Minimizing tardiness for job shop scheduling under uncertainties

Zakaria YAHOUNI
*LUNAM, Nantes university / IRCCyN
UMR CNRS 6597, Nantes, France
zakaria.yahouni@irccyn.ec-nantes.fr
**Manufacturing Engineering Laboratory
of Tlemcen (MELT), University of
Tlemcen, Algeria

Nasser MEBARKI
LUNAM, Nantes university / IRCCyN
UMR CNRS 6597, Nantes, France
nasser.mebarki@univ-nantes.fr

Zaki SARI
Department of Industrial Engineering
Izmir University of Economics, Turkey
zaki.sari@izmirekonomi.edu.tr

***Abstract*—Many disturbances can occur during the execution of a manufacturing scheduling process. To cope with this drawback, flexible solutions are proposed based on the offline and the online phase of the schedule. Groups of permutable operations is one of the most studied flexible scheduling methods bringing flexibility as well as quality to a schedule. The online phase of this method is based on a human-machine system allowing to choose in real-time one schedule from a set of schedules that fits best the real state of the system. In this paper, we propose and evaluate a new criterion called the best-case in order to be used in real-time during the online phase of the groups of permutable operations. This criterion offers an optimal or near-optimal solution from a set of solutions. The usefulness of this criterion is showed using a comparative review with two other criteria on a benchmark instances using the maximum tardiness objective.**

I. INTRODUCTION

We consider the manufacturing job shop scheduling problem where a set of operations has to be scheduled on a set of machines without preemption. We have n jobs $J_1, J_2, ..., J_n$ to be processed on m machines $M_1, M_2, ..., M_m$, each machine can treat only one operation at a time. Job i consists of n_i operations. Associated with every operation O_i : a machine allocation μ_i , a starting time t_i, a release date r_i, a processing time p_i, a due date d_i and a completion time C_i. Γ_i^- and Γ_i^+ denote respectively the predecessor and the successor of a given operation. Generally, the job shop problem uses a regular objective function f that is a non decreasing function of the C_i. In this work, we focus on the maximum tardiness objective $T_{max} = \max_{i\in\{0..n\}}(C_i - d_i, 0)$.

During the scheduling process, two phases are considered; the $offline\ phase$, called also the $predictive\ phase$, where a static schedule is generated based on the available information. Then, this schedule will be established in real-time during the $online\ phase$ ($reactive\ phase$) of the scheduling process.

Unfortunately, manufacturing systems are not so deterministic. They usually operate in highly dynamic and uncertain environments where many disturbances may occur, like arrival of new jobs, machine breakdown, variation of processing time, etc. [1, 2, 3]. Even that real problems are dynamic and non-deterministic in nature, most of the solutions in the literature use static and deterministic models [4]. [5, 6, 7, 8] are among who first considered the scheduling problem under uncertainties. Several scheduling solutions has been proposed under various names: proactive scheduling, real-time scheduling, predictive-reactive scheduling, proactive-reactive scheduling [2].

In this paper, we tackle the Groups of Permutable Operations method (GoPO) also called group sequence. This method is one of the most studied scheduling method to cope with the perturbations of the shop [9, 10, 11, 12, 13, 14, 15, 16].

This method was created at the LAAS-CNRS laboratory of Toulouse [10]. The goal of this method is to provide during the predictive phase a sequential flexibility characterizing a set of schedules allowing the operator to choose during the reactive phase one schedule which fits best to the real state of the shop. This method has an interesting property; It can guarantee a minimal quality corresponding to the worst-case (worst possible schedule). This evaluation of the worst-case can be computed using a polynomial time algorithm [17]. Thus, it can be used as a decision criterion during the reactive phase of GoPO.

The literature review has primarily focused on the worst-case performance as to be used during the reactive phase of GoPO. In this paper, we are interested in the use of the best-case criterion. This best-case represents the best possible permutation in GoPO leading to the best final schedule for a given objective. The calculation of this final schedule is based on adapted lower bounds measuring the best completion time of operations and groups in a GoPO schedule [18]. This evaluation can be done in real-time and thus can be used during the reactive phase of GoPO.

In this paper, we have proposed a comparative study of the usefulness of the best-case criterion regarding the worst-case criterion and the free sequential margin criterion introduced by [19]. For the experimental protocol, we have developed a new branching algorithm using these criteria. This algorithm plays the role of a human in taking the decisions of a GoPO schedule.

