A study on the group sequencing method in regards with transportation in an industrial FMS

Guillaume Pinot, Olivier Cardin and Nasser Mebarki

Abstract— **We are interested in jobs scheduling in flexible manufacturing systems. We try to combine predictive methods which guarantee optimal or good results but are not very robust to perturbations and reactive methods which take into account the real state of the system. For this, we use a method called the group sequencing method which aims at characterizing a set of solutions for the scheduling problem, instead of a single one. This method enables to add sequential flexibility to the operations to process. At the same time, this method guarantees a minimal quality in regards with the performance objectives. In this paper, we are interested in the behavior of this method in regards with both predictive and reactive methods, especially with various degrees of uncertainties. We present the experiments conducted on a real flexible manufacturing system to compare the performances of three scheduling methods in regards with uncertainties on the transportation times: the group sequencing approach, a predictive one and a reactive one. The results show very good performance with the group sequencing method.**

I. Introduction

A Flexible Manufacturing System (FMS) is an integrated system composed of automated workstations such as computer numerically controlled machines with tool changing capability, a material handling and storage system such as automated guided vehicles or conveyors, and a computer control system which controls the operations of the whole system [8]. It is designed to combine high productivity and production flexibility. But to achieve simultaneously these two goals, a FMS needs an adapted control. One of the most important and difficult problem to solve in order to control a FMS is to propose an efficient schedule for the production planning.

The scheduling problem can be considered as the allocation of a set of tasks to a set of resources under specified constraints. A job is a set of interrelated tasks. In manufacturing systems, an important type of constraint is precedence constraint, which partitions the scheduling problem in three major classes:

- the flow-shop problem where the processing sequence is the same for all the jobs;
- the job-shop problem where the processing sequence is fixed for each job but can be different for different jobs;
- the open-shop problem where the processing sequence is free.

In this paper, we are dealing with an industrial FMS which can be considered as a job-shop problem. Schedules can also be classified in three sets:

- semi-active schedules which are feasible schedules (*i.e.* schedules satisfying all the constraints) where no task can be completed earlier, in order to obtain a feasible schedule, without any permutation on a resource;
- active schedules which are feasible schedules where no task can be completed earlier, in order to obtain a feasible schedule, without delaying another task;
- non-delay schedules which are feasible schedules where no machine is kept idle while it could start a waiting task.

Scheduling problems [10] where the number of jobs and their properties (*e.g.* dates, sequences and times) are known at the beginning of the schedule are referred as static scheduling problems. In that case, the scheduling problem can be considered as an optimization problem, where the objective function is a set of performance criteria such as the makespan (i.e the duration required to complete the given set of jobs), the total tardiness, the maximum lateness, the total waiting time. Job-shop scheduling problems is an NP-hard optimization problems. To solve these optimization problems, predictive methods which give an optimal or good feasible schedule in regards with the objective function are used. These methods guarantee the quality of the schedule if this schedule is then strictly followed in the shop. Unfortunately, because of the frequent perturbations in flexible manufacturing systems, once an optimal schedule has been proposed by using a costly predictive method, it won't be long before this schedule has to be challenged. Lastly, these predictive methods are using hypotheses on the system or are relaxing some constraints of the problem. They do not take into account neither the model's uncertainties nor the real state of the system.

Control methods, such as dispatching rules [9], model the problem as a set of decisions to take dynamically for creating the schedule. These are reactive methods which try to take into account the real state of the system. But they do not guarantee the quality of the schedule.

In this paper, we use an original scheduling method which combines the advantages of both approaches: the predictive one and the reactive one. This method, called the group sequencing method, aims at characterizing a set of solutions for the scheduling problem, instead of a single one, by using groups of permutable operations [1]. This method also guarantees a minimal quality corresponding to the worst case. So, this method enables to add flexibility to the sequence scheduling and, as a matter of fact, it should be more robust to the perturbations.

In this paper, our aim is to study the behavior of this

G. Pinot, O. Cardin and N. Mebarki are with IRCCyN, 1 rue de la Noé, BP 92101, 44321 Nantes Cedex 3.
surname.name@irccyn.ec-nantes.fr

1-4244-0991-8/07/$25.00/©2007 IEEE

method in regards with both predictive and reactive methods. We conduct experiments on a real flexible manufacturing system to compare the performances of three scheduling methods: the group-scheduling approach, a predictive one, a reactive one, in regards with perturbations and uncertainties. Our real system has been modeled as a job-shop model with various precisions for transportation times (*i.e.* uncertainties for the model). Our main question is to discuss the compared performances of these three scheduling methods on a job-shop model of an industrial application, with various degrees of uncertainties.

