



**HAL**  
open science

# New Lower Bounds for The Best-Case Schedule in Groups of Permutable Operations

Zakaria Yahouni, N. Mebarki, Z. Sári

► **To cite this version:**

Zakaria Yahouni, N. Mebarki, Z. Sári. New Lower Bounds for The Best-Case Schedule in Groups of Permutable Operations. 15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control in Manufacturing (INCOM'2015) , May 2015, ottawa, Canada. hal-01740852

**HAL Id: hal-01740852**

**<https://hal.science/hal-01740852>**

Submitted on 22 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# New Lower Bounds for The Best-Case Schedule in Groups of Permutable Operations

Z. Yahouni<sup>\*\*\*</sup> N. Mebarki<sup>\*</sup> Z. Sari<sup>\*\*</sup>

<sup>\*</sup> LUNAM, Université de Nantes / IRCCyN, Institut de Recherche en Communications et Cybernétique de Nantes, UMR CNRS 6597, Nantes, France (e-mail: *FirstName.FamilyName@irccyn.ec-nantes.fr*).

<sup>\*\*</sup> Manufacturing Engineering Laboratory of Tlemcen, Tlemcen - Algeria (e-mail: *z.FamilyName@mail.univ-tlemcen.dz*)

---

**Abstract:** Groups of permutable operations is a well-known robust scheduling method that represents a particular set of schedules to be used in a real-time human-machine decision system where the aim is to absorb uncertainties. This method guarantees a minimal quality corresponding to the worst-case. The best-case quality is also of interest; associated with the worst-case, it will provide a range of all possible qualities of the final schedule. The best-case quality is an NP-hard problem that can be solved optimally using an exact method. The performance of this exact method relies on the accuracy of its lower bounds. In this paper, we propose new improved lower bounds for the best-case quality of the groups of permutable operations. These lower bounds can either be used in an exact method to seek for the optimal best solution or can be used in a real-time human-machine decision system. The experiments made on very well-known job shop instances, using the makespan objective, exhibit very good performances.

*Keywords:* Groups of permutable operations, Best-case, Scheduling, Uncertainty, Job Shop, Lower bounds.

---

## 1. INTRODUCTION

In an industrial context the most classical scheduling situation is the predictive scheduling where variables, constraints, all the data are considered as known. The problem is that manufacturing systems are not so deterministic. They present a lot of uncertainties, e.g., the breakdown of a machine, late material, new orders to proceed immediately, etc. During the execution of the schedule, it is frequently necessary to repair the schedule while preserving the solutions quality. In this context, scheduling methods which provide flexible solutions taking into account the uncertainties of the workshop should be very interesting. Groups of permutable operations (GoPO) is one of the most studied methods which provides flexibility.

GoPO is used according to two stages: a predictive phase and a reactive phase. The predictive phase is done offline. It aims at introducing flexibility in the sequence of operations by creating groups of permutable operations which enables to describe a set of schedules without enumerating them. Then, the reactive phase is done online on the floor. It needs the intervention of a human, named the operator, who chooses during the execution of the GoPO the operation to be executed in each group of permutable operations that fits best the real state of the system. This method has been successfully addressed in the literature Erschler and Roubellat (1989); Billaut and Roubella (1996); Esswein (2003); Pinot (2008).

GoPO guarantees a minimal quality corresponding to the worst-case schedule; this value can be computed using a polynomial time algorithm for min-max regular objectives, see Esswein (2003); Artigues et al. (2005). However, taking a decision using only the worst-case value can lead to prioritize lower quality solutions over the good ones. The best-case quality of GoPO should also be interesting by providing to the operator the next operation to be chosen for the optimal final solution. The best-case quality  $Z_{best}$  can be associated with the worst-case quality  $Z_{worst}$  to present the range of all possible qualities of the final schedule  $[Z_{worst} \dots Z_{best}]$ .

