

Flexible job shop scheduling problem with sequence-dependent transportation constraints and setup times

Sacha Varone, David Schindl, Corentin Beffa
University of Applied Sciences Western Switzerland (HES-SO),
HEG Genève, Switzerland
Rue de la Tambourine 17, 1227 Carouge, Switzerland
Email: {sacha.varone, david.schindl}@hesge.ch

Abstract—We study a production scheduling problem, which addresses on the one hand the usual operational constraints such as the precedence of operations, time windows, delays, uniqueness of treatment, availability of resources, and waiting times. On the other hand, the problem takes into account possible restricted movements according to production orders. This problem is a variant of a flexible job shop scheduling problem with several types of sequence-dependent constraints. We consider additional sequence-dependent setup times, as well as sequence-dependent transportation and assignment restrictions. We propose a mixed integer programming model (MIP). It is based on the MIP model of a flexible job shop scheduling problem, in which we add those sequence-dependent constraints. We solve it with a general purpose MIP solver.

I. INTRODUCTION

THE metallic pieces production needs several steps in different machines, as warming the metal, soak the pieces and drain them. For a company needing to produce batches of different pieces, the optimal production scheduling is difficult to create, as different pieces need different machine's temperatures. The production line may have different equivalent machines. The choice of one of them may have an impact on the production, as the use of a machine may hinder the displacement between several machines. This problem may be seen as a variation of the well studied job shop scheduling problem (JSP) with additional constraints.

The usual job shop scheduling problem consists in scheduling a set of jobs on machines in order to minimize the makespan, defined as the completion time of the last job. These jobs are composed of different operations to be realized in a determined order, with some known processing times. Only one operation may be realized on a machine at the same time and no preemption is allowed. In the classical Job Shop Scheduling problem (JSP), the operations have to be scheduled on predefined machines, whereas in the Flexible Job Shop Scheduling problem (FJSP), each operation can be carried out on one machine from a set of compatible machines.

A review of the problem with different methods to obtain exact or approximated solutions is presented by Zhang et al. [1], with an analysis of the problem for the requirements of current industries. The result of a comparison of four different

models for the JSP problem, realized by Ku and Beck [2], highlights the performances of the disjunctive model. Contrary to the majority of the research, Karimi et al. [3] chose to take into account the transportation time between the machines in the purpose to be closer to the real case. Bentaleb et al. [4] propose a model taking into account the non-availability of the machines that can be caused by maintenance for example. In addition to take into account the periodic maintenance of the machines, Krim et al. [5] model the setup time for a one machine problem. The electrical consumption reduction can save significant costs to companies and reduce the impact on the environment. Therefore, Mansouri et al. [6] propose a multi objective optimization problem, while Assia et al. [7] propose a bi-objective function, to find the trade-off between minimizing the makespan and the total energy consumption.

This article presents an usual mixed integer linear programming model to solve the job shop scheduling problem, with different extensions to handle multi machines re-entrant jobs, time-windows, setup times, hindering movement and dedicated waiting machines. The problem is described in Section II, and its mathematical model in Section III. The model's evaluation is illustrated in Section IV, followed by a conclusion in Section V.

II. PROBLEM DESCRIPTION

Our study considers the flexible job shop scheduling problem with sequence-dependent constraints. It is stated as follows, in which we use a similar notation and vocabulary as in [8]. A set J of n jobs has to be scheduled on a set M of m machines. A job j is composed of a sequence of n_j consecutive operations, noted O_{j1}, \dots, O_{jn_j} . The l^{th} operation of job j , noted O_{jl} , can be processed on any of the compatible machines from the set $M_{jl} \subset M$. We denote $p_{O_{jl}}$ the processing time of operation O_{jl} , which is supposed to be independent of the machine on which the operation is carried out and no preemption is allowed. A sequence-dependent setup time is incurred between any two consecutive operations carried out on a same machine..