Firstly, we will present the group sequencing method, then we will present the industrial case. The conducted experiments and the results obtained will finally be discussed.

II. Sequencing Groups of Permutable Operations

Group of permutable operations was first introduced in [5]. The goal of this method is to have a sequential flexibility during the execution of the schedule and to guarantee a minimal quality corresponding to the worst case. This method has been widely studied in the last twenty years, in particular in [5], [2], [11], [1]. For a theoretical description of the method, see [1].

A group of permutable operations is a set of operations that will be performed on a given resource in any order. A group sequence is defined by an ordered list of groups (of permutable operations) on each machine, performed in this order. A group sequence is feasible if every permutation between all the operations of the same group gives a satisfying schedule (*i.e.* a schedule which satisfies all the constraints of the problem). As a matter of fact, a group sequence describes a set of valid schedules, without enumerating them.

The quality of a group sequence is expressed the same way a classical schedule is. However, it is measured as the quality of the worst semi-active schedule hold in the group sequence, as defined in [1].

To illustrate these definitions, let us study an example. Fig. 1a presents a job shop problem with three machines and three jobs, while Fig. 1b presents a feasible group sequence solving this problem. It is made of seven groups: two groups of two operations and five groups of one operation. This group sequence describes four different semi-active schedules shown in Fig. 2. Note that these schedules do not always have the same makespan noted $C_{\max}$: The best case quality is $C_{\max} = 10$ and the worst case quality is $C_{\max} = 17$.

Group sequencing has an interesting property: the quality of a group sequence in the worst case can be computed in polynomial time for minmax regular objective function as the makespan and the maximum lateness (see [1] for the description of the algorithm). Thus, it is possible to compute the worst case quality for large scheduling problems. Consequently, this method may be used to compute the worst case quality in real time during the execution of the schedule. Due to this real-time property, it is possible to use it in a decision support system dynamically.

This method enables to describe a set of schedules in an implicit manner (*i.e.* without enumerating the schedules) which guarantees a minimal performance. Indeed, as it proposes a group of permutable operations, one can choose inside a group the most fitted operation to the real state of the system.

Furthermore, the flexibility added to the schedule should be able to absorb uncertainties. Only two studies have tried to verify this property. Wu, Byeon and Storer in [11] study the impact of disturbed processing times on the weighted tardiness sum objective in comparison with static and dynamic heuristics. When processing times are not so much disturbed, they observe that group sequencing obtain better performances. Esswein in [6] studies the impact of disturbed processing times, due dates and release dates on a one machine problem and compares its results with a static heuristic method. In average, performances are better with group sequencing than with the static method.

In this article, we propose to conduct a new study on the robustness of group sequencing with various degrees of precision for transportation times in an industrial scheduling problem.

III. Adaptation of the Method to an Industrial Flexible Manufacturing System

To conduct experiments on a large time horizon, we are using an emulation [3] of a Flexible Manufacturing System. The system (Fig. 3 for a picture of the real system) is modeled as a six machines job shop, with automated transfers (Fig. 4). Each machine has a finite capacity upstream queue. Four of them are only FIFO queues, the two others enabling a free choice. Automated transfers between machines are performed by 42 unidirectional transporters, stocked in a storehouse during the inactivity periods. They cannot handle more than one part at a time, but they also carry all data of the part updated in real-time (identification, routing, progress, *etc.*).

The structure of the transport system is built around a central loop. Each transporter moves on the loop until it can enter the queue of a machine. To be more specific, when a transporter arrives at the entrance of any machine, a decision, based on the stock level, on the routing of the part and on the schedule of the machine, is made to allow (or not) the transporter to enter the queue of the machine. If not, the transporter goes on to the next machine entrance.

