Production System Modelling MEC-E7001

Assignment 3

Optimizing flow shop

Name: Nguyen Xuan Binh

Student ID: 887799

Mail: binh.nguyen@aalto.fi

Group: 17

Date: 19.02.2024

1. INTRODUCTION

The purpose of the report is to study how the job schedule can be optimized in a four-machine flow shop using 10 jobs, each job having processing time and deadline as input data. Flow shops play a key role in the manufacturing environment, where continuous operation through subsequent processing processes is essential. A single process flow for all objects in such systems allows queuing and overtaking, opening requires a very complex mixed integer linear programming problem.

The comparison of four different scheduling strategies—First-In-First-Out (FIFO), Earliest Due Date (EDD), Total Tardiness Minimization, and Maximum Tardiness Minimization—is the main emphasis of this report. Every technique provides a distinct way to handle workflow and has a different effect on key performance metrics including average throughput time, maximum tardiness, total tardiness, and machine usage.

Furthermore, the report will also run analysis for the multi-objective optimization model that uses weighted throughput time and total tardiness. This approach will tell us how different schedulings can affect the flow shop performance, and we can tune the weight between different metrics to achieve the average best result in a flow shop job sequence

2. PROBLEM AND MODEL

Modeling method and software

The softwares that I used are Excel, Python (some light processing), and CPLEX solver. The "CPLEX Studio IDE" is the Integrated Development Environment for CPLEX, and it is the main program where we can open, edit, and solve OPL (Optimization Programming Language) files.

Since flow shops with more than 2 machines can not be optimized with simple rules such as the Johnson algorithm, we need to obtain the optimization model from the Manne's single machine model, which is an IP model, by adding indices for the machines and determining order of machines in the process. Furthermore, tardiness is only counted for the last step in this model

There are various relevant optimization criteria, which are total tardiness, max tardiness, makespan, total throughput time and utilization rate. We need to know their definitions first for a single machine. The same concepts can be extended for all machines from 1 to K machines.

- **Total Tardiness (TT)**:

Total tardiness is the sum of the delays of all jobs beyond their due dates. If D_i is the due date for job i, t_i is the starting time for the job i, and P_i is the processing time for job i, then tardiness f_i for job i is given by $f_i = \max(t_i + P_i - D_i, 0)$.

The total tardiness is then: $TT = \sum_{i=1}^{n} f_i$

- **Maximum Tardiness (MT):**

Maximum tardiness is the largest delay of any single job beyond its due date. Using the same notation as above:

$$MT = \max_{i=1}^{n} \{ f_i \}$$

- **Makespan (C_{max}):**

Makespan is the total time required to complete all jobs, which is also the finishing time of the last job in the schedule. We have P_i as processing time for each job i, then the makespan is:

$$C_{max} = \max_{i=1}^{n} \{ P_i \}$$

- **Total Throughput Time (TTT):**

Total throughput time is the sum of the times taken to process each job from its start to completion. If t_i is the start time for job i and C_i is the finished time for job i, then the throughput time for job i is $C_i - t_i$. The total throughput time is:

$$TTT = \sum_{i=1}^{n} \{ C_i - t_i \}$$

- **Utilization Rate (UR):**

Utilization rate is the ratio of the active processing time to the total available time for a machine or set of machines. If P_{ij} is the processing time of job i on machine j, and K is the number of machines, the utilization rate for machine j is:

$$UR_j = \left(\sum_{i=1}^{n} \{ P_{ij} \} \right) / C_{max}$$

For all machines:

$$UR = (1/K) * \sum_{j=1}^{K} \{ UR_j \}$$

Model description

Expanding on Manne's single machine model, which has been adapted for a flow shop environment, our major purpose now is to arrange a set of jobs, each of which must pass through a series of steps. Extension to parallel machines will imply more considerations.

1. Sequence preservation: In a flow shop, each task must follow the same set of machines. The model requires that a work finishes its operation on machine k before being processed on machine k+1.

