobs require to be scheduled across various machines. In this case, the system deals with several machines that undertake several jobs at once.

The setup involves:

- Determining number of parallel machines to be in existence.
- Taking the capability of the machine in mind for instance in the area of processing speed and efficiency.

To accommodate the multiple machines during scheduling and to allocate every job to an ideal machine during scheduling the framework incorporates the scheduling logic formulated in our earlier work.

5.2. Assignment Rules and Machine Allocation Techniques

Depending on the jobs to be assigned to parallel machines there are different techniques that can be applied. The following assignment rules are typically applied:

Shortest Processing Time (SPT): Shorter processing-time jobs are loaded on machines as they become vacant, which is useful in equal distribution of workload.

Longest Processing Time (LPT): Longer processing time jobs are allocated to the machines first with the goal of minimizing machine idle time..

5.3. Deliverable: Scheduling Tool for Parallel Machine Environments

Thus, the deliverable for this component is an effective tool to track and coordinate job schedules in parallel machines. This tool enables users to enter job data and machine capacity, choose the requisite scheduling rules and obtain the best job schedule. The tool will also pre-plan the job schedules on parallel machines whereby it will generate a Gantt chart, to check on any imaginable bottlenecks.

The feasibility of the scheduling tool will be evaluated by experimenting with various machines and job sequences to demonstrate versatility and stability in various forms of parallel machine scenarios.