Each machine can perform at most one operation at a time. We suppose that each machine and each job are available

at time 0. The problem is to assign each operation O_{jl} to an eligible machine $k \in M_{jl}$ starting at a time t_{jl} . The objective is to minimize the sum of the starting times of the last operation of each job. This particular objective function limits the waiting time of jobs. This objective function is stricter than the minimization of the makespan C_{\max} , which is the time necessary to complete all jobs.

Our real case study is based on the FJSP, to which additional constraints have to be fulfilled. The machines are only available during determined periods of time and between two operations on a same machine, a setup-time is allowed to recondition the machine. Sometimes, the operations may not be processed right after the end of the previous one, so waiting machines are used with waiting operations without determined duration. A waiting time is not systematically allowed. Finally, regarding the topology of the machines, certain displacements are not allowed according to the use of certain machines. An example of topology for metal piece production is presented on Figure 1 with two ovens (O_1 and O_2) and where a drain operation (H'_1 or H'_2) is executed over the soak machine (H_1 or H_2). These drain operations may hinder moving a job between two machines as illustrated on Figure 2. Such operations are called blocking operations.

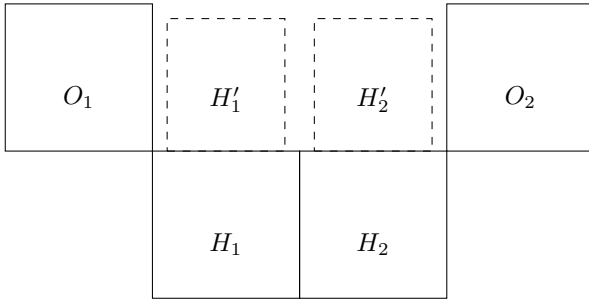


Fig. 1. Example of a machine line topology with two ovens (O_1 and O_2), two soak machines (H_1 and H_2) and two drain machines (H'_1 and H'_2).

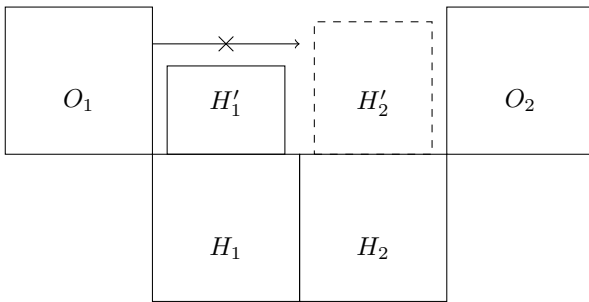


Fig. 2. Example of forbidden movement between the machines O_1 and H_2 , induced by the draining operation on the machine H'_1

The notation is presented on Table I. The constant $bigM$ is defined as follows:

$$bigM = \sum_{j \in J} \sum_{l=1}^{|O_j|} p_{O_{j,l}}$$

This is representing the time needed if the recipe of all the jobs are composed of all operations and only one job can be done at a time.

III. MODEL

A. Variables

a) *Usual variables*: The proposed model works with a continuous representation of the time. Three principal sets of variables are needed. They express the beginning time of each operation of the recipe of a job on a machine, the assignment of a machine to a job's operation and the precedence between two jobs' operations.

$x_{j,O_{j,l}} \in \mathbb{R}$: start time of the l^{th} operation of job j

$$y_{j,O_{j,l},m} = \begin{cases} 1 & \text{if the } l^{th} \text{ operation of job } j \text{ is carried out on} \\ & \text{machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$z_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} = \begin{cases} 1 & \text{if the couple } (O_{j_1,l_1}, j_1) \text{ occurs before} \\ & \text{the couple } (O_{j_2,l_2}, j_2) \\ 0 & \text{otherwise} \end{cases}$$

As a machine may be visited several times during the recipe execution, the predecessor variables take into account the operation in addition to the job.