2. Job Dependencies: The model must consider the dependencies between jobs. A job can only start on a machine if the preceding job has been completed on that machine. This is where binary variables become essential, as they define the processing sequence of the jobs.

3. Utilization rate: Because we are now considering the case of parallel machines, the utilization rate of each machine is an important indicator. The model must try to keep all machines as occupied as possible while minimizing idle time.

4. Tardiness: By calculating tardiness only at the end machine, the model shows that intermediate deadlines are less important than final delivery time. However, keeping in mind intermediate completion times could be useful for detecting delays early on and allocating tasks as needed.

Now, we have the extension to Manne's formulation of the IP model, which generates starting time of the jobs for each machine k.

IP model, $k,..,K$ is machine index

$\text{Min} \sum_{\forall i} f_i$	
$t_{iK} + P_{iK} - D_i \leq f_i,$	$\forall i$
$t_{ik} \geq 0, f_i \geq 0,$	$\forall i, k$
$t_{ik} + P_{ik} \leq t_{ik+1},$	$\forall i, k \in \{1,.., K-1\}$
$My_{ijk} + (t_{ik} - t_{jk}) \geq P_{jk},$	$\forall i \in \{1,.., I-1\}, j \in \{i+1,.., I\}, k$
$M(1 - y_{ijk}) + (t_{jk} - t_{ik}) \geq P_{ik},$	$\forall i \in \{1,.., I-1\}, j \in \{i+1,.., I\}, k$
$y_{ijk} \in \{0,1\},$	$\forall i, j, k$

- Objective Function: Min Σ f_i ∀ i, or Min Σ max(f_i) ∀ i

This is the objective function, which minimizes the total tardiness of all jobs, or minimizing the maximum tardiness

- Variables and Constraints:

t_{ik}: The start time of job i on machine k.

P_{ik}: The processing time of job i on machine k.

D_i: The due date for job i.

f_i: The tardiness of job i. Tardiness is calculated as the amount of time by which the completion of a job exceeds its due date. It's only counted for the last step in the process.

y_{ijk}: A binary variable that is 1 if job i precedes job j on machine k, and 0 otherwise.

M: A large number, used in big-M method constraints to enforce logical conditions.

- Constraints:

$t_{iK} + P_{iK} - D_i \leq f_i, \ \forall i$: This ensures that the tardiness of the last job K is at least the amount by which the completion time (start time + processing time) exceeds its due date.

$t_{ik} \geq 0, f_i \geq 0, \ \forall i, k$: The start times and tardiness must be non-negative.

$t_{ik} + P_{ik} \leq t_{i(k+1)}, \ \forall i, k \in \{1,..., K - 1\}$: Each job must be completed on machine k before it can start on machine k+1.

$M * y_{ijk} + (t_{ik} - t_{jk}) \geq P_{jk}, \ \forall i \in \{1,..., I-1\}, j \in \{i+1,..., I\}, k$: If job i does not precede job j on machine k ($y_{ijk} = 0$), then job j must be completed (including its processing time) before job i starts on the same machine.

$M(1 - y_{ijk}) + (t_{jk} - t_{ik}) \geq P_{ik}, \ \forall i \in \{1,..., I-1\}, j \in \{i+1,..., I\}, k$: If job j does not precede job i on machine k ($y_{ijk} = 1$), then job i must be completed (including its processing time) before job j starts on the same machine.

$y_{ijk} \in \{0,1\}, \ \forall i, j, k$: The binary constraint on y_{ijk} ensuring it can only take the values 0 or 1.

- Specific constraints from the scheduling FIFO and EDD

FIFO (First In, First Out) and EDD (Earliest Due Date) are prioritization methods and not objective functions themselves. As a result, we can only add them as additional constraints.