The remainder of the paper is structured as follows: in section 2, the offline phase of GoPO is described in detail. Next, in section 3 and section 4, the reactive phase of GoPO is developed using a branching algorithm for the three criteria. section 5 is devoted to the experiment study. Finally, main conclusions are summarized in section 6.

978-1-5090-2188-8/16/$31.00 ©2016 IEEE

II. OFFLINE PHASE OF GOPO

The offline phase of GoPO, also called the proactive phase, aims at computing a flexible solution offline. This solution is defined as a sequence of groups (of permutable operations) on each machine $M_m : g_{m,1}, ..., g_{m,k}$, performed in this particular order. Every group contains one or many operations that can be executed in an arbitrary order. A GoPO schedule is feasible if for each group, all the permutations among all the operations of the same group give a feasible schedule, i.e, a schedule which satisfies all the constraints of the problem. This solution is a partial-solved schedule characterizing a set of schedules

To illustrate this problem, let us study a job shop example :

TABLE I: Example of a job shop problem

$\mathbf{j_i}$	$\mathbf{O_i}$	μ_i	$\mathbf{\Gamma_i^-}$	ρ_i	$\mathbf{d_i}$
	O_1	M_1	/	1	5
$\mathbf{j_1}$	O_2	M_2	O_1	4	7
	O_3	M_3	O_2	1	10
	O_4	M_2	/	2	2
$\mathbf{j_2}$	O_5	M_3	O_4	3	8
	O_6	M_1	O_5	1	9
	O_7	M_1	/	4	5
$\mathbf{j_3}$	O_8	M_3	/	2	8
	O_9	M_2	/	3	10

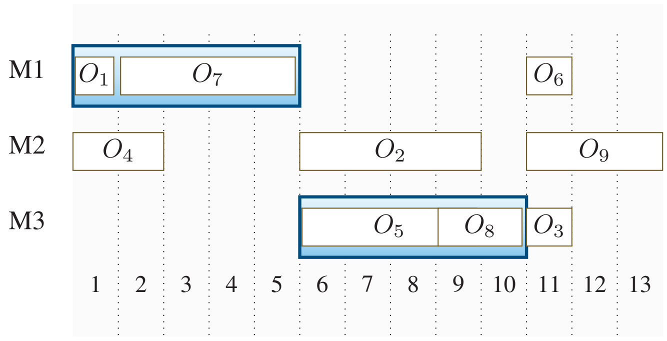


Fig. 1: Groups of Permutable Operations solution.

Table I presents a job shop problem with three machines and three jobs, while Fig. 1 presents a feasible GoPO solving this problem. This GoPO is made of seven groups: two groups of two operations and five groups of one operation. The execution of this GoPO during the online phase consists in choosing a particular schedule among the different possibilities described by GoPO.

III. ONLINE PHASE OF GOPO

The reactive phase of GoPO needs the intervention of a human, named the operator, who chooses during the execution of a GoPO solution, the order of operations to be executed in each group of permutable operations. This phase can be viewed as a sequence of decisions: each decision consists in choosing an operation to execute in a group when this group is composed of two or more operations. For instance, for the solution described on Fig. 1, there are two decisions to be taken: on M_1, at the beginning of the scheduling, either operation O_1 or O_7 has to be executed. Let us suppose that the decision taken is to schedule O_1 before O_7. There is another decision to be taken on M_2 : scheduling operation O_5 or O_8 first, so at the end we have four different semi-active schedules shown in Fig. 2. Note that these schedules do not always have the same performance: $T_{max}(a) = 0$, $T_{max}(b) = 3$, $T_{max}(c)$ and $T_{max}(d) = 2$.