Finding the best-case quality in GoPO is an NP-Hard optimization problem. It can be solved optimally using an exact method like the Branch and Bound approach, see Pinot and Mebarki (2009); Yahouni et al. (2014). The efficiency of this method relies directly on the quality of its lower bounds.

Over the years, several adjustment techniques were proposed for the computational of the job shop lower bounds, most of them based on the one machine relaxation problem : Carlier and Pinson (1990, 1994); Baptiste and Le Pape (1996); Nuijten and Pape (1998). Sourd and Nuijten (2000) proposed two classes of adjustment techniques; the first class is the one-machine problems, the second class concerns the conjunctive constraints between operations and consequently machines. Pinot and Mebarki (2008) developed an adaptative lower bounds for GoPO taking into

account the precedence properties between operations and groups on the same machine. In this paper, we take into consideration that operations, groups and machines are connected to each other through precedence constraints to exploit the potential of improving the lower bounds proposed by Pinot.

The remainder of the paper is structured as follows: in Section 2, groups of permutable operations method is described in detail. Next, in Section 3, we present the lower bounds developed by Pinot and Mebarki (2008) and we propose an improvement of these lower bounds. Section 4 is devoted to the evaluation of our lower bounds by a simulation study on a benchmark job shop instances. Finally, main conclusions are summarized in Section 5.

## 2. GROUPS OF PERMUTABLE OPERATIONS

Groups of permutable operations (GoPO) also called group sequence method was first introduced by LAAS-CNRS laboratory, Toulouse, France Erschler and Roubelat (1989). this approach has been used in the ORDO software, it describes a set of valid schedules, without enumerating them. The objective of this method is to provide to the decision-maker a sequential flexibility during the execution of the schedule and to ensure a certain quality that is represented by the worst-case Artigues et al. (2005); Alloulou and Artigues (2007).

A GoPO schedule is composed of groups  $G_i$ , each group contains one or many operations that will be executed in the same resource  $G_i := \{O_i, O_2, \dots, O_n\}$ ,  $O_i$  denotes the operation number  $i$ ,  $n!$  denotes the number of permutations that can be represented from this group. A GoPO schedule is said feasible if any permutation among all the operations of the same group gives a feasible schedule that satisfies all the problem constraints.

To illustrate this definition, let us study a job shop example where the problem is described in Table 1,  $j_i$ ,  $p_i$ ,  $M_i$  and  $\Gamma^-(i)$  denote respectively the job number  $i$ , the processing time, the machine assignment and the predecessor of the current operation)

Table 1. Example of a Job shop problem

$j_i$	$j_1$			$j_2$			$j_3$		
$O_i$	$O_1$	$O_2$	$O_3$	$O_4$	$O_5$	$O_6$	$O_7$	$O_8$	$O_9$
$M_k$	$M_1$	$M_2$	$M_3$	$M_2$	$M_3$	$M_1$	$M_1$	$M_3$	$M_2$
$\Gamma^-(i)$	/	$O_1$	$O_2$	/	$O_4$	$O_5$	/	$O_7$	$O_8$
$p_i$	1	4	1	2	3	1	4	2	3

Table 1 presents a job shop problem with three machines and three jobs, while Fig. 1 presents a feasible GoPO schedule solving this problem for the makespan objective. This GoPO schedule is made of seven groups: two groups of two operations and five groups of one operation. It describes four different semi-active schedules shown in Fig. 2. Note that these schedules do not always have the same makespan: the best-case quality is with  $C_{max}=10$  and the worst-case quality is with  $C_{max}=12$ .