FIFO constraint: $t_{(i+1)k} >= t_{ik} \ \forall i \in \{1,..., I -1\} \ \ \forall k \in \{1,..., K\}$

EDD constraint: Sorting the jobs based on due dates, and then use FIFO constraint on the sorted data to ensure the jobs are started based on the order of the sorted due dates

3. EXPERIMENT

a. The used values in this modeling assignment are reported below

There are in total 10 orders, which is the order in FIFO priority. The processing time P_ik and due date D_i are reported for all 10 jobs in the table below

	Processing times/step				Due date
Order#	Machine 1	Machine 2	Machine 3	Machine 4	DD
1	13	6	10	12	44
2	18	7	4	6	46
3	12	8	13	23	73
4	11	11	12	9	61
5	8	11	6	7	55
6	4	15	9	7	54
7	15	12	9	12	74
8	13	18	13	8	115
9	4	14	8	8	70
10	25	5	10	6	108

How many decision variables do we have from the CPLEX model?

```
dvar float+ t[1..I][1..K];          // Starting times of jobs on machines
dvar boolean y[1..I][1..I][1..K];   // Takes value 1, if job i is before job j on machine k
dvar float+ f[1..I];                // Tardiness of job i
```

I = 10, K = 4

=> Number of decision variables: 10 x 4 + 10 x 10 x 4 + 10 = 450

b. Model testing

Follow this tutorial video on how to navigate the OPL project.

https://www.youtube.com/watch?v=PcMj5nY_D2k&t=195s

These are the documentations on the website

Understanding OPL project:
https://www.ibm.com/docs/en/icos/22.1.1?topic=studio-understanding-opl-projects

CPLEX Studio handles OPL project files, data files, model files, setting files, and run configurations.

From the documentation website:

1. CPLEX Studio uses the concept of a project to associate an OPL model file with one or more data files and one or more settings files. The project file in the root folder organizes all the related model, data and settings files.

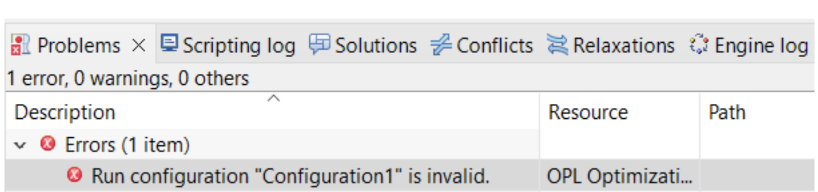
2. OPL model files: Model (.mod) files contain all of our OPL statements. The data and the objective function are not mandatory and there may be more optional components, such as scripting statements, in a model file.

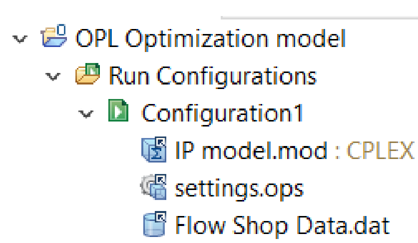
3. Data files: By separating the model of the problem from the instance data, we can organize large problems better. Each set of data is stored in a separate data file.

4. Settings files: If you change the default values of settings, the new user-defined values are stored in the settings file associated with a project.

5. Run configurations are a way of handling model, data, and settings files within a project.

If we encounter this error when we try to run the configuration, it is because the configuration hasn't had the model files yet. We can simply drag the model file to the configuration. We can also add the .ops settings file and .dat file, if they are accompanied with the model file