The emulation model allows us to simulate several scenarios, characterized by two parameters:

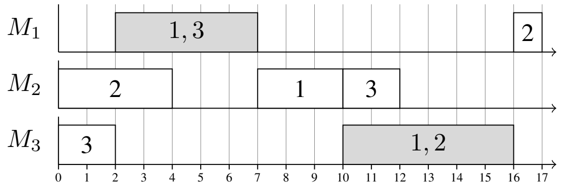
- the speed of the transporters (from zero to infinity);
- the decision rules to enter a queue.

Depending on this decision rule, we can define three different execution modes:

- A predictive schedule based on sequence of operations: to execute this schedule on the FMS, the sequence of operations of each machine must follow a predefined sequence given at the beginning of the execution. The entrance rule might be formulated as “the part enters the upstream queue of the machine if the queue is not full and its next operation corresponds to the next operation

i	j	$M_{i,j}$	$p_{i,j}$
1	1	1	3
1	2	2	3
1	3	3	3
2	1	2	4
2	2	3	3
2	3	1	1
3	1	3	2
3	2	1	2
3	3	2	2

(a) A job shop problem



(b) A group sequence solving the problem describe in Fig. 1a

Fig. 1: A Job Shop Problem Solved by a Group Sequence

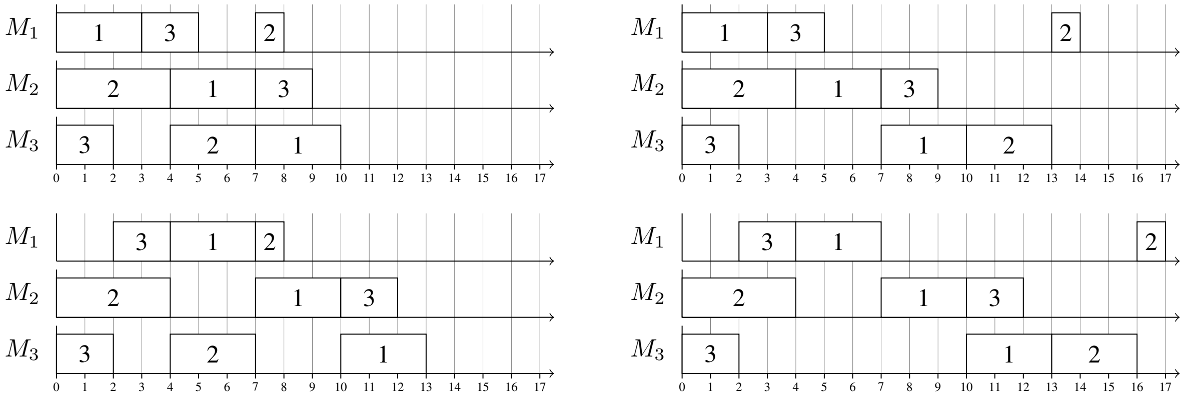


Fig. 2: Semi-active Schedules Described by Fig. 1b

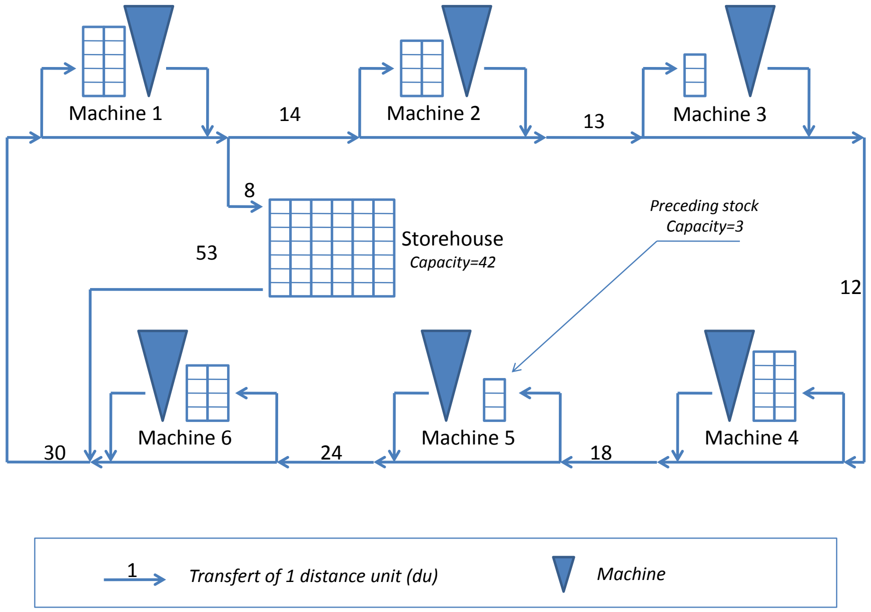


Fig. 4: Job Shop Model of the studied FMS

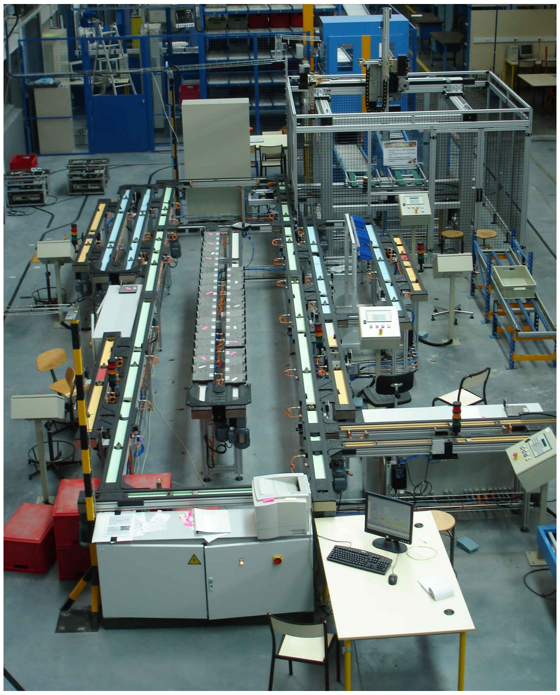


Fig. 3: Picture of the Production Line

of the machine's schedule." This decision rule is called the *operation sequence decision rule*.

- A predictive-reactive schedule based on group sequences. The entrance rule is then: "The part enters the queue of the machine if the queue is not full and its next operation corresponds to one of the operations of the current group of the machine's schedule." This decision rule is called the *group sequence decision rule*.
- A reactive schedule, lying on the concept of "first come, first served." The entrance rule becomes: "The part enters the queue of the machine if the queue is not full and the machine has the capability to perform the next operation of the part." This decision rule is called the *dynamic decision rule*.

IV. Experiments

To verify the efficiency of group sequencing for this real production system, we chose to run benchmarks.

We took a well known benchmark instance called `la14` from [7], because this problem is well adapted to our industrial case and it is possible to introduce a lot of flexibility. This problem is made of five machine and twenty jobs of five operations, *i.e.* 100 operations. This job shop problem instance does not take into account the transportation. These transportation times are the uncertainties of our problem. We focused on the makespan objective.

We compare four different schedules computed for the instance `la14`:

- A schedule obtained with a predictive method. This solution is an optimal solution for the problem without transportation time. The quality of this solution is $C_{\max} = 1292$. This schedule is based upon the *operation sequence decision rule*. It will be referred as the *operation sequence schedule (OSS)*.
- A group sequence where the best case quality is equal to the worst case quality when there are no transportation. In that context, the quality for all schedules is $C_{\max} = 1292$. It is based upon the *group sequence decision rule*. It will be referred as the *optimal group sequence schedule (OGSS)*.
- Another sequence of groups where the worst case quality for the problem without transportation is degraded. In that context, the worst-case quality is $C_{\max} = 1382$ while the best-case quality is $C_{\max} = 1292$. This group sequence gives more flexibility for the choice of operations in the groups. It is based upon the *group sequence decision rule*. It will be referred as the *degraded group sequence schedule (DGSS)*.
- For each machine, a dynamic sequence of jobs obtained with the *dynamic decision rule* based FMS working mode. This sequence will be referred as the *dynamic schedule (DS)*.

To generate these schedules, we only used existing algorithms:

- To solve the classical job shop problem, we used the branch and bound algorithm described in [4].
- To generate group sequences, we used a greedy algorithm that merge two successive groups according to different criteria until no group merging is possible without degradation of the quality. This algorithm begins with a one-operation-per-group sequence computed by the algorithm described in [4]. It is described in [6].

There is a relationship between the transporters' speed and the ratio between the average processing time of an operation and the average transportation distance between two operations. In the problem `la14`, the average processing time of an operation is about 50 tu (time units). In our FMS, the average transportation distance between two operations is about 55 du (distance units). As a matter of fact, if the transporters' speed is greater than 1 du/tu, the average processing time of an operation is greater than the average transportation time between two operations. Thus, the fastest is the transportation system, the the more important are processing times in regards with our objective. Around a transporters' speed of 1 du/tu, processing times has the same scale as transportation times between two operations.