The execution of a GoPO schedule consists in choosing a particular schedule among the different possibilities described by the GoPO. It can be viewed as a sequence of decisions: each decision consists in choosing an operation

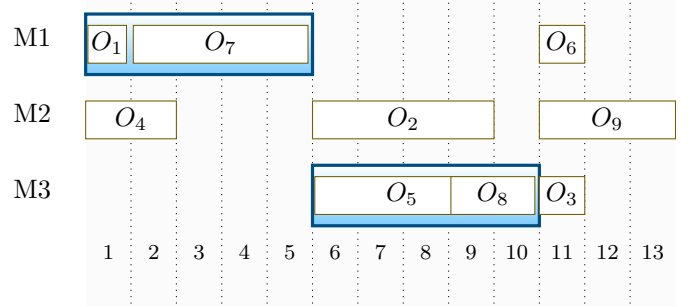


Fig. 1. GoPO Schedule

to execute in a group when this group is composed of two or more operations. For instance, for the schedule described on Fig. 1, there are two decisions to be taken: on  $M_1$ , at the beginning of the scheduling, either operation  $O_2$  or  $O_7$  has to be executed. Let us suppose the decision taken is to schedule  $O_2$  before  $O_7$ , on  $M_2$ , there is another decision: scheduling operation  $O_5$  or  $O_8$  first, so at the end we have four semi-active schedules.

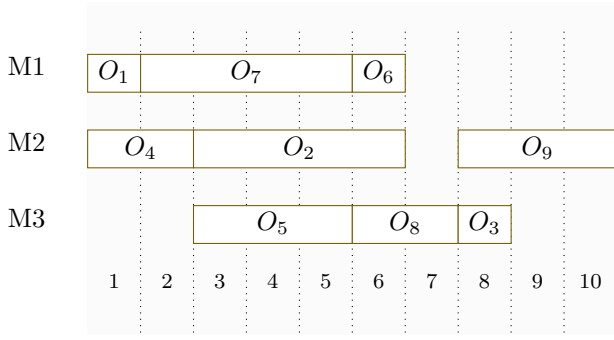
GoPO has an interesting property: the quality in the worst-case can be computed in real-time for minmax regular objective functions like makespan Esswein (2003); Artigues et al. (2005); Alloulou and Artigues (2007). Thus, this method can be used to compute the worst-case quality in real-time during the execution of the schedule even for large scheduling problems.

To summarize, it can be said that GoPO enables the description of a set of schedules in an implicit manner (i.e. without enumerating the schedules) and guarantees a minimal performance that corresponds to worst-case quality. But the best-case quality should also be interesting to know which operation to choose from a current group to get possibly the best optimal schedule.

## 3. LOWER BOUNDS FOR THE BEST-CASE STARTING/COMPLETION TIME OF AN OPERATION

To compute the best-case quality, Pinot and Mebarki (2008) proposed a branch and bound method to solve the problem, which needs the computational of the lower bounds. The accuracy of these lower bounds is very important for two reasons: The most accurate are the lower bounds, the more efficient will be the branch and bound procedure. Also, as the best-case quality is an NP-hard optimisation problem, it is sometimes very useful to use these lower bounds directly to take the scheduling decisions during the reactive phase.

Pinot and Mebarki (2008) compute such lower bounds using a relaxation on the resources by making the assumption that each resource has an infinite capacity. In this case, the best-case lower bound for starting time of an operation ( $\theta_i$ ) is computed as the maximum of the best-case (lower bound) completion time ( $\chi_j$ ) of all its predecessors: for an operation  $O_i$ , its predecessors include the predecessors given by the problem ( $\Gamma^-(i)$ ) but also the group predecessor of this operation on the same machine. Thus, it needs the computation of the optimal makespan of



$$\begin{cases} \theta_i = \max(r_i, \gamma_{g^-(i)}, \max_{j \in \Gamma^-(i)} \chi_j) \\ \chi_i = \theta_i + \rho_i \\ \gamma_{g_{l,k}} = C_{max} \text{ of } 1|r_i|C_{max}, \forall O_i \in g_{l,k}, r_i = \theta_i \end{cases} \quad (1)$$