c. Modeling results

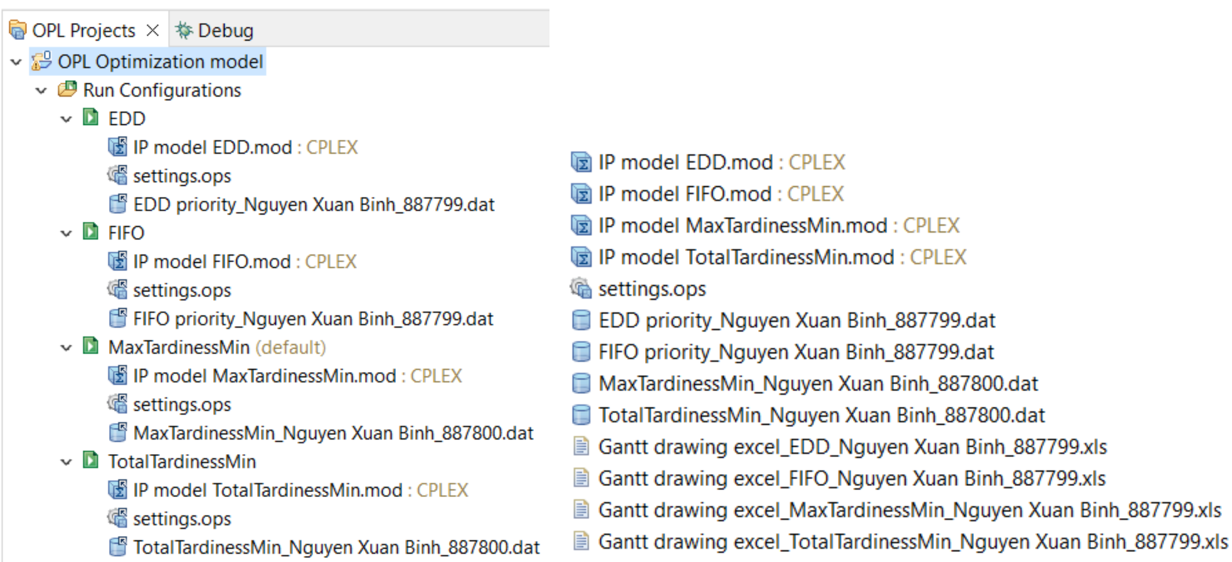
1. Experiment with 10 given orders (”FlowShopAssignmentData.xls”) and the 4-machine flow shop optimization model that you have to modify. Compare four scheduling methods:

- FIFO priority at all machines. Take job order in the data file as arrival order
- EDD (Earliest Due Date) priority at all machines
- Total tardiness minimization
- Maximum tardiness minimization

The first two are permutation schedules and can be solved by first sorting the orders in Excel and then forcing the schedule to obey that order in optimization.

Compare the results according to the following criteria: total tardiness, maximum tardiness, average throughput time (For each: finish of last - start time of first step), utilization of machines (For one: Total processing time / (finish time of last job - start time of first job)

After setting up everything, the CPLEX working directory should be looking like this



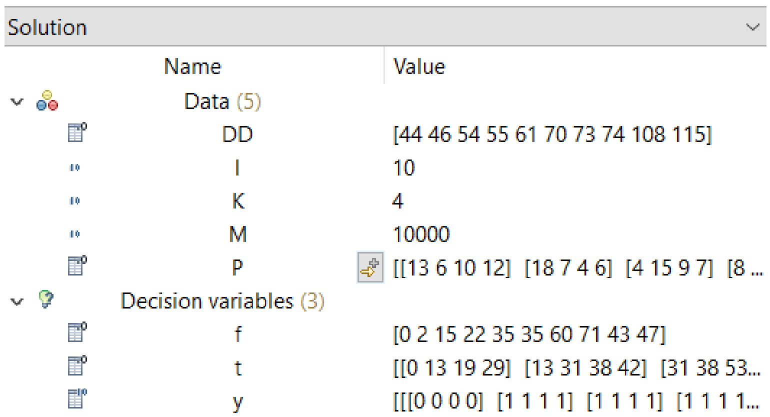
For FIFO and EDD priority scheduling, there is no objective function at all. The problem only contains the constraints, turning the MILP optimization problem into a feasibility problem.

We can use the dummy objective function as “minimize 1;”

However, we need to add the constraints to simulate FIFO/EDD into the .mod file
forall (i in 1..I - 1, k in 1..K)t[i+1][k] >= t[i][k];

The .dat file of FIFO should contain the original data given in the excel file. On the other hand, the .dat file of EDD should contain the jobs ordered by due dates, and the FIFO constraint will force the jobs to follow EDD based on the sorted order of the jobs.

After we run optimization, we would get output values for f (tardiness) and t (starting time).