The decision of ordering the operations in each group can be freely chosen in order to fit best the real state of the shop taking into account the perturbations. To help taking these decisions, different criteria may be offered to the operator. These criteria are described bellow:

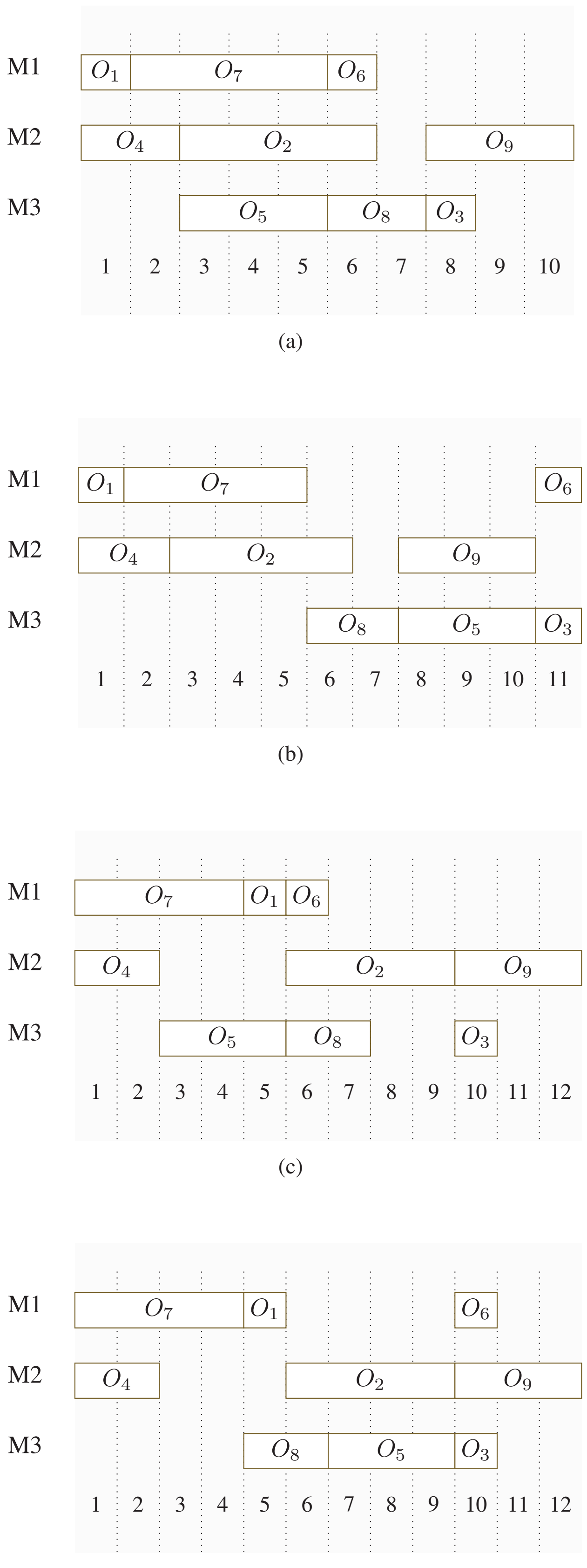
A. *Free sequential margin criteria*

To evaluate the tardiness, [19] present an adaptation of the free margin to GoPO. This measure computes for an operation according to its earliest execution, the maximum tardiness which ensures that all schedules enumerated in GoPO will present no tardiness. Moreover, choosing the operation with the highest free sequential margin in a group may permit to increase the margins of the other operations of the group, and thus enable to preserve the flexibility of the schedule. The free sequential margin of an operation has two components, the operations net margin, which is related to the operation itself regardless the other operations of the group, and the operations group margin, which is related to the other operations of the group.

During the reactive phase, several situations may occur:

- All the free sequential margins of the current group are positive, in that case whatever the chosen operation, the schedules will present no tardiness.
- There is one or several (but not all) operations in the group which present negative (or zero) free sequential margin. In that case, there may be schedules with tardiness, especially those beginning with an operation with a negative free sequential margin and [9] recommends executing operations with large free sequential margins in order to increase the negative margins, trying to make them become positive.
- All the operations of the group have negative (or zero) free sequential margins. In that case, there will be schedules with tardiness whatever the chosen operation, but it is also possible to have schedules with no tardiness.

Thus, the free sequential margin is a precious criterion to choose in real-time an operation during the reactive phase. But the major drawback of this criterion is that it does not permit to know if there is a schedule with no tardiness in case of one or several (but not all) negative free sequential margin(s) in a group.



(a)

(b)

(c)

(d)

Fig. 2: Possible final schedules

B. Worst-case criteria

The quality of GoPO schedule is measured as the quality of the worst semi-active schedule found in GoPO. This value is calculated in polynomial time using the latest completion time of operations and groups. The maximum value of these latest completion times leads to the worst possible permutation for a min-max regular objective function like the makespan and the maximum tardiness objective [17, 20]. Thus, it is possible to use the worst-case criterion in real-time during the reactive phase of GoPO. The main drawback of this criterion is that it does not permit to know which operation is closest to the optimal or near-optimal solution.