b) *Waiting machines*: The proposed model uses waiting shelves, permitting to liberate the machines in the case of waiting time. The idea is to recreate the x , y and z variables dedicated to the waiting operations. To each operation, we associate a waiting operation "prime" with no determined duration. This duration is the difference between the end of the corresponding job's previous operation and the beginning of the next one. Currently, all waiting machines are equivalent and can therefore be used independently of the last job's operation.

$$x'_{j,O_{j,l}} \geq 0, \forall j \in J, l \in \{1, \dots, |O_j|\}$$

$$y'_{j,O_{j,l},m} \in \{0, 1\}, \forall j \in J, l \in \{1, \dots, |O_j|\}, m \in StationWaiting$$

$$z'_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} \in \{0, 1\}, \forall j_1 \in J, l_1 \in \{1, \dots, |O_{j_1}|\}, j_2 \in J, l_2 \in \{1, \dots, |O_{j_2}|\}$$

c) *Time windows*: Machines may be unavailable for different reasons. To take into account these unavailabilities, boolean variables are needed to express which time window is used for each operation's execution.

$$tw_{j,O_{j,l},k} \in \{0, 1\}$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}, k \in TW \text{ such that } Comp_{m_k,O_{j,l}} = 1$$

B. Constraints

- 1) *Precedence*: to ensure the coherence of the precedence variable, only one of two jobs may be executed before the other:

$$z_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} + z_{j_2,O_{j_2,l_2},j_1,O_{j_1,l_1}} \leq 1 \\ \forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, \forall l_1 \in \{1, \dots, |O_{j_1}|\}$$

TABLE I
TABLE OF NOTATIONS

Notations:	Signification:
$j \in J$:	indices of jobs
$m \in M$:	indices of machines
$w \in TW$:	indices of the time windows
d_j :	deadline of job j
O_{jl} :	l^{th} operation for job j
O'_{jl} :	l^{th} waiting operation for job j
$Tb_{O_{jl}}$:	temperature at the beginning of operation O_{jl}
$Te_{O_{jl}}$:	temperature at the end of operation O_{jl}
po_{jl} :	processing time of operation O_{jl}
$Comp_{m,O_{jl}}$:	set of compatibility with value 1 if operation O_{jl} may be executed on machine m and 0 otherwise.
$prept_{t_1,t_2,m}$:	preparation time (or setup time) to go from temperature t_1 to temperature t_2 on machine m .
$bigM$:	constant representing a big number
$hinder_{m_i,O_{j_1}m_jm_k}$:	inconvenience caused by the use of machine m_i for operation O_{j_1} on the motion between machines m_j and m_k . A value of 1 is given if such a case occurs and 0 otherwise.
$nodelay_{O_{j_1}}$	set of boolean values indicating if a delay is permitted between l^{th} operation and the next one
StationWaiting	set of index of the waiting machines
CycleWaiting	set of indexes of the waiting operations. To each operation O_{j_l} correspond a waiting operation O'_{j_l}
mc_w	machine corresponding to time window w
$earliest_w$	beginning of time window w
$latest_w$	end of time window w

$$O_{j_1} \}, l_2 \in \{1, \dots, |O_{j_2}|\}$$

The same idea has to be applied between two operations of a same job:

$$z_{j,O_{j_1,l_1},j,O_{j_2,l_2}} + z_{j,O_{j_2,l_2},j,O_{j_1,l_1}} \leq 1$$

$$\forall j \in J, \forall (l_1, l_2) \in \{(l_1, l_2) \in \{1, \dots, |O_j|\}^2 \mid l_1 \neq l_2\}$$

2) Precedence uniqueness on the waiting machines:

$$z'_{j_1,O_{j_1,l_1},j_2,O'_{j_2,l_2}} + z'_{j_2,O_{j_2,l_2},j_1,O'_{j_1,l_1}} \leq 1$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, l_1 \in \{1, \dots, |O_{j_1}|\}, l_2 \in \{1, \dots, |O_{j_2}|\}$$

3) Operation order: the operations order in the recipe has to be respected. So the beginning time of an operation has to be larger than the ending time of the previous operation of the same job.