The results are written to Gantt chart excel visualization files, where the command of writing and reading data from excel files in CPLEX in .dat file are:

// Data reading and writing to Excel worksheet

SheetConnection sheet("Gantt drawing excel_EDD_Nguyen Xuan Binh_887799.xls");

// We can also read data from excel file like this

// P from SheetRead(sheet, "A1:A5");

// DD from SheetRead(sheet, "B1:B5");

// We write data from CPLEX into excel file

t to SheetWrite(sheet, "W21:Z30");

f to SheetWrite(sheet, "AB21:AB30");

All statistics of total tardiness, maximum tardiness, average throughput time and utilization of machines are calculated inside the Gantt chart visualization excel file for the four scheduling approaches, FIFO, EDD, total tardiness minimization and max tardiness minimization. The statistics are reported in the table below

	FIFO	EDD	Total tardiness minimization	Maximum tardiness minimization
Total tardiness	364	330	204	418
Maximum tardiness	81	71	61	44
Average throughput time	51.9	52.8	45.7	48.6

Utilization rate	FIFO	EDD	Total tardiness minimization	Maximum tardiness minimization
Machine 1	100%	100%	100%	93.89%
Machine 2	84.25%	83.59%	86.29%	81.06%
Machine 3	70.15%	69.63%	78.99%	73.44%
Machine 4	75.38%	73.68%	86.73%	77.78%

Naturally, total tardiness minimization should have smallest total tardiness and max tardiness minimization should have smallest max tardiness. This is true as observed from the tables.

The next question we can verify from these two tables is, what schedule has the best result overall. We can see that all statistics of total tardiness minimization strictly dominates the other 3 schedules in all criteria, suggesting that it is the most efficient schedule.

For FIFO, EDD, and maximum tardiness minimization, none of them strictly dominates one another. However, based on utilization rate, the ranking could be max tardiness minimization > FIFO > EDD. For maximum tardiness and average throughput time, max tardiness minimization has the lowest one (naturally). Therefore, to rank the four schedules as a whole, we have this:

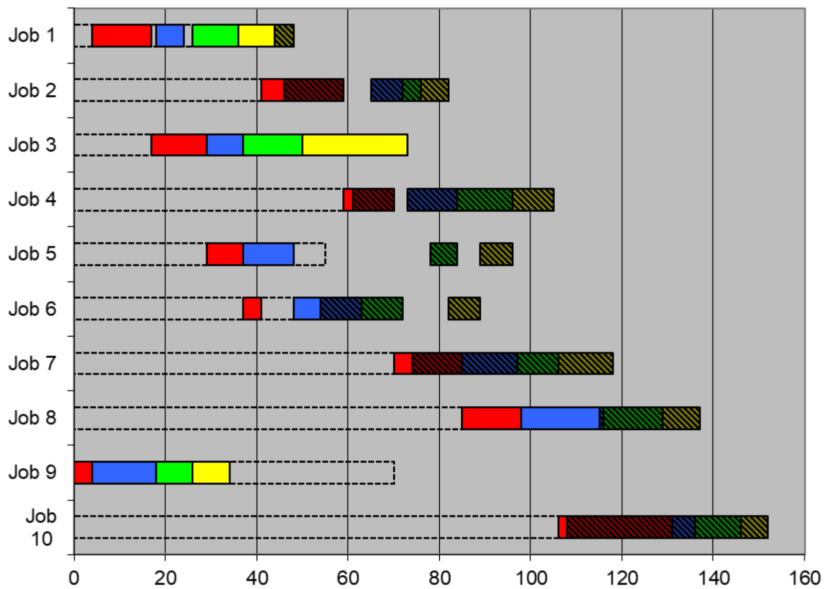
total tardiness minimization > max tardiness minimization > (FIFO, EDD) (equally bad)