As a matter of fact, we ran numerous tests to measure the quality of the four schedules according to the transporters' speed, named TS (operation times are constant). Numerical results are displayed in Table I while Fig. 5 shows a representation of their evolution.

V. Discussion

A. TS > 1.4 du/tu

In this case, in average, processing times of operations are greater than transportation times between two operations in average. OSS, OGSS and DGSS performances are stable and

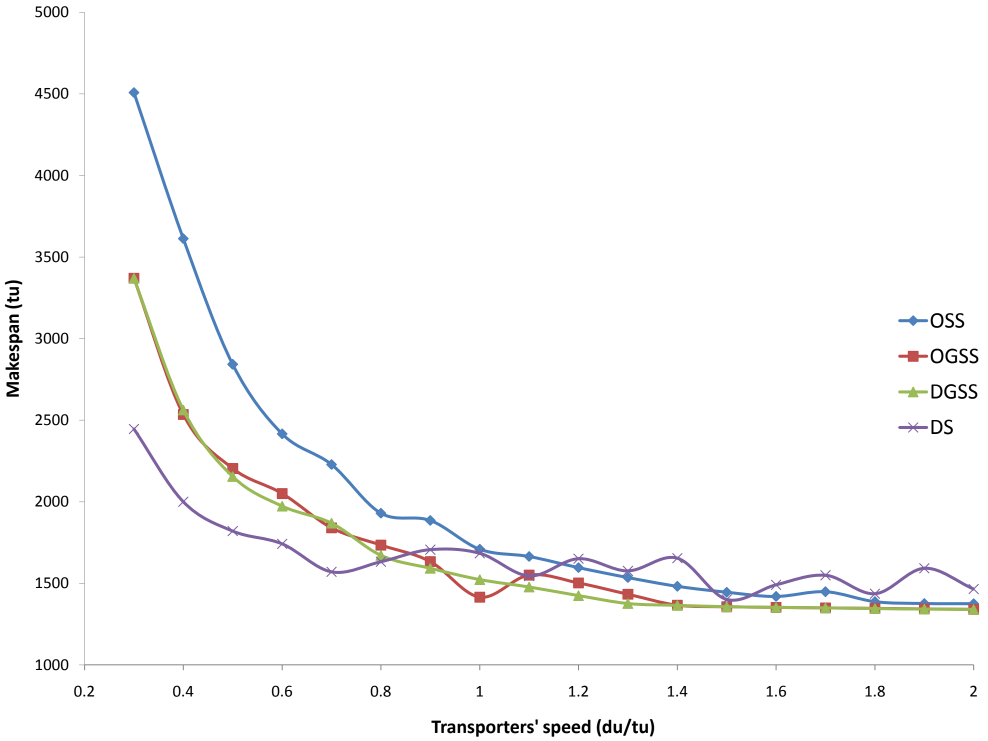


Fig. 5: Results Graphic

tend, at infinite speed, to the optimal makespan of `la14`. This result is quite normal for OSS and OGSS, whereas this was not obvious for DGSS: the schedule on the FMS corresponds to one of the best case semi-active schedules hold by the group sequence used for DGSS. On the other hand, DS has a globally weak and unstable quality compared with the others. Indeed, the decision rule, chosen because of the configuration and the technical constraints of our system, is not very successful for the makespan. Furthermore, its erratic performance is due to the local decisions, which may modify the schedule because of slight modification of TS.

As the transporters decreases, transportation times become more important and OSS gets a worse quality than DGSS and OGSS. This is significant of the absorption of the uncertainties due to the flexibility present in these two last methods.

B. 0.7 du/tu < TS < 1.4 du/tu

In this case, average processing and transportation times are of the same scale. In this situation, OSS has the worst quality. We can conclude that the sequence is not adapted, even compared to DS. DS stays erratic but has competitive performances. OGSS and DGSS have the best performances. This result is very encouraging: it means that group sequencing is robust, even when disturbances on transportation times are high. In average, DGSS is better than OGSS, which can be explained by the fact that DGSS is more flexible.