In our Job shop example, the predecessors of operation  $O_6$  executed on  $M_1$  are: operation  $O_5$  (executed on  $M_3$ ) because of the precedence constraints, and the optimal makespan of the ending time of the group containing  $O_1$  and  $O_7$  (executed on the same machine  $M_1$  because they are in the previous group ( $g^-(6)$ ). So we have:

Table 2. Lower bounds

$O_i$	$\theta_i$	$\chi_i$	$\gamma_{g_{l,k}}$
$O_1$	0	1	5
$O_7$	0	4	5
$O_4$	0	2	2
$O_2$	2	6	6
$O_5$	2	5	7
$O_8$	4	6	7
$O_3$	7	8	8
$O_6$	5	6	6
$O_9$	6	9	9

Table 2 represents the lower bounds best-case starting/completion time of our job shop example. The maximum lower bound completion time of operations is nine which is not the optimal value.

### 3.1 Precedence dominance constraint

In this section, we present an improvement of the previous lower bounds using an adjustment technique based on the combination between precedence constraints and disjunctive constraints between groups on the same machine.

This improvement is based on a property concerning any two operations ( $O_i$  and  $O_j$ ) that belongs to two successor groups ( $G_i^-$  and  $G_i$ ) in the same machine and have their predecessors (direct or indirect) in the same group ( $O_i^-$  and  $O_j^- \in G_{l,k}$ ). In this case, one of the operations  $O_i^-$  or  $O_j^-$  has to start before the other one. Executing one of these operations at first and delaying the other operation will have an influence on the completion time of  $G_i$ . Fig. 3 and Fig. 4 shows this precedence property for our job-shop example:

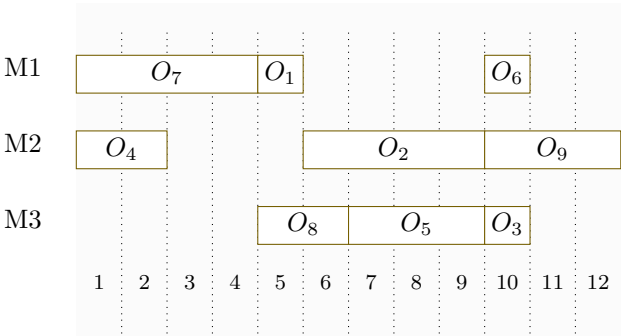
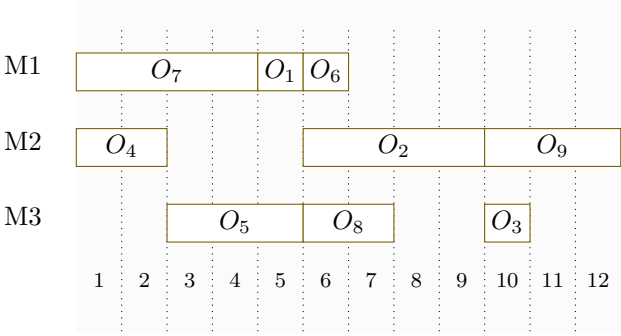
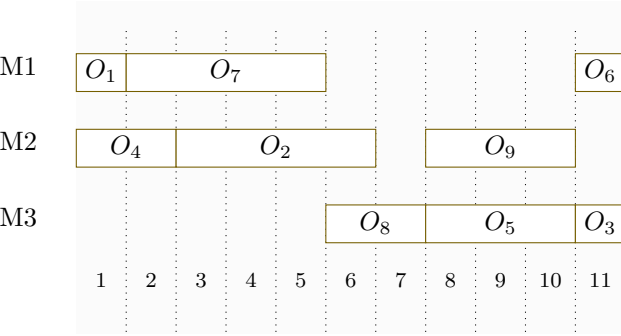


Fig. 2. Set of Semi-active schedules

a group (named as  $\gamma_{g_{l,k}}$ ). First we compute  $\theta_i$  as a release date so we can generate a  $1|r_i|C_{max}$  problem instance that corresponds to our problem, with  $r_i = \theta_i$ , see equation(1). This problem is polynomially solvable by ordering the operations in ascending release date Lawler (1973); Brucker and Knust (2007).