2. Flow shop assignment

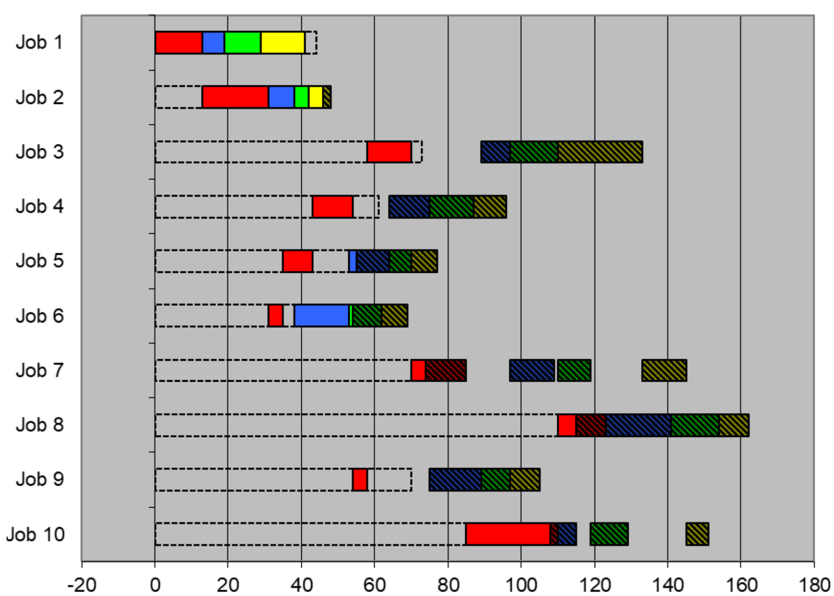
Use "Gantt-drawing-Excel.xls" to draw Gantt charts of the schedules. Automate data writing to Excel by inserting the commands in your OPL data file. Examining and verifying the schedules is much easier graphically. If optimization does not seem to converge, adjust the time limit in the .ops file. If you stop the run manually, the results are not written to the Excel file.

