A new constructive heuristic for the Fm|block|∑T

Companys R[1],Ribas I[2]

Abstract This paper deals with the blocking flow shop problem and proposes new constructive procedures for the total tardiness minimization of jobs. The heuristic has three-phases to build the sequence; the first phase selects the first job to be scheduled, the second phase arranges the remaining jobs and the third phase uses the insertion procedure of NEH to improve the sequence. The proposed procedures evaluate the tardiness associated to the sequence obtained before and after the third phase in order to keep the best of both because the insertion phase can worsen the result. The computational evaluation of these procedures against the benchmark constructive procedures from the literature reveals their good performance.

Keywords: flow shop, blocking, tardiness.

1 Introduction

In a flow shop, there are *n* jobs that have to be processed in *m* machines. All jobs follow the same route in the machines. The processing time of job $i \in \{1,2,...,n\}$ on machine *j*, $j \in \{1,2,..., m\}$, is $p_{j,i} > 0$. In the traditional version of the problem, it is assumed that there are buffers of infinite capacity between consecutive machines, where jobs, after being processed by the previous machine, can wait until the subsequent machine is available. However, in many industrial systems this supposition cannot be made, since the capacity of buffers is zero, due to the char-

---

[1]Ramon Companys Pascual (✉ e-mail: ramón.companys@upc.edu)
Dpto. de Organización de Empresas. Escola Tècnica Superior d'Enginyers Industrials de Barcelona. Universitat Politècnica de Catalunya. BarcelonaTech. Avda. Diagonal, 647, ed. H, P.7 08028 Barcelona.

[2]Imma Ribas Vila (✉ e-mail: imma.ribas@upc.edu)
Dpto. de Organización de Empresas. Escola Tècnica Superior d'Enginyers Industrials de Barcelona Universitat Politècnica de Catalunya. BarcelonaTech. Avda. Diagonal, 647, ed. H, P.7 08028 Barcelona.

acteristics of the process (Hall and Sriskandarajah 1996). In this type of productive configuration, a machine can be blocked by the job already processed if the buffer is full or absent. Therefore, an accurate scheduling is necessary to avoid or to minimise the blocking time of machines. Some examples can be found in the production of concrete blocks where storage is not allowed in some stages of the manufacturing process (Grabowski and Pempera 2000); in the iron and steel industry (Gong, Tang, Duin 2010); in the treatment of industrial waste and the manufacture of metallic parts (Martinez and others 2006); or in a robotic cell, where a job may block a machine while waiting for the robot to pick it up and move it to the next stage (Sethi and others 1992).

The tardiness criterion has been less studied than the makespan or total flowtime criteria, despite the fact that scheduling according to this performance measure helps companies offer a high service level to their customers, which is essential for survival in the market. In particular, to the best of our knowledge, only Armentano and Ronconi (2000) and Ronconi and Henriques (2009) dealt with the blocking flow shop problem for total tardiness minimization. Armentano and Ronconi (2000) propose a Tabu Search procedure that uses the LBNEH method proposed in Armentano and Ronconi (1999) to obtain the initial solution. Alternatively, in Ronconi and Henriques (2009), a constructive method (FPDNEH) and a Greedy Adaptive Search Procedure (GRASP) is proposed for this problem.

In this paper we propose a new constructive heuristic which explores specific characteristics of the problem. The comparison between the most popular constructive heuristic used in the literature reveals its efficacy to find good solutions for the problem dealt with.

The paper is organized as follows: after this brief introduction, the problem is formally defined in section 2. Section 3 describes the proposed constructive heuristics, section 4 shows the computational evaluation and section 5 summarizes the conclusions.

2 Problem definition

The tardiness blocking flow shop problem denoted as $Fm|block|\sum T$ according to the notation proposed by Graham et al (1979) can be formulated with the following equations, where d_i denotes the due date of job i, $e_{j,k}$ the time in which the job in position [k] starts to be processed on machine j and $c_{j,k}$ is the departure time of this job:

$$e_{j,k} + p_{j,[k]} \leq c_{j,k} \quad j=1, 2, ..., m \quad k=1, 2, ..., n \tag{2.1}$$

$$e_{j,k} \geq c_{j,k-1} \quad j=1, 2, ..., m \quad k=1, 2 ,..., n \tag{2.2}$$

$$e_{j,k} \geq c_{j-1,k} \quad j=1, 2, ..., m \quad k=1, 2, ..., n \tag{2.3}$$

$$c_{j,k} \geq c_{j+1,k-1} \quad j=1, 2, ..., m \quad k=1, 2, ..., n \tag{2.4}$$

$$TT = \sum_{i=1}^{n} \max(c_{m,i} - d_i, 0) \tag{2.5}$$

$c_{j,0=0}$ $\forall j$, $c_{0,k}=0$, $c_{m+1,k}=0$ $\forall k$ are the initial conditions.