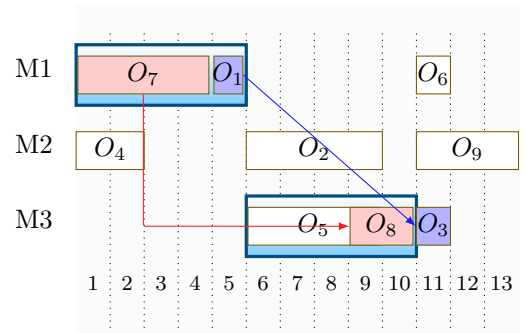


Fig. 3. Precedence dominance constraint between  $O_8$  and  $O_3$

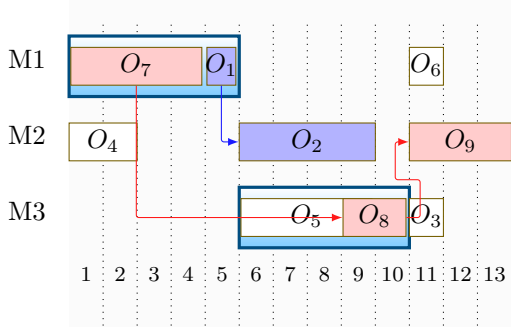


Fig. 4. Precedence dominance constraint between  $O_2$  and  $O_9$

From the example shown on Fig. 4, we have on Machine  $M_2$  operation  $O_9$  that has to start after  $O_8$  because of the precedence constraint, and after operation  $O_2$  because of the group precedence in the same machine. Both predecessors of these operations  $O_1$  and  $O_7$  are in the same group, executing one of them first will have an influence on the starting time of operation  $O_9$ .

Computing the best-case starting time of  $O_9$  using equation (1) will lead to put  $O_2$  and  $O_8$  on their both best starting time, as well as  $O_7$  and  $O_1$  by transitivity, which is impossible, we know for fact that  $O_7$  and  $O_1$  cannot start together at their best starting time on the same time (only one of them can starts at it best-case), This is why using equation (1) to compute  $\theta_9$  has given six on Table 2, which is not the optimal solution = seven as shown on Fig. 2.

To take into account this situation, we propose an adjustment technique, the calculation of the new lower bounds requires two times the calculation of the makespan using equation (1), one time for the case when  $O_1$  is executed before  $O_7$  and the other one when  $O_7$  is executed before  $O_1$ . The new makespan lower bound will be the minimum value between these two evaluations. The new lower bounds for our job shop example are shown on table 3.

Table 3. Improved Lower bounds

$O_i$	$r_1 < r_7$			$r_7 < r_1$			min ( $\gamma_{1_{g_l,k}}, \gamma_{2_{g_l,k}}$ )
	$\theta_{1_i}$	$\chi_{1_i}$	$\gamma_{1_{g_l,k}}$	$\theta_{2_i}$	$\chi_{2_i}$	$\gamma_{2_{g_l,k}}$	
$O_1$	0	1	5	4	5	5	5
$O_7$	1	5	5	0	4	5	
$O_4$	0	2	2	0	2	2	2
$O_2$	2	6	6	5	9	9	6
$O_5$	2	5	7	2	5	7	7
$O_8$	5	7	7	4	6	7	
$O_3$	7	8	8	9	10	10	8
$O_6$	5	6	6	5	6	6	6
$O_9$	7	10	10	9	12	12	10

Our new lower bounds give the optimal value for the best-case schedule of our job shop example as shown on Table 3.  $\theta_i$  can be computed on polynomial time using equation (1), its complexity is equal to  $O(n^2)$  ( $n$  being the number of operations). We suppose that for the worst-case the precedence constraints property is found  $k$ -time ( $k < n$ ) when computing  $\theta_i$ . So we have a complexity of  $kO(n^2) =$

$O(n^3)$ . In the next section we will present an experiment on a benchmark job shop instances to evaluate our new lower bounds.