$$x_{j,O_{j,l+1}} \geq x_{j,O_{j,l}} + po_{O_{j,l}}$$

$$\forall j \in J, l \in \{1, \dots, |O_j| - 1\}$$

If two operations are realized on the same machine, the preparation time of the second one has to be taken into account. So the starting time of that operation has to be larger than the ending time of the previous one plus the preparation time. If they are not realized on the same machine, the constraint is relaxed due to the "M" term:

$$x_{j,O_{j,l+1}} \geq x_{j,O_{j,l}} + po_{O_{j,l}} + prept_{Te_{O_{j,l}},Tb_{O_{j,l+1}},m} - bigM(2 - y_{j,O_{j,l},m} - y_{j,O_{j,l+1},m})$$

$$\forall j \in J, l \in \{1, \dots, |O_j| - 1\}, m \in \{m \in M \mid Comp_{m,O_{j,l}} = 1, comp_{m,O_{j,l+1}} = 1\}$$

4) No conflict on the same machine: two operations O_{j_1,l_1} and O_{j_2,l_2} may not overlap on the same machine, so O_{j_1,l_1} has either to finish before the beginning of O_{j_2,l_2} , or to begin after the end of O_{j_2,l_2} .

$$x_{j_1,O_{j_1,l_1}} \geq x_{j_2,O_{j_2,l_2}} + po_{O_{j_2,l_2}} +$$

$$prept_{Te_{O_{j_2,l_2}},Tb_{O_{j_1,l_1}},m} - bigM * (pred_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} + \sum_{m' \neq m} (y_{j_1,O_{j_1,l_1},m'} + y_{j_2,O_{j_2,l_2},m'}))$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, \forall l_1 \in \{1, \dots, |O_{j_1}|\}, l_2 \in \{1, \dots, |O_{j_2}|\}, \forall m \in M$$

and:

$$x_{j_2,O_{j_2,l_2}} \geq x_{j_1,O_{j_1,l_1}} + po_{O_{j_1,l_1}} + prept_{Te_{O_{j_1,l_1}},Tb_{O_{j_2,l_2}},m} - bigM * ((1 - z_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}}) + \sum_{m' \neq m} (z_{j_1,O_{j_1,l_1},m'} + y_{j_2,O_{j_2,l_2},m'}))$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, \forall l_1 \in \{1, \dots, |O_{j_1}|\}, l_2 \in \{1, \dots, |O_{j_2}|\}, \forall m \in M$$

5) No conflict on a waiting machine: the use of a waiting machine may happen only after or before the use of the same machine by another job.

$$x'_{j_1,O_{j_1,l_1}} \geq x_{j_2,O_{j_2,l_2+1}} - bigM * (z'_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} + (2 - y'_{j_1,O_{j_1,l_1},m} - y'_{j_2,O_{j_2,l_2},m}))$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, l_1 \in \{1, \dots, |O_{j_1}|\}, l_2 \in \{1, \dots, |O_{j_2}|\} - 1\}$$

and:

$$x'_{j_1,O_{j_1,l_1}} \leq x'_{j_2,O_{j_2,l_2}} + bigM((1 - z'_{j_1,O_{j_1,l_1},j_2,O_{j_2,l_2}} + (2 - y'_{j_1,O_{j_1,l_1},m} - y'_{j_2,O_{j_2,l_2},m}))$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 \mid j_1 \neq j_2\}, l_1 \in \{1, \dots, |O_{j_1}|\}, l_2 \in \{1, \dots, |O_{j_2}|\}$$

6) Respect of compatibilities: an operation may be executed on a machine only if they are compatible.