If equations (2.2) and (2.3) are summarized as (2.6) and equation (2.1) and (2.4) as (2.7), the schedule obtained is semi-active, which is interesting because an optimal solution can be found in the subset of the semi-active set of solutions.

$$e_{j,k}=\max\{c_{j,k-1}; c_{j-1,k}\} \tag{2.6}$$

$$c_{j,k} = \max\left\{e_{j,k} + p_{j,[k]}, c_{j+1,k-1}\right\} \tag{2.7}$$

3 Constructive heuristics for the Fm | block |∑T Problem

The most popular constructive heuristics used to deal with the tardiness criterion consist of using the *early due date* (EDD) rule or *slack* (sl) rule to create an initial sequence, which is then processed by the insertion phase of NEH (Nawaz, Enscore Jr, Ham 1983) adapted to the tardiness criterion. We denote to these procedures N_{EDD} and N_{sl}. In both procedures, the creation of the initial sequence is done by assigning an index to each job which is used for their prioritization. However, Ronconi and Henriques(2009) proposed a method, named FPD, which can be seen as a two-phase procedure; the first phase selects the job to be scheduled first and the second phase sequences the rest of the jobs. They also proposed to improve the obtained sequence by FPD with the insertion procedure. According to our previous notation, we named to this procedure N_{FPD}.

3.1 New constructive heuristics

The proposed procedures consist of three phases according to the philosophy of the N_{FPD} method. We have implemented four alternatives to choose the first job in the sequence (first phase) and three alternatives to sequence the remaining jobs (second phase). The combination of them has led us to implement twelve methods for creating a sequence which is then processed by the insertion procedure.

We denote as P_i, the sum of processing time of job i, and as sl_i the slack of job i calculated as the difference between its due date and the sum of its processing time. Next, we describe the rules used in the first phase:

- A1: Select the job with minimum $\lambda \cdot p_{1,i} + (1 - \lambda) \cdot d_i$, if λ=0.5 this index is exactly the same than the used in FPD.
- A2: Select the job with minimum $\lambda \cdot P_i$ / m + $(1 - \lambda) \cdot sl_i$, to break ties select the job with minimum d_i.
- A3: Select the job with minimum $\lambda \cdot \frac{P_i - P_{min}}{P_{max} - P_{min}} + (1-\lambda) \cdot \frac{sl_i - sl_{min}}{sl_{\max} - sl_{min}}$ where P_{min} and P_{max}, are the maximum and minimum processing time and sl_{min} and sl_{max} the maximum and minimum slack of jobs, respectively.
- A4: Select the job with minimum $\lambda \cdot \frac{dif_i \text{-} dif_{min}}{dif_{max} \text{-} dif_{min}} + (1-\lambda) \cdot \frac{sl_i \text{-} sl_{min}}{sl_{\max} \text{-} sl_{min}}$ where $dif_i = \sum_{j=1}^{m} | p_{j,i} - P_i / m |$, to break ties select the job with minimum d_i.

These procedures are variants of the used in FPD. A1 allows weighting the two factors, A2 considers the important of the total processing time instead of the processing time in the first stage and changes the due date for the slack of jobs, A3 is an evolution of A2 and A4 change the first factor in order to prioritise jobs with a more regular processing time.

The implemented rules for the second phase are:

- B1: the same rule than in FPD.
- B2: select the job with minimum $\mu \cdot \frac{tm_i - tm_{\min}}{tm_{max} - tm_{min}} + (1-\mu) \cdot \frac{dsl_i - dsl_{min}}{dsl_{\max} - dsl_{min}}$, in case of ties choose the job with early due date. tm_i is the idle time generated when job i is scheduled at the end on the partial sequence σ, in position k+1, $tm_i = \sum_{j=1}^{m} f_{j,k+1}(i) - f_{j,k}(\sigma) - p_{j,i})$ and dsl_i is the dynamic slack of job i when it is scheduled at the end of σ, $\mathrm{dsl_i = d_i - c_{m,i}}$
- B3: Select, among the jobs with $\frac{dsl_i \text{-} dsl_{min}}{dsl_{max} \text{-} dsl_{min}} < \mu$, the job with minimum tm_i.

As it can be observed, B2 changes the two terms of B1 in order to precise their evaluation. The first term of B1, fit_i, which is an approximation of the idle time, is changed by the real idle time resulting from scheduling job i in that position. The second term, dynamic slack of job i, is evaluated considering that this job is scheduled at the end of the partial sequence. B3, instead, uses hierarchical multicriteria decision to select a job.