C. TS < 0.7 du/tu

Finally, let us consider the case when average transportation times are greater than average processing times. In this context, the optimizations done in OSS, OGSS and DGSS are meaningless, as processing times on which optimization is made are insignificant compared to transportation times. Without any surprise, DS comes out on top against the other schedules. The flexibility of DGSS and OGSS limits the below-average performance while the quality of OSS is really bad.

D. Conclusion

First, in these experiments, group sequencing always outperforms static scheduling. This results gives credence to group scheduling: the use of this predictive-reactive schedule is preferable to a predictive one when considering non-negligible transportation times.

Then, group sequencing shows better results than the dynamic approach used in this comparison, until transportation times becomes more relevant than processing times. DS becomes an interesting method when the ratio between processing times and transportation times is unknown, when quality is not important or when transportation times dominate processing times.

Thus, group sequencing is a promising method that allows to bypass transportation times in the studied FMS when processing times are preponderant data of the problem.

TABLE I: Makespan of the Different Schedules in Function of the Speed of the Transporters

Speed (du/tu)	OSS	OGSS	DGSS	DS
0.1	13851	10482	10489	5617
0.2	7582	5101	5093	3255
0.3	4508	3370	3370	2446
0.4	3613	2535	2563	2000
0.5	2842	2204	2155	1820
0.6	2416	2050	1973	1742
0.7	2228	1840	1869	1570
0.8	1930	1734	1671	1632
0.9	1885	1633	1592	1706
1.0	1708	1415	1523	1684
1.1	1664	1550	1477	1545
1.2	1596	1502	1425	1651
1.3	1535	1433	1376	1576
1.4	1481	1366	1366	1654
1.5	1445	1356	1356	1401
1.6	1419	1352	1352	1491
1.7	1448	1349	1349	1549
1.8	1387	1346	1346	1436
1.9	1376	1343	1343	1592
2.0	1375	1340	1340	1465
5.0	1311	1311	1311	1374
10.0	1301	1301	1301	1301
∞	1292	1292	1292	1425

VI. Conclusions and Future Works

A. Conclusions

Group sequencing is a method that enables to add flexibility on the sequence of operations. To verify the possibility of this flexibility to absorb uncertainties, we made some experiments on a real production system.

The results show that group sequencing have good performances on this problem, and enable to optimize the quality of the schedule even if the model does not take into account the transportation times present in the reality.

B. Future Works

It could be interesting to add classical transportation times in the model to have a more precise model, and thus maybe improve the performances of the system.

Experiments with other objective function such as maximum lateness could be interesting to verify if these results are not linked to the makespan objective.

References

[1] Christian Artigues, Jean-Charles Billaut, and Carl Esswein. Maximization of solution flexibility for robust shop scheduling. *European Journal of Operational Research*, 165(2):314–328, September 2005.

[2] Jean-Charles Billaut and François Roubellat. A new method for workshop real-time scheduling. *International Journal of Production Research*, 34(6):1555–1579, 1996.

[3] R. W. Brennan. Performance comparison and analysis of reactive and planning-based control architectures for manufacturing. *Robotics and Computer-Integrated Manufacturing*, 16(2-3):191–200, April 2000.

[4] Peter Brucker, Bernd Jurisch, and Bernd Sievers. A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics*, 49(1-3):107–127, 1994.

[5] Jacques Erschler and François Roubellat. An approach for real time scheduling for activities with time and resource constraints. In R. Slowinski and J. Weglarz, editors, *Advances in project scheduling*. Elsevier, 1989.

[6] Carl Esswein. *Un apport de flexibilité séquentielle pour l'ordonnancement robuste*. Thèse de doctorat, Université François Rabelais Tours, 2003.

[7] S. Lawrence. Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement). Technical report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.

[8] B. L. MacCarthy and Jiyin Liu. A new classification scheme for flexible manufacturing systems. *International Journal of Production Research*, 31(2):299–309, 1993.

[9] H. Pierreval and N. Mebarki. A real-time scheduling approach based on a dynamic selection of dispatching rules. *International Journal of Production Research*, 35(6):1575–1591, 1997.

[10] Michael Pinedo. *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, 2nd edition.

[11] S. David Wu, Eui-Seok Byeon, and Robert H. Storer. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations Research*, 47(1):113–124, 1999.