$$y_{j,O_{j,l},m} \leq Comp_{m,O_{j,l}}$$

$$\forall j \in J, l \in \{1, \dots, |O_{j,l}|\}, m \in M$$

7) Each operation of a job has to be realised on exactly one compatible machine:

$$\sum_{m \in \{m \in M \mid Comp_{m,O_{j,l}} = 1\}} y_{j,O_{j,l},m} = 1$$

$$\forall l \in \{1, \dots, |O_j|\}, j \in J, m \in M$$

None of the other machines are used for the operation, so all other $y_{j,O_{j,l},m}$ variables are set to 0:

$$\sum_{m \notin \{m \in M | Comp_{m,O_{j,l}}=1\}} y_{j,O_{j,l},m} = 0$$

$$\forall l \in \{1, \dots, |O_j|\}, j \in J, m \in M$$

8) At most one waiting machine per operation and job:

$$\sum_{m \in machineWaiting} y'_{j,O_{j,l},m} \leq 1$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}$$

9) No delay: if the nodelay value is set to true, the beginning of the following operation has to be the same as the ending time of the previous operation.

$$\sum_{(j,O_{j,l}) \in NDNL} (x_{j,O_{j,l+1}} - (x_{j,O_{j,l}} + p_{O_{j,l}})) = 0$$

Where NDNL represents the set of all the couples $(J \times O)$ preceding the operation where the nodelay value is set to true in the associated recipe.

10) Required succession of machines: sometimes, the use of a certain machine for an operation implies the use of the same machine for the next operation.

$$y_{j,O_{j,l},m} = y_{j,O_{j,l+1},m}$$

$$\forall j \in J, l \in \{l \in \{1, \dots, |O_j|\} | nodelay_{O_{j,l}}=1\}, m \in M$$

11) Impossible machine successions: regarding the disposition of the machines, some of them may not be reached from another machine.

$$y_{j,O_{j_1},m_1} + y_{j,O_{j_2},m_2} = 0$$

$$\forall j \in J, \forall l \in \{1, \dots, |O_j|\} - 1, \forall (m_1, m_2) \in incompatibleSuccession$$

Where incompatibleSuccession is the set of all ordered pairs of incompatible machine successions.

12) Blocking operation: if a blocking operation is running, the concerned movement has to be realized either before the beginning of the blocking operation, or after the end of the blocking operation.

$$x_{j_1,O_{j_1},l_1} \geq (x_{j_2,O_{j_2},l_2} + p_{O_{j_2},l_2}) * hinder_{m_i,O_{j_2},l_2,m_j,m_k} - bigM * (3 - y_{j_2,O_{j_2},l_2,m_i} - y_{j_1,O_{j_1},l_1+1,m_j} - y_{j_1,O_{j_1},l_1,m_k}) - bigM * z_{j_1,O_{j_1},l_1;j_2,O_{j_2},l_2}$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 | j_1 \neq j_2\}, \forall l_1 \in \{1, \dots, |O_{j_1}|\} - 1, l_2 \in \{1, \dots, |O_{j_2}|\}, \forall m_i, m_j, m_k \in M$$

and:

$$x_{j_2,O_{j_2},l_2} \geq x_{j_1,O_{j_1},l_1} * hinder_{m_i,O_{j_2},l_2,m_j,m_k} - bigM * (3 - y_{j_2,O_{j_2},l_2,m_i} - y_{j_1,O_{j_1},l_1+1,m_j} - y_{j_1,O_{j_1},l_1,m_k}) - bigM * (1 - z_{j_1,O_{j_1},l_1;j_2,O_{j_2},l_2})$$

$$\forall (j_1, j_2) \in \{(j_1, j_2) \in J^2 | j_1 \neq j_2\}, \forall l_1 \in \{1, \dots, |O_{j_1}|\} - 1, l_2 \in \{1, \dots, |O_{j_2}|\}, \forall m_i, m_j, m_k \in M$$

13) Deadline: each job has to be completed before the deadline.