C. Best-case criteria

We have proposed and implemented the criterion of the best-case evaluation for measuring the quality of GoPO. The problem of the best-case quality consists of finding the best possible permutation leading to the optimal schedule for a regular objective. This evaluation is an NP-Hard optimization problem that can be seen as a partial-solved scheduling problem. Tight lower bounds for the job shop scheduling problem have been proposed, the algorithm is described in [18, 21]. Using this best-case evaluation during the reactive phase of GoPO should be of good interest for different reasons:

- It permits to know, during the execution of the GoPO solution the next optimal operation to be chosen in each group of permutable operations.
- It also permits to know, if there is at least no schedule in GoPO that has no late jobs.
- Associated with the worst-case value, it permits to better represent the quality of GoPO. This quality can be measured by a range of all possible performances $[Z_{worst}...Z_{best}]$, Z_{worst} and Z_{best} denote respectively the worst-case performance and the best-case performance for a scheduling objective.

Our main objective in this paper, is to evaluate the usefulness of the best-case criteria during the reactive phase of GoPO.

IV. REACTIVE ALGORITHM USING THE BEST-CASE

In quantifying the performance of our proposed criterion (best-case) during the reactive phase of GoPO, we proposed a branching algorithm playing the role of an operator and sequencing the operations using the best-case criteria; For each group, a decision is made by selecting the operation having the minimum lower bound for the best case ($best-case(O_i)$). The process of this algorithm ($LB\_BA\_best-case$) is described bellow ;

```
$L(g) = \{g_{m,k}, g_{m',k'}, ...g_{m'',k''}\}$;
($g_{m,k}$ is a group containing more than one operation)
$LB\_best - case := maximumvalue$;
for every group $g_{m,k}$ in $L(g)$ do
    while $Card(g_{m,k}) > 1$ do
        for every operation $O_i$ in $g_{m,k}$ do
            - Put $O_i$ first ;
            - Calculate $LB\_best - case(O_i)$ ;
        end
        - $LB\_best - case :=$
        $\min_{O_i \in g_{m,k}}(LB\_best - case(O_i))$;
        - Remove $O_i$ from $g_{m,k}$;
    end
    - remove $g_{m,k}$ from $L(g)$.;
end
```

Algorithm 1: $BA\_best - case$ for the reactive phase of GoPO

As an illustration of the algorithm, let us enumerate all the groups of our job shop GoPO example.

$$g_{1,1} : \{O_1, O_7\}\ ,\ g_{1,2} : \{O_6\}\ ,\ g_{2,1} : \{O_4\}\ ,\ g_{2,2} : \{O_2\}\ ,$$
$$g_{2,3} : \{O_9\}\ ,\ g_{3,1} : \{O_5, O_8\}\ ,\ g_{3,2} : \{O_3\}$$
$$L := \{g_{1,1}, g_{3,1}\}$$

The branching procedure is executed on the first group of the list $g_{1,1} : \{O_1, O_7\}$:

- $LB\_best-case(O_1) = 0$ and $LB\_best-case(O_7) = 2$) (using [18]).
- Remove O_1 from the group and update $LB\_best - case = Min(0, 2) = 0$
- The current group contains only O_7 and is removed from the list.

Then the branching procedure is executed on the second group of the list $g_{3,1} : \{O_5, O_8\}$:

- $LB\_best-case(O_5) = 0$ and $LB\_best-case(O_8) = 3$
- Remove O_5 from the group
- The current last group contains only O_8 and is removed from the list.

At the end of the reactive phase, we have an optimal schedule (Fig. 2.a) with no tardiness.

Similarly to $BA\_best - case$, two new branching algorithms using the worst-case $BA\_worst - case$ and the free sequential margin $BA\_free - seq - margin$ criteria are generated; For $BA\_worst - case$, the operation having the minimum worst-case value is chosen to be executed first. And for $BA\_free - seq - margin$, the operation having the maximum free sequential margin is chosen to be executed first.

V. EXPERIMENT

In this section, we assess the relative importance of the three algorithms to their overall performances on a well-known job shop instances.