Finally, the obtained sequence after these two phases is tried to improve by the insertion procedure, adapted to the tardiness criterion. But, as during this research we detected that, in some instances, the tardiness associated to the sequence given by the constructive procedures was smallest than the associated to the sequence obtained after the insertion, the implemented procedures evaluate the sequence before and after the insertion procedure in order to keep the best of both.

3.2 Experimental parameter adjustment of the rules

The proposed procedures have two parameters, λ and μ, that should be adjusted. The calibration of these parameters has been done with 480 instances generated *ad hoc*, 10 instances for each combination of n={20, 50, 100, 200} and m={5, 10, 20} and 4 ranges of due dates, which are named scenarios from now on. The due dates of jobs were uniformly distributed between LB·(1-T-R/2) and LB·(1-T+R/2) as in Potts and Van Wassenhove (1982), where T and R are the tardiness factor of jobs and the dispersion range of due dates, respectively. LB is a lower bound of the C_{max} with unlimited buffer in the flow shop (Taillard 1993) problem. Therefore, each of the scenarios correspond to a combination of R={0.6, 1.2} and T={ 0.2, 0.4}. The experiments were carried out on an Intel Core 2 Duo E8400 CPU, with 3GHz and 2GB RAM memory. To analyze the experimental results obtained, we measured the relative deviation index (*RDI*), calculated as (3.1) for each procedure:

$$RDI=\frac{Heur_{h,s}-Best_s}{Worst_s-Best_s} \tag{3.1}$$

where $Heur_{h,s}$ is the average of tardiness values obtained by heuristic h, in instance s, and $Best_s$ and $Worst_s$ are the minimum and maximum value of tardiness obtained for this instance, among all the combinations of parameters.
A preliminary test showed us that any Ai procedure (first phase) combined with B2 outperformed the procedures that used B1 or B3 in the second phase, which led us to discard B1 and B3. Hence, the adjustment of λ and μ was done for those procedures that use B2 in the second step, which have been denoted as N_{A1}, N_{A2}, N_{A3} and N_{A4}, to indicate the method used in the first step.

As an example, the calibration of N_{A1} is shown in Fig. 1, where the overall average of index RDI obtained, for several values of λ and μ can be seen. Notice, that in this case the lower values of RDI are obtained when λ=0.45 and μ=0.35.

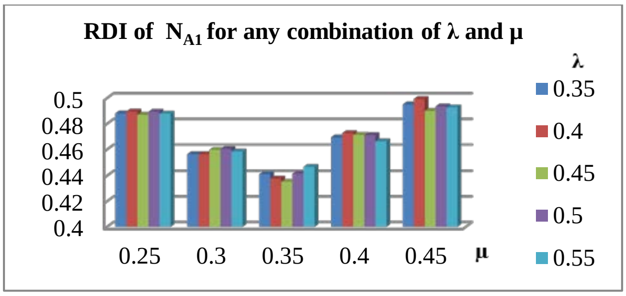

Fig. 1 Overall average of RDI obtained with N_{A1} procedure for any combination of λ and μ

The same experiment was done for each procedure and the parameters were fixed according to the values showed in Table 1.

Table 1 best values of λ and μ per each procedure

	N_{A1}	N_{A2}	N_{A3}	N_{A4}
λ	0.45	0.1	0.1	0.15
μ	0.35	0.35	0.30	0.35

4 Computational evaluation

In this section we compare the proposed procedures with N_{FPD}, N_{EDD} and N_{sl} in order to analyse their performance. This test was done against the 480 instances used in Ronconi and Henriques (2009). The comparison between procedures was done using the index RDI as in (3.1), where $Best_s$ and $Worst_s$ are the minimum and maximum value of the total tardiness obtained by any of the procedures evaluated in this test.

Table 2 shows the average RDI values obtained and the number of best solutions found by each procedure in each *n*x*m* set. The first observation is that the new procedures have better performance than N_{FPD}, N_{EDD} or N_{sl}. However, the difference between them is very poor. We can say that, N_{A3} performs slightly better, but its behaviour is worse for *n*=20 and *n*=200, which dilutes the overall average value for this procedure. This can also be seen by means the number of best solutions found by each procedure in each set. As the only difference between these procedures is the selection of the first job, we can say that this decision has a great influence in the obtained sequence. Therefore, one line of future research is to explore the convenience of using one of these procedures to select the first job according to the number of jobs to schedule.

Another interesting result is the performance of N_{EDD} and N_{sl} compared to N_{FPD}. The obtained results in this test are much better than those reported in Ronconi and Henriques (2009). This fact can be due to the tie-break criterion used to select the jobs. In N_{EDD}, we break ties by selecting the job with higher P_i whereas in N_{sl} the job selected is the one with less P_i. This observation indicates that can be interesting to analyse several criteria to break ties because it has an appreciable effect in the obtained results.