The Gantt charts of the four schedules are reported below

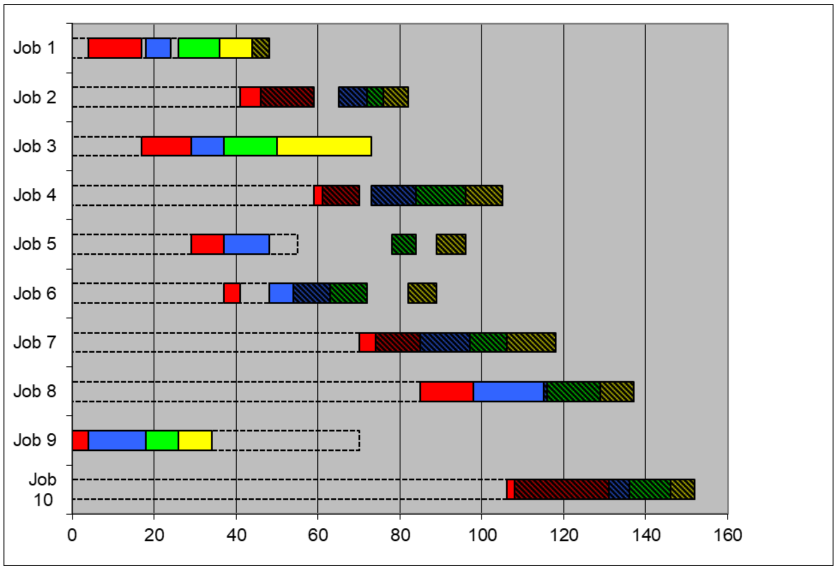
FIFO result



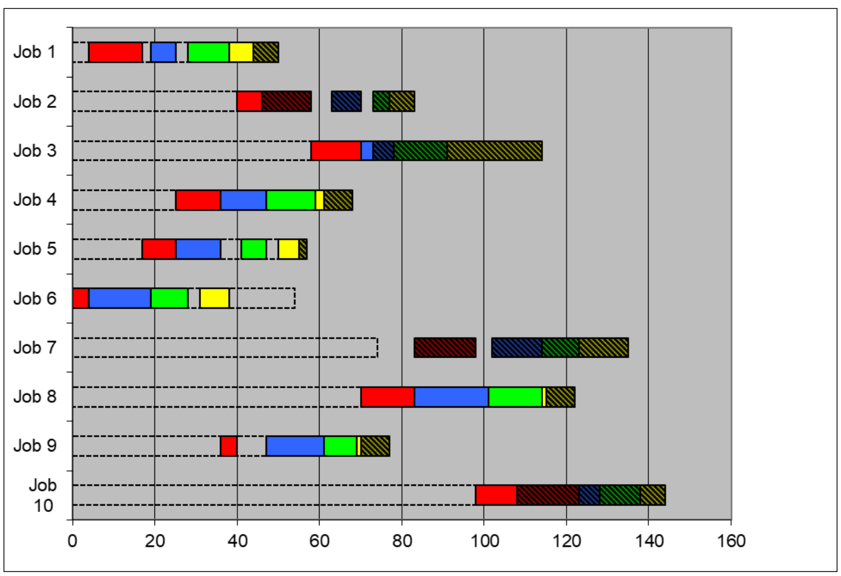
EDD result



Maximum tardiness minimization result



Total tardiness minimization result



We can see that for almost all jobs, they are always late over the deadline, suggesting that the deadlines given are quite unrealistic and very tight. One interesting thing is that total tardiness minimization has 9 overdue jobs, compared to only 8 overdue jobs in FIFO and maximum tardiness minimization schedules. However, it nonetheless has the smallest total tardiness, suggesting that the total tardiness is not correlated with the number of overdue jobs.

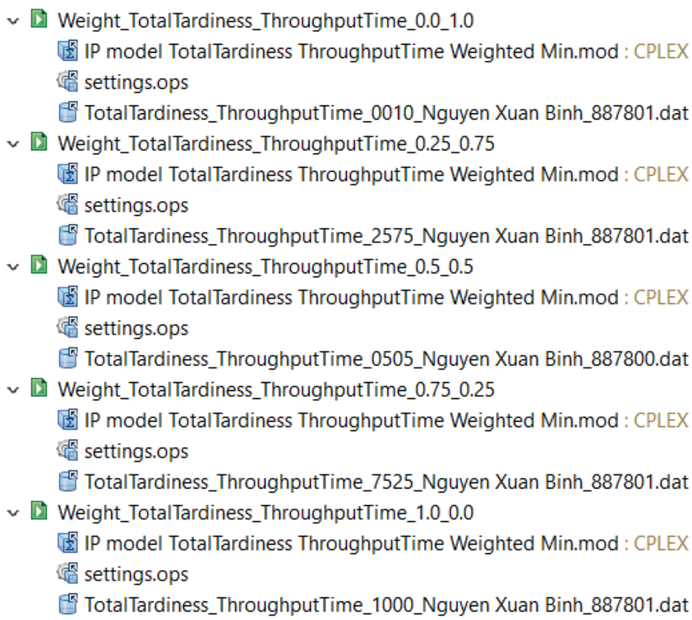
3. Extra problem

Modify your model to a multi-objective one so that throughput time and total tardiness are combined in the objective with a weight that you can adjust. Examine how tardiness changes when throughput time is taken into account with different weighing and vice versa.

In this extra problem, I try 5 different weights as follows

total tardiness - throughput time

0.0 - 1.0, 0.25 - 0.75, 0.5 - 0.5, 0.75 - 0.25, 1.0 - 0.0



Now, we need to define the weights in the .mod file like this

```
float Wtard = ...; // Weight for total tardiness
float Wtp = ...; // Weight for throughput time
```

and define them accordingly in their respective .dat file.