$$x_{j,O_{j,end}} + p_{O_{j,end}} \leq d_j$$

$$\forall j \in J$$

14) Relation between the x and x' variables:

$$x'_{j,O_{j,l}} = x_{j,O_{j,l}} + p_{O_{j,l}}$$

$$\forall x \in J, l \in \{1, \dots, |O_{j,l}|\}$$

15) If there is a waiting time between two operations, a waiting machine has to be used:

$$bigM * \sum_{m \in machineWaiting} y'_{j,O_{j,l},m} \geq x_{j,O_{j,l+1}} - (x_{j,O_{j,l}} + p_{O_{j,l}})$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}$$

and:

$$\sum_{m \in machineWaiting} y'_{j,O_{j,l},m} \leq bigM * (x_{j,O_{j,l+1}} - (x_{j,O_{j,l}} + p_{O_{j,l}}))$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}$$

16) Only one time window per couple operation job is allowed:

$$\sum_{k \in \{w \in TW | \exists m' \in mc_w, m' = m\}} tw_{j,O_{j,l},k} = y_{j,O_{j,l},m}$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}, m \in M$$

17) Each operation cannot start earlier than the beginning of the time window:

$$\sum_{k \in TW} tw_{j,O_{j,l},k} * earliest_k \leq x_{j,O_{j,l}}$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}$$

18) Each operation has to be completed before the end of the time window:

$$\sum_{k \in TW} tw_{j,O_{j,l},k} * latest_k \geq x_{j,O_{j,l}} + p_{O_{j,l}}$$

$$\forall j \in J, l \in \{1, \dots, |O_j|\}$$

C. Objective function:

Contrary to the usual job shop problem, where the objective is to minimize the makespan, the proposed objective to minimize is the sum of the starting times of the last operation of each job:

$$\min \sum_{j \in J} x_{j,O_{j,end}}$$

This is used in the purpose to minimize the production time of each job and limit the unnecessary waiting time of the jobs having no influence on the makespan.

IV. EVALUATION

The proposed model has been implemented in the Julia programming language [9], [10] using the Gurobi solver [11]. Gurobi is considered as one of the state of the art commercial solver [2].

A. Numerical experiments

The experiments were conducted on three types of data: data generated specifically according to the constraint tested, randomly generated, as well as real data. The real data made it possible to ensure that the model proposed fits the real problem. The data presented here is an example, among the tests performed on real data. Four jobs have to be realized on the machines. The topology of the machines is shown in Figure 3, with the possible operations on each machine. Furthermore, five waiting machines, not represented in Figure

TABLE II
RECIPE FOR EACH JOB

job id	recipe (operation id)
1	1, 4, 4', 13, 9, 10
2	2, 5, 5', 14, 9, 11
3	7, 3, 6, 6', 15, 9, 11
4	12, 13, 8

TABLE III
INCOMPATIBILITY

machine id	machine id
D_1	E_3
D_3	E_1
D_3	E_2

3, are available. The manipulator is used for all displacements between machines that are not provided by the moving oven themselves. In order to maintain a reasonable computing time, the utilization of the manipulator has been neglected. The arrows on the figure symbolize the moving machines. Table II presents the recipes, which are defined as successions of operations to be accomplished for a given job. Data about the time-windows and the preparation times are not given here, but a full data set may be provided on request.

According to the arrangement of the machines, some displacements are not possible, whatever operations are carried out on them. This sequence of machine utilization is stored in Table III. Sometimes, the accessibility of a machine from another one is hindered by a specific operation on a third machine. These cases are presented in Table IV.

B. Solution

The solution returned by the model is presented as a Gantt chart, shown in Figure 4. The resolution time comprising the creation of the model and the resolution of the problem is less than five minutes. No better solution can be found since all jobs begin at time 0 and there are no waiting times between operations.