Table 2 Average RDI value obtained and number of best solutions found by each procedure per *n* and *m* combination

	Average RDI							Number of best solutions						
*n*x*m*	N_{A1}	N_{A2}	N_{A3}	N_{A4}	N_{EDD}	N_{sl}	N_{FPD}	N_{A1}	N_{A2}	N_{A3}	N_{A4}	N_{EDD}	N_{sl}	N_{FPD}
20x5	0.47	**0.31**	0.49	0.31	0.44	0.41	0.48	11	14	7	14	11	**16**	12
20x10	0.47	**0.43**	**0.43**	**0.43**	**0.43**	0.57	**0.43**	7	9	9	9	**15**	2	14
20x20	0.56	**0.41**	0.45	**0.41**	0.42	0.43	0.60	5	**12**	8	**12**	**12**	8	8
50x5	0.37	0.38	**0.33**	0.38	0.42	0.49	0.65	**11**	10	10	10	10	8	3
50x10	0.51	0.54	**0.41**	0.54	0.49	0.48	0.59	6	6	**12**	6	7	5	8
50x20	0.51	0.54	0.47	0.53	0.46	**0.42**	0.56	2	4	10	4	5	**14**	6
100x5	0.29	0.27	**0.22**	0.27	0.34	0.30	0.74	10	9	11	9	11	**12**	6
100x10	0.33	**0.32**	0.37	**0.32**	0.34	0.33	0.68	**13**	8	4	8	6	4	11
100x20	0.44	0.46	**0.42**	0.46	0.45	0.46	0.56	3	6	6	6	7	6	**13**
200 x10	0.33	0.33	0.33	**0.33**	0.47	0.53	0.79	**14**	11	9	12	9	7	0
200x20	**0.32**	0.37	0.44	0.38	0.42	0.54	0.68	10	**12**	8	11	8	1	4
50 x20	**0.27**	0.42	0.34	0.42	0.51	0.50	0.73	**18**	14	7	13	3	5	2
All	0.41	0.40	**0.39**	0.40	0.43	0.45	0.62	110	**115**	101	114	104	88	87

5 Conclusions

In this paper we have proposed effective constructive heuristics procedures to deal with the total tardiness flow shop problem with blocking. The presented procedures have three steps; step one selects the first job in the sequence, step two builds the remaining sequence and step three uses the insertion procedure of NEH, adapted to the tardiness criterion, to try to improve the sequence. It has been implemented four procedures for step one and three for step two, which have been combined to build twelve heuristics. The best method for step two was B2 which is focused on the precise evaluation of the idle time and slack time of jobs; instead, the best procedure for the first step is not clear because it seems to be dependent of the number of jobs to schedule. The computational evaluation has reveal that the tie-break criterion used to select a job in the constructive procedures has a great influence in the quality of the obtained sequence. Therefore, we recommend to the researchers a detailed analysis of the criterion used to break ties.

6 References

Armentano VA, Ronconi DP (1999) Tabu search for total tardiness minimization in flowshop scheduling problems. Comput Oper Res 26(3):219-35

Armentano VA, Ronconi DP (2000) Minimização do tempo total de atraso no problema de flowshop com buffer zero através de busca tabu. Gestao & Produçao 7(3):352

Gong H, Tang L, Duin CW (2010) A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times. Comput Oper Res 37(5):960-9

Grabowski J, Pempera J (2000) Sequencing of jobs in some production system. European Journal of Operational Research 125(3):535-50

Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation In: deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics 5:287-326

Hall NG, Sriskandarajah C (1996) A survey of machine scheduling problems with blocking and no wait in process. Operations Research 44(3):510-25

Martinez S, Dauzère-Pérès S, Guéret C, Mati Y, Sauer N (2006) Complexity of flowshop scheduling problems with a new blocking constraint. Eur J Oper Res 169(3):855-64

Nawaz M, Enscore Jr EE, Ham I (1983) A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. Omega 11(1):91-5

Potts CN, Van Wassenhove LN (1982) A decomposition algorithm for the single machine total tardiness problem. Operations Research Letters, 1(5):177-81

Ronconi DP, Henriques LRS (2009) Some heuristic algorithms for total tardiness minimization in a flowshop with blocking. Omega 37(2):272-81

Sethi SP, Sriskandarajah C, Sorger G, Blazewicz J, Kubiak W (1992) Sequencing of parts and robot moves in a robotic cell. International Journal of Flexible Manufacturing Systems 4:331-58

Taillard E (1993) Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2):278-85.