TABLE II: Comparative results

	NB_Dec	BA_worst-case	BA_free-seq-margin	BA_best-case
La01	31	150	132	0
La02	32	111	0	0
La03	35	297	122	0
La04	36	14	21	0
La05	35	105	74	0
La06	56	136	129	0
La07	58	218	154	0
La08	59	405	197	0
La09	59	137	78	0
La10	56	139	121	0
La11	80	117	112	0
La12	79	343	210	0
La13	82	383	116	0
La14	81	0	0	0
La15	78	295	83	0
La16	57	0	0	0
La17	64	0	0	0
La18	60	0	0	0
La19	61	0	0	0
La20	61	0	0	0
La21	96	0	0	0
La22	101	596	0	0
La23	96	65	0	0
La24	100	0	0	0
La25	100	162	7	0
La26	147	711	78	0
La27	144	348	150	0
La28	137	265	32	0
La29	140	563	189	0
La30	142	413	44	0
La31	231	473	151	0
La32	236	705	191	0
La33	236	936	220	0
La34	228	281	73	0
La35	227	614	0	0
La36	137	4	0	0
La37	141	0	0	0
La38	137	393	0	0
La39	139	1705	0	0
La40	139	1924	0	0

A. *Protocol of the experiment*

We took a set of benchmark instances called la01 to la40 from [22] with a well-known optimal solutions for the makespan objective [14, 23, 24]. These instances are widely used in the job shop literature. These are classical job shop composed of 40 instances of different sizes (5 instances for each size).

For each instance, we generate GoPO schedule with high flexibility. For the generation of GoPO schedules, we used a greedy algorithm that merged two successive groups according to different criteria until no group merging is possible. The greedy algorithm is described in [14].

For each problem type, we compare the schedule obtained by the three algorithms $BA\_worst - case$, $BA\_free - seq - margin$ and $BA\_best - case$. These algorithms are compared in terms of the maximum tardiness objective value.

B. *Results and discussion*

The results of these experiments are exposed on table II . For each instance, we give the number of decisions (NB_Dec) and the GAP tardiness between the final schedule and the best solution found using the three algorithms ($BA\_worst - case$, $BA\_free - seq - margin$ and $BA\_best - case$).

The data presented in table II show that $BA\_best - case$ dominates all the instances compared to seventeen instances for

$BA\_free-seq-margin$ and nine instances for $BA\_worst-case$. The total performance of $BA\_free-seq-margin$ overpass almost five times the performance of $BA\_worst-case$.

It is obviously that the worst-case criterion gave the worst performance over the other algorithms. Overall, this is not that surprising in light of what is expected; This criterion is more effective in controlling the performance of the final schedule than guaranteeing an optimal or near-optimal final schedule. However, it can be concluded that using the worst-case criterion alone lead the operator to prioritize lower quality solutions over the good ones.

The free sequential margin criterion has been shown to be effective for almost half of the experiment instances. But it is proven from the result that this criterion is myopic; because it focuses only on the current group to choose the next operation to execute, and neglect the future decisions on the other groups. In case of two positive margin in a current decision, this criterion does not permit to know which one is the optimal decision contrarily to the best-case criterion. The major drawback of this criterion compared to the others, is that it can be used only on tardiness or latest objectives. It can be suggested from this result that this criterion should be better integrated with the best-case criterion to get closer to the optimal or near-optimal solution.

VI. CONCLUSION

In this article, we study the efficiency of the best-case criterion. This criterion can be used during the execution of GoPO, the evaluation must be done in real-time. The computation of this criterion is based on an adapted proposed lower bounds using the maximum tardiness objective. These lower bounds are calculated at each event of the manufacturing system where the state of GoPO should be updated, and then, every decision should be evaluated: for all operations in the groups to be executed, three criteria are offered for taking better decisions.

We demonstrated the benefits of the proposed criterion on a well-known job shop benchmark instances. The comparative result shows clearly that the best-case criterion exhibits better performance than the worst-case and the free sequential margin criteria.

For further research, the proposed criteria should be implemented in a real decision-aid system involving the human temper factor in order to confirm and extend the theoretical results.

REFERENCES

[1] G. Y. P. Kouvelis, “Robust discrete optimisation and its applications.” 1997, kluwer Academic.