V. CONCLUSION

This study is a first part of a real-world industrial project. This production scheduling problem has the goal to minimize

TABLE IV
HINDERS

machineHinders	operationHinders	machineOrigin	machineDestination
E_1	4'	D_1	E_2
E_1	5'	D_1	E_2
E_1	6'	D_1	E_2
E_2	4'	D_2	E_1
E_2	5'	D_2	E_1
E_2	6'	D_2	E_1

the production time of different jobs by optimizing the assignment of their operations on the machines, subject to several constraints. These constraints include limited availability of the machines, setup times for reconditioning of the machines, dynamic forbidden displacements and waiting machines. As a future work, manipulator constraints have to be integrated, and the objective function will also consider the monetary cost minimization, taking into account electricity tariffs. The use of heuristics will help in the design of a good real-time solution, able to adapt the solution to events occurring each minute, such as a breakdown of a machine or the arrival of new jobs.

REFERENCES

- [1] J. Zhang, G. Ding, Y. Zou, S. Qin, and J. Fu, "Review of job shop scheduling research and its new perspectives under Industry 4.0," *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1809–1830, Apr. 2019. doi: 10.1007/s10845-017-1350-2. [Online]. Available: <https://doi.org/10.1007/s10845-017-1350-2>
- [2] W.-Y. Ku and J. C. Beck, "Mixed Integer Programming models for job shop scheduling: A computational analysis," *Computers & Operations Research*, vol. 73, pp. 165–173, Sep. 2016. doi: 10.1016/j.cor.2016.04.006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0305054816300764>
- [3] S. Karimi, Z. Ardalan, B. Naderi, and M. Mohammadi, "Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm," *Applied Mathematical Modelling*, vol. 41, pp. 667–682, Jan. 2017. doi: 10.1016/j.apm.2016.09.022. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0307904X16304929>
- [4] M. Benttaleb, F. Hnaïen, and F. Yalaoui, "Two-machine job shop problem under availability constraints on one machine: Makespan minimization," *Computers & Industrial Engineering*, vol. 117, pp. 138–151, Mar. 2018. doi: 10.1016/j.cie.2018.01.028. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360835218300354>
- [5] H. Krim, R. Benmansour, and D. Duvivier, "On the single machine scheduling problem with sequence-dependent setup times and periodic maintenance," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, Apr. 2018. doi: 10.1109/CoDIT.2018.8394827 pp. 374–378.
- [6] S. A. Mansouri, E. Aktas, and U. Besikci, "Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption," *European Journal of Operational Research*, vol. 248, no. 3, pp. 772–788, Feb. 2016. doi: 10.1016/j.ejor.2015.08.064. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377221715008206>
- [7] S. Assia, E. A. Ikram, A. El Barkany, and E. Biyaali Ahmed, "Two-Machine Job Shop Scheduling Problem Under Availability Constraints on one machine: Total Energy Consumption (TEC) minimization," 2018.
- [8] L. Shen, S. Dauzère-Pérès, and J. S. Neufeld, "Solving the flexible job shop scheduling problem with sequence-dependent setup times," *European Journal of Operational Research*, vol. 265, no. 2, pp. 503–516, Mar. 2018. doi: 10.1016/j.ejor.2017.08.021. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S037722171730752X>
- [9] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017. doi: 10.1137/141000671. [Online]. Available: <http://julialang.org/publications/julia-fresh-approach-BEKS.pdf>
- [10] [Online]. Available: <https://julialang.org>
- [11] [Online]. Available: <http://www.gurobi.com/>