In the model file, we need to modify the objective function into the weighted multi objective, and also add a constraint for throughput time like the tardiness variable.

```
dvar float+ throughputTime[1..I];  // Throughput time for all jobs
```

```
// Objective: Minimization of sum of tardinesses
minimize Wtard * sum (i in 1..I)f[i] + Wtp * sum (i in 1..I) throughputTime[i];
```

```
// Constraint

subject to{

    ….

  forall (i in 1..I) throughputTime[i] == t[i][K] + P[i][K] - t[i][1];

}
```

The reported total tardiness and throughput time as reported by the CPLEX solver are:

Weight (total tardiness - throughput time)	Total tardiness	Total throughput time
0.0 - 1.0	442	422
0.25 - 0.75	234	425
0.5 - 0.5	215	433
0.75 - 0.25	204	454
1.0 - 0.0	204	454

We can observe that total tardiness and throughput time has an inverse relationship. If total tardiness is more emphasized, then the total throughput time would increase and vice versa. We can also see that the total tardiness would decrease very fast if it is given very minor weight. On the other hand, increasing weights for total throughput time would not significantly decrease it. Therefore, tardiness is much more sensitive to optimization than total throughput time.

4. ANALYSIS

Analysis of Scheduling Methods:

This reports compares and analyzes the performance of the four scheduling methods, FIFO, EDD, total tardiness minimization (TTM), and maximum tardiness minimization (MTM) based on 4 criteria, maximum tardiness, total tardiness, average throughput time, and machine utilization rate. TTM appears to be the best schedule, surpassing the others in total tardiness, average throughput time, and machine utilization rates. This is because TTM as a whole improves the total system performance rather than focusing on individual job timings. Despite having one more delayed job than FIFO and MTM, TTM effectively minimizes delay across all jobs rather than simply lowering the number of tardy jobs. FIFO and EDD, without any optimization objective, show much less utilization rate of the machines, which shows us the significance of informed scheduling beyond basic job permutation processing.

Furthermore, MTM, while ensuring the smallest delay for the jobs, does not provide a thorough reduction in tardiness, resulting in a performance that is superior to FIFO and EDD but inferior to total tardiness minimization. This is because it does not give a complete decrease in tardiness but rather only on the maximum tardiness. For example, 3 jobs with tardiness of 44 would have lower max tardiness than 3 jobs with tardiness of 1,1, 60, even though the latter case is a much better option in reality.

Impact of Multi-Objective Optimization:

Multi-objective optimization involves balancing overall tardiness and throughput time with different weights, which results in a clear trade-off. As the emphasis switches to throughput time, total tardiness rises, and vice versa. This inverse connection emphasizes the difficult balance of completing work quickly (throughput time) and meeting deadlines (tardiness). Additionally, the model is more responsive to changes in the weights for tardiness. A little emphasis on tardiness reduction results in a significant reduction in total tardiness, but increasing the significance of throughput time does not provide a proportionate improvement. This shows that in cases when deadlines are tight and penalties for delays are large, prioritizing tardiness in the objective function may result in greater performance benefits than focusing on throughput time. The sensitivity of tardiness to weight modifications might be used when the cost of delays surpasses the advantages of quick processing, directing planners to choose scheduling approaches that reduce tardiness above those that maximize throughput time.

5. CONCLUSION

a. General and specific conclusions

EDD is optimal for minimizing the maximum tardiness on a single machine, while FIFO is optimal when the objective is to minimize flow time on a single machine where jobs arrive at different times. However, these two schedules simply fail when it comes to parallel machines in a flow shop. However, general findings show that no single scheduling technique beats others on all measures, even though this report has a special case where TTM is the optimal one. To account for different measures, the simple weighted multi-objective optimization model provided a balanced prioritization of multiple objectives based on the flow shop's demands.

b. Practical value of this assignment

This assignment has practical value by evaluating various scheduling techniques and their influence on flow shop performance. By using the multi-objective optimization, planners may fine-tune scheduling to balance throughput time and tardiness, resulting in increased efficiency and fewer bottlenecks. This is especially useful in industrial environments, where meeting deadlines and increasing equipment use are important to making profits.

c. Reliability of results

The use of MILP model, as well as CPLEX solver, ensures that the findings are reliable. The study is based on known mathematical models, ensuring reliable results. However, the model's assumptions may oversimplify the real-world operations and thereby restrict the application of the obtained optimal solution found by the solver.