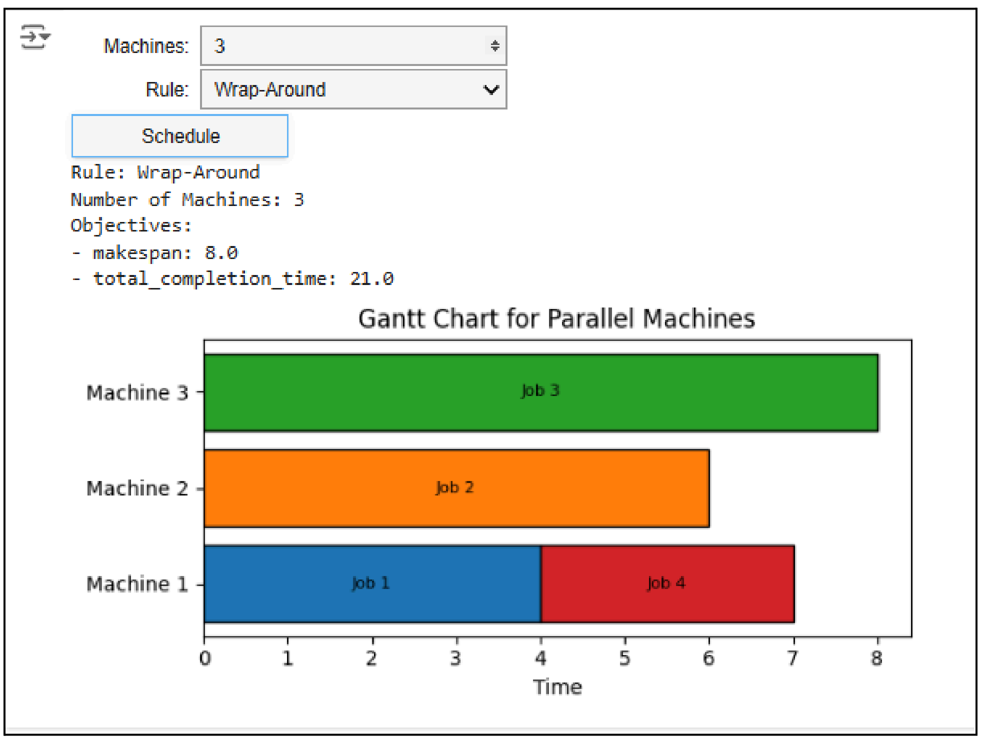

Figure 10. Gantt Chart for Parallel Machines using the Wrap-Around Rule

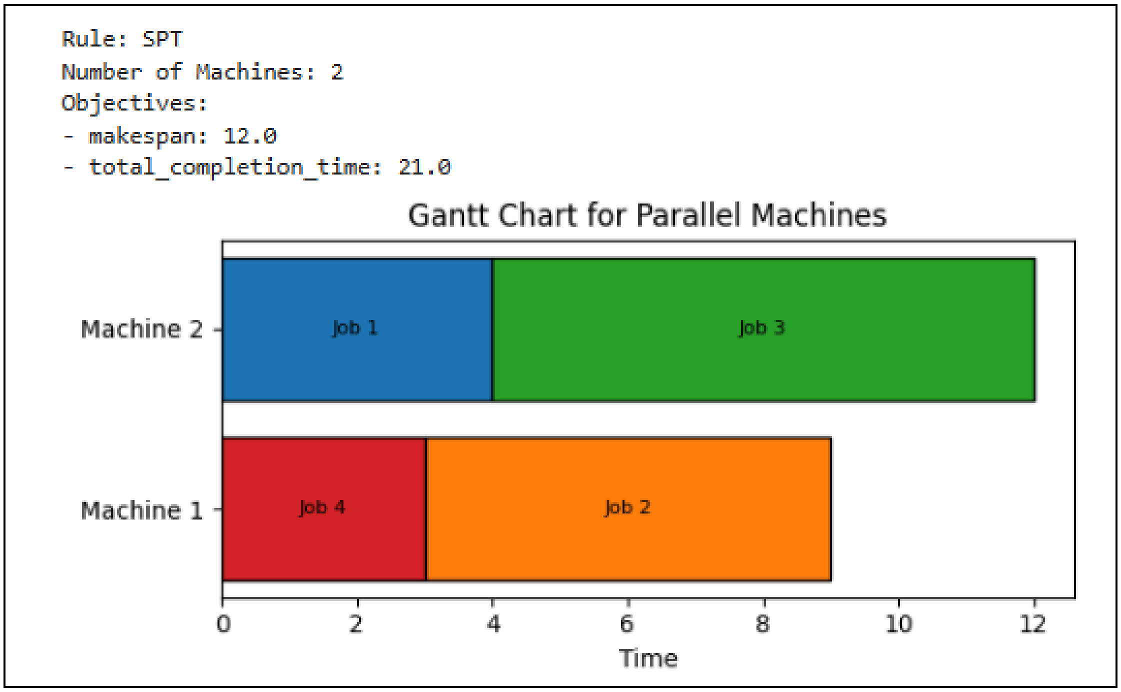

Figure 11. Gantt Chart for Parallel Machines using the Shortest Processing Time (SPT) Rule