#### 4. COMPUTATIONAL TESTS

We took a well-known set of benchmark instances called la01 to la40 from Lawrence (1984) with known optimal values. These instances are widely used in the job shop literature. For each instance, we generate a group sequences with known flexibility value using a greedy algorithm called EBJG Esswein (2003) that merges two successive groups according to different criteria until no group merging is possible. This algorithm begins with a one-operation-per-group sequence. We refer to the lower bounds used in Pinot and Mebarki (2008) by  $LB_1$  and the improved ones  $LB_2$ . We study the accuracy of each of these lower bounds compared to the optimal solution, outcomes of the experiment are shown on the next table.

Table 4.  $LB_1$  VS  $LB_2$

	GAP error $LB_1$	GAP error $LB_2$
La1	35	0
La2	0	0
La3	9	3
La4	2	2
La5	0	0
La6	0	0
La7	17	17
La8	0	0
La9	0	0
La10	0	0
La11	0	0
La12	0	0
La13	0	0
La14	0	0
La15	0	0
La16	0	0
La17	23	10
La18	0	0
La19	1	1
La20	1	1
La21	0	0
La22	0	0
La23	0	0
La24	1	1
La25	1	0
La26	0	0
La27	0	0
La28	0	0
La29	6	3
La30	0	0
La31	0	0
La32	0	0
La33	0	0
La34	0	0
La35	0	0
La36	1	1
La37	2	0
La38	1	1
La39	1	1
La40	0	0
$\Sigma$	101	41

Based on the results,  $LB_1$  gives optimal values on 26 instances, where  $LB_2$  has an advancement of six instances

over  $LB_1$ . Table 4 shows that  $LB_2$  provides a lower maximum gap error for almost all instances. The sum gap error for the test problems is 41, which is more than two times smaller than the sum gap error of  $LB_1 = 101$ . Our lower bounds always performed either the same or better than  $LB_1$  depends on whether the precedence dominance constraint between groups is present on the GoPO or not. The improvement of the results are shown on six instances (La01,La02,La17,La25,La29 and La37).

The computational time for all instances together for  $LB_1$  and  $LB_2$  is less than one second. For almost all instances this value converge to zero for both lower bounds.

In our next experiment, we use these lower bounds in a branch and bound method B&B on the same instances. The protocole experiment is described in Yahouni et al. (2014). The results of this experiment are shown on Table 5. The columns are representing the number of processed nodes to find the optimal best solution using  $LB_1$  and  $LB_2$ . The number of visited nodes to find the best-case is the sum of the number of nodes to find the optimal solution in the tree and the number of nodes to prove it optimality.

The results of experiment shows that B&B  $LB_2$  has an advance over B&B  $LB_1$  on three instances (La17, La25 and La 29) where the number of visited nodes has been improved more than ten times. The sum number of visited nodes for the best-case schedule for all instances using B&B  $LB_1$  and B&B  $LB_2$  are consecutively 254498 and 247593 (GAP = 6905).

## 5. CONCLUSION

In this paper we have proposed an adjustment technique to improved the lower bounds for the computational of the best-case schedule in groups of permutable operations for the makespan objective. Our improved lower bound is based on the precedence dominance constraints between two successive groups on the same machine. Experiements were done on benchmark job-shop instances showed the efficiency of our improved lower bounds compared with the one developed by Pinot and Mebarki (2008).

These lower bounds can either be used implicitly in a real-time human-machine decision system or in a branch and bound algorithm. However, the effectiveness of the branch and bound method relies directly on the accuracy of its lower bounds. For this, we may explore in further research more dominance rules to improve these lower bounds by extending our precedence dominance constraints on multiple groups instead of only two successive groups.