[2] J. B. A.J. Davenport, “A survey of tehniques for sheduling with uncertainty,” 2000. [Online]. Available: http://www.eil.utoronto.ca/profiles/chris/chris.papers.html

[3] J. Billaut, A. Moukrim, and E. Sanlaville. Hermès, 2005, ch. Introduction à la fexibilité et à la robustesse en ordonnancement, pp. 15–34.

[4] I. Sabuncuoglu and S. Goren, “Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research,” *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 2, pp. 138–157, 2009.

[5] J. Birge, M. Dempster, H. Gassman, E. Gunn, A. King, and S. Wallace, “A standard input format for multi-period stochastic linear programs,” *Mathematical Programming Society*, vol. 17, pp. 1–21, 1987.

[6] G. Gallego, “Linear control policies for scheduling a single facility after an initial disruption,” School of operations research and industrial engineering, College of Engineering, Cornell univeresity, ITHACA, New york 14853, Tech. Rep., January 1988.

[7] ——, “Produce-up-to policicies for scheduling a single facility after an initial disruption,” School of operations research and industrial engineering, College of Engineering, Cornell univeresity, ITHACA, New york 14853, Tech. Rep., January 1988.

[8] J. Bean, J. Birge, J. Mittenthal, and C. Noon, “Match-up scheduling with multiple resources, release dates and disruptions,” *Option research*, vol. 39(3), p. 470483, 1991.

[9] V. Thomas, “Aide la décision pour lordonnancement datelier en temps rel,” Ph.D. dissertation, Université Paul Sabatier, 1980.

[10] R. Erschler, “An approach for real time scheduling for activities with time and resource constraints,” 1989, in Slowinski, R. and Weglarz, J., editors, Advances in project scheduling. Elsevier.

[11] C. Le Pape, “Constraint propagation in planning and scheduling,” Ph.D. dissertation, Robotics Laboratory, Department of Computer, stanford, 1991.

[12] J.-C. Billaut, “Prise en compte des ressources multiples et des temps de préparation dans les problmes dordonnancement en temps rel. thse de doctorat,” Ph.D. dissertation, Université Paul Sabatier, 1993.

[13] C. Artigues, “Ordonnancement en temps réel d’ateliers avec temps de préparation des ressources,” Ph.D. dissertation, LAAS - Laboratoire d’analyse et d’architecture des systmes [Toulouse], 1997.

[14] C. Esswein, “Un apport de flexibilité séquentielle pour l’ordonnancement robuste,” Ph.D. dissertation, Université François Rabelais Tours(France), December 2003.

[15] G. Pinot, “Coopération homme-machine pour l’ordonnancement sous incertitudes,” Ph.D. dissertation, Université de Nantes (France), November 2008.

[16] H. T. La, “Utilisation d’ordres partiels pour la caractérisation de solutions robustes en ordonnancement,” Ph.D. dissertation, Institut National des Sciences Appliques de Toulouse (France), Janvier 2005.

[17] C. Artigues, J. Billaut, and C. Esswein, “Maximization of solution flexibility for robust shop scheduling,” *European Journal of Operational Research*, vol. 165, pp. 314–328, 2005.

[18] Z. Yahouni, N. Mebarki, and Z. Sari, “New lower bounds for the best-case schedule in groups of permutable operations.” Ottawa Canada: 15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control in Manufacturing, May 11-13 2015.

[19] P. Lopez and F. Roubellat, “Scheduling and constraint propagation,” *Production scheduling*, 2008.

[20] M. Alloulou and C. Artigues, “Worst-case evaluation of flexible solutions in disjunctive scheduling problems,” *Computational Science and Its Applications - ICCSA*

2007 International Conference, Proceedings, Part III, vol. 1205, pp. 1027–1036, August 26-29 2007.

[21] Z. S. Z. Yahouni, N. Mebarki, "A new approach for the best-case schedule in a group seqence." Nancy, France: 10 International Conference of Modeling and Simulation, Nov 5-7 2014.

[22] S. Lawrence, "Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement)," Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.

[23] B. S. P. Brucker, B. Jurisch, "A branch and bound algorithm for the job-shop scheduling problem," *Discrete Applied Mathematics*, vol. 49, pp. 107–127, 1994.

[24] K. Morikawa and K. Takahashi, "A flexible branch and bound method for the job shop scheduling problem," *Industrial Engineering and Management Systems*, vol. 8(4), pp. 239–246, Oct. 2009.