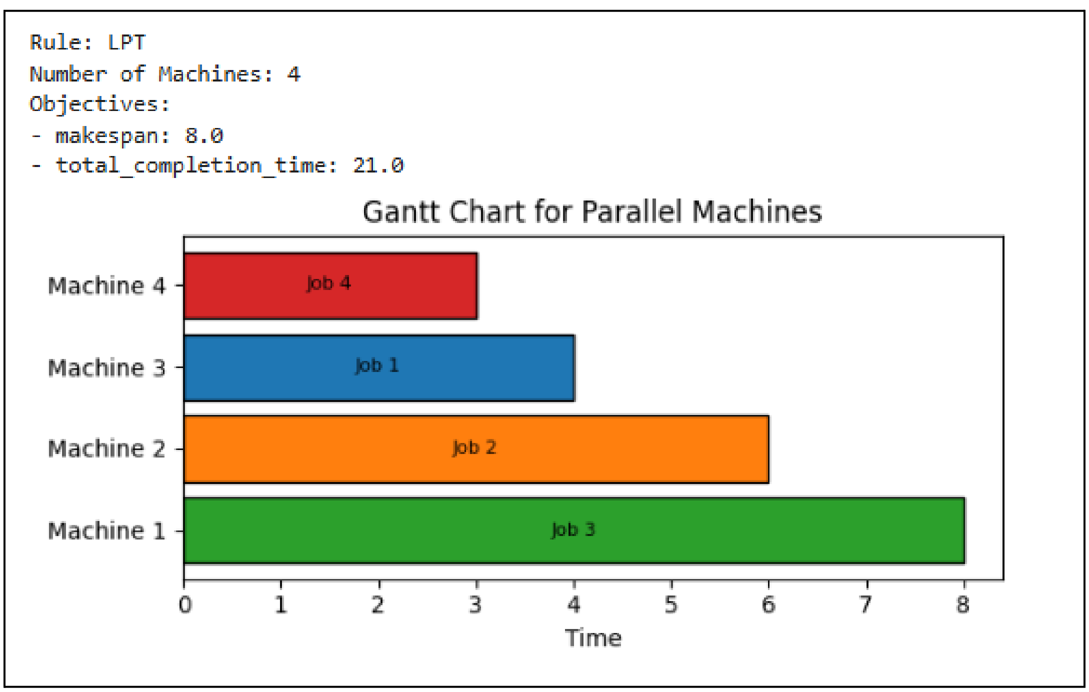

Figure 12. Gantt Chart for Parallel Machines using the Longest Processing Time (LPT) Rule

6. OPTIMIZATION TECHNIQUES USING META-HEURISTICS

6.1. Local Search Techniques: Swap, Remove and Insert, Adjacent Swap

In this project, local search techniques are used in search of better sequences for jobs in iteration through the solution space. These techniques include:

- Adjacent Swap: Jobs that are adjacent in the sequence are swapped, to evaluate for performance improvement relative to the scheduling goals.

Through these local search methods, the project intends to find better job sequences in order to minimize the completion time, lateness, or any other performance measure.

6.2. Search and Acceptance Criteria

Scheduling is a process of identifying various job combinations and assessing them in relation to advanced scheduling goals. The acceptance criteria are therefore anchored on the selected change in the objective value. For example:

- Improvement in Objective: As mentioned earlier if the new sequence reduces the makespan or decreases the level of tardiness it is considered.
- Stochastic Acceptance: In some cases it might even not improve on the previous one and yet it'll be accepted probabilistically to avoid local optima and give it a try in other regions of the solution space.

This process is repeated indefinitely if more results can be achieved until we reach a state where it can no longer be optimized, or a stop criterion is reached (number of iterations, etc.).

6.3. Results and Improvements Demonstrated

Following the usage of the local search techniques, the system is run on a number of scheduling scenarios and the results are compared before and after the optimization. The improvements are measured based on the selected objectives such as:

- Makespan: Saving the total time that is needed to accomplish all the jobs.
- Total Completion Time: Reduction of the total of completion times.
- Tardiness: Reduced accumulated total tardiness of all the jobs.

The results show that meta-heuristic optimization outperforms simple dispatching rules when it comes to scheduling since the job order is adjusted to achieve more objectives at a time.

```
Select an objective (M, TCT, TT, TWCT, ML, TWT, TWLJ):
```

Figure 13.

```
Select an objective (M, TCT, TT, TWCT, ML, TWT, TWLJ): TWT
Initial Sequence: [2, 3, 4, 1]
Initial Solution: [2, 3, 4, 1] -> TWT: 31.0
Sequence: [3, 2, 4, 1] -> TWT: 31.0
Sequence: [2, 4, 3, 1] -> TWT: 11.0
Sequence: [2, 3, 1, 4] -> TWT: 44.0
Accepted Sequence: [2, 4, 3, 1] -> TWT: 11.0
Sequence: [4, 2, 3, 1] -> TWT: 11.0
Sequence: [2, 3, 4, 1] -> TWT: 31.0
Sequence: [2, 4, 1, 3] -> TWT: 6.0
Accepted Sequence: [2, 4, 1, 3] -> TWT: 6.0
Sequence: [4, 2, 1, 3] -> TWT: 6.0
Sequence: [2, 1, 4, 3] -> TWT: 7.0
Sequence: [2, 4, 3, 1] -> TWT: 11.0
No improvement found. Stopping.
Best Sequence: [2, 4, 1, 3]
```

Figure 14.