## REFERENCES

Alloulou, M. and Artigues, C. (2007). Worst-case evaluation of flexible solutions in disjunctive scheduling problems. *Computational Science and Its Applications - ICCSA 2007 International Conference, Proceedings, Part III*, 1205, 1027–1036.

Artigues, C., Billaut, J., and Esswein, C. (2005). Maximization of solution flexibility for robust shop scheduling. *European Journal of Operational Research*, 165, 314–328.

Table 5. B&B  $LB_1$  VS B&B  $LB_2$

	B&B $LB_1$	B&B $LB_2$
La01	15	15
La02	9	9
La03	15	15
La04	16	16
La05	18	18
La06	27	27
La07	214862	214862
La08	26	26
La09	30	30
La10	29	29
La11	33	33
La12	34	34
La13	33	33
La14	31	31
La15	36	36
La16	23	23
La17	837	521
La18	21	21
La19	248	248
La20	22	22
La21	36	36
La22	32	32
La23	36	36
La24	34	34
La25	5793	36
La26	50	50
La27	52	52
La28	58	58
La29	891	59
La30	55	55
La31	102	102
La32	106	106
La33	100	100
La34	100	100
La35	99	99
La36	8487	8487
La37	42	42
La38	21651	21651
La39	370	370
La40	39	39
$\Sigma$	254498	247593

Baptiste, P. and Le Pape, C. (1996). Edge-finding constraint propagation algorithms for disjunctive and cumulative scheduling. In *Scheduling, Proceedings 15th Workshop of the U.K. Planning Special Interest Group*.

Billaut, J.C. and Roubella, F. (1996). A new method for workshop real time scheduling. *International Journal of Production Research*, 34, 1555–1579.

Brucker, P. and Knust, S. (2007). Complexity results for scheduling problems. URL <http://www.mathematik.uni-osnabrueck.de/research/OR/class/>.

Carlier, J. and Pinson, E. (1990). A practical use of jacksons preemptive schedule for solving the job shop problem. *Annals of Operations Research*, 26, 269287.

Carlier, J. and Pinson, E. (1994). Adjustment of heads and tails for the job-shop problem. *European J. Operational Research*, 78, 146161.

Erschler and Roubellat (1989). An approach for real time scheduling for activities with time and resource constraints. In Slowinski, R. and Weglarz, J., editors, *Advances in project scheduling*. Elsevier.

- Esswein, C. (2003). *Un apport de flexibilité séquentielle pour l'ordonnancement robuste*. Ph.D. thesis, Université François Rabelais Tours(France).
- Lawler, E.L. (1973). Optimal sequencing of a single machine subject to precedence constraints. *Management Science*, 19, 544–546.
- Lawrence, S. (1984). Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement).
- Nuijten, W. and Pape, C.L. (1998). Constraint-based job shop scheduling with ilog sheduler. *J. of Heuristics*, 3, 271286.
- Pinot, G. (2008). *Coopération homme-machine pour l'ordonnancement sous incertitudes*. Ph.D. thesis, Université de Nantes (France).
- Pinot, G. and Mebarki, N. (2008). Best-case lower bounds in a group sequence for the job shop problem. In *17th IFAC World Congress*. Seoul, Korea.
- Pinot, G. and Mebarki, N. (2009). An exact method for the best case in group sequence: Application to a general shop problem. In *13th IFAC Symposium on Information Control Problems in Manufacturing*. Moscow, Russia.
- Sourd, F. and Nuijten, W. (2000). Multiple-machine lower bounds for shop-scheduling problems. *INFORMS Journal on Computing*, 12(4), 341–352.
- Yahouni, Z., Mebarki, N., and Sari, Z. (2014). A new approach for the best-case schedule in a group sequence. 10 International Conference of Modeling and Simulation, Nancy, France.