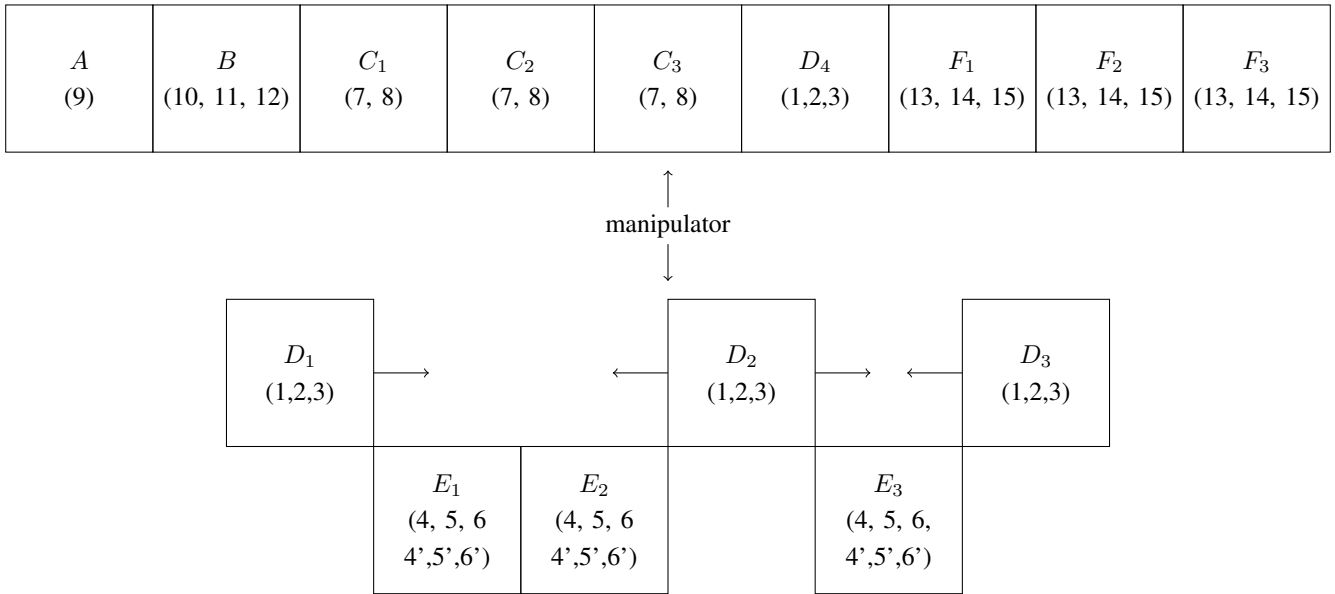


Fig. 3. Machine line topology with the name of each machine and in parenthesis the cycle that can be done on each machine (compatibility).

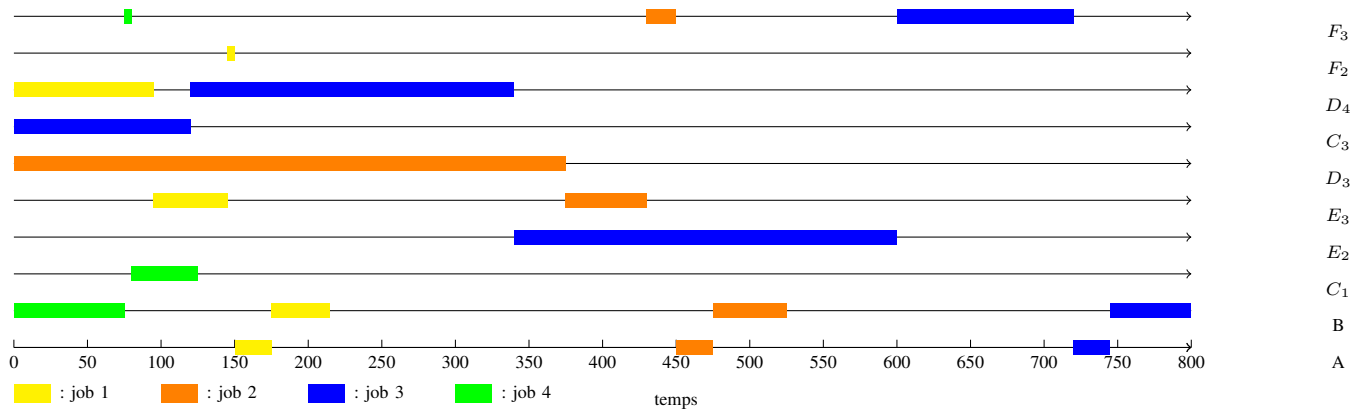


Fig. 4. time table representing the solution