An acceleration procedure for several objective functions in the permutation flow shop scheduling problem

Victor Fernandez-Viagas[1], Jose M. Molina-Pariente[1], Carla Talens[1] and Jose M. Framinan[1]

Industrial Management, School of Engineering, University of Seville, Spain
vfernandezviagas,jmolina1,cartafa,framinan@us.es

Keywords: Scheduling, Flowshop, flow shop, accelerations, speed-up, total completion time, total tardiness, total earliness, just-in-time,critical path.

1 Introduction

In a flowshop layout, n jobs have to be processed on m machines following all jobs the same route of machines. The flowshop scheduling problem involves the search of the best order to process the jobs in each machine. Traditionally, a common simplification, denoted as Permutation Flowshop Scheduling Problem (PFSP), is adopted to avoid an extensive use of manpower or machines in the shop, where the order of jobs does not change between machines. This particular problem is one of the most studied optimization problems in Operations Research (Fernandez-Viagas *et. al.* 2017), probably due to the following reasons: this flowshop layout is very common in real manufacturing scenarios (Vakharia and Wemmerlov 1990); many job shops can be simplified to a reduced flow shop under several constraints (Storer *et. al.* 1992); and many models and solution procedures for different constraints and layouts have their origins in the flowshop scheduling problem. The PFSP is denoted by $Fm|prmu|\gamma$, where γ is the goal to be solved. Due to the NP-hard nature of the problem, many approximate algorithms have been proposed in the literature for the traditional problem and/or related constrained PFSP. Without any doubt, one of the key factor for the efficiency of these approaches is the use of methods to accelerate the calculation of the objective functions or local search methods. In this regards, Taillard (1990) proposed the first speed-up procedure for $Fm|prmu|C_{max}$ (denoted as Taillard's accelerations) in the literature. These accelerations have been incorporated in hundreds of papers and its use is nowadays mandatory to obtain an efficient approximate algorithm for the $Fm|prmu|C_{max}$ problem. Unfortunately, they can be only applied in a very few amount of objectives and/or constraints, and the search for more efficient accelerations is still open in the literature. Thereby, in the race for finding accelerations for approximate algorithms in other related problems, different speed-up procedures have been proposed in the literature taking into account their specific problem properties. In this paper, we propose a new speed-up procedure for $Fm|prmu|\sum C_j$, $Fm|prmu|\sum T_j$, and $Fm|prmu|\sum E_j + \sum T_j$, using specific properties based on the critical path, which clearly outperforms previous proposals.

2 Literature review

A number of speed-up procedures have been proposed to accelerate approximate algorithms in the related flowshop problems. To the best of our knowlegde, the well-known accelerations proposed by Taillard (1990) was the first proposal published in the literature for flowshop layouts. Using the specific properties of the $Fm|prmu|C_{max}$ problem, the author develops a very efficient procedure to accelerate the evaluation of approximated algorithms in this problem. More specifically, using this procedure the complexity of insertion local search methods can be reduced from $n^3 \cdot m$ to $n^2 \cdot m$. Due to the repercussion and excellent results of this paper, several other authors have been adapted them to related PFSP. Thereby, Mercado and Bard (1998) adapt the Taillard's accelerations for the PFSP with sequence-dependent and anticipatory setup times and makespan minimisation, i.e. $Fm|s_{ijk}, prmu|C_{max}$. Naderi and Ruiz (2010) adapted them for the distributed permutation flowshop scheduling problem to minimise makespan, $DF|prmu|C_{max}$. For the blocking PFSP and makespan minimisation ($Fm|blocking|C_{max}$), they have been first proposed by Wang *et. al.* (2010). For the no-idle case (idle times no allowed on machines), they are adapted by Fatih Tasgetiren *et. al.* (2013) to minimise the makespan and by Pan and Ruiz (2010) for the mixed no-idle permutation flowshop scheduling problem with makespan minimisation. In the non-permutation case also for makespan, they are extended by Benavides and Ritt (2016). In the PFSP with time lags constraints, Wang *et. al.* (2018) adapt the Taillard's accelerations to minimise the makespan. Regarding other objective functions, these accelerations have been only adapted in some very particular cases. For the no-idle case, they have successfully adapted by Fatih Tasgetiren *et. al.* (2013) for total completion time ($Fm|prmu, no-idle|\sum C_j$) and total tardiness ($Fm|prmu, no-idle|\sum T_j$), respectively. Fernandez-Viagas and Framinan (2015) adapted the accelerations to a different objective function, the $Fm|prmu|\epsilon(C_{max}/T_{max})$ problem, but allowing some infeasible solutions that should be repaired. They adapt the accelerations for $Fm|prmu|\epsilon(C_{max}/T_{max})$. In addition, they incorporated a bounded local search based on problem properties, also to reduce the number of positions where the jobs are inserted.

Other accelerations have been also proposed in the PFSP literature, when the completion time of each job is needed to evaluate the objective function. Thereby, Framinan and Leisten (2008) propose accelerations (denoted as FL's accelerations) to reduce the evaluation of insertion local search in the PFSP with minimisation of total tardiness ($Fm|prmu|\sum T_j$). Note that these accelerations can be easily extended to interchange local search, or other related PFSP as e.g.: $Fm|prmu|\sum C_j$, $Fm|prmu|\sum E_j + \sum T_j$, and $DF|prmu|\sum C_j$. Finally, some methodologies -reducing the number of moves in local searches- have also been proposed in the literature, denoted as bounded local searches. Basically, they reduce these moves by evaluating some lower or upper bounds and comparing them against the so-far best solution found by the algorithm. This is the case of the aforementioned bounded local search proposed by Fernandez-Viagas and Framinan (2015) in $Fm|prmu|\epsilon(C_{max}/T_{max})$ which bound the feasible positions where insert the jobs. Fernandez-Viagas and Framinan (2014) also use such as a procedure avoiding the insertion or interchange of jobs in factories for the distributed PFSP ($DF|prmu|C_{max}$). Finally, a lower bound is used in Pagnozzi and Stützle (2017) to discard the evaluation of some insertions in the PFSP with weighted total tardiness ($Fm|prmu|\sum w_j T_j$).

3 New speed up procedure

In this section, we propose a new speed up procedure, denoted as FMF accelerations, for insertion-based local search methods in the PFSP. The procedure highly reduces the CPU time of this type of local search methods by avoiding the calculation of a number of operations. Basically, it is based in the fact that, when inserting job σ in position j, only the completion times of the jobs between the critical path and job π_{j+1} need to be calculated to obtain any objective function in the $Fm|prmu|-$ problem because: jobs over the critical path do not influence in the completion times on the critical path; and the completion times of jobs before $j+1$ are known from previous iteration. In Figure 1, we present an example of a sequence composed of four jobs (1,2,4,5) where a new job 3 wants to be inserted in position 3. The critical path is shown in solid gray and the only operations whose completion times must be evaluated are with diagonal green lines. As a consequence only the completion times of the jobs between the critical path and job π_{j-1} need to be calculated to obtain any objective function in the $Fm|prmu|-$ problem.

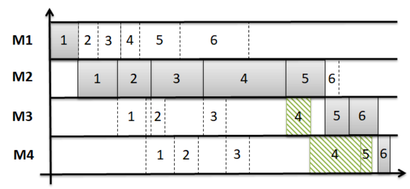

Fig. 1. Example of the new speed-up

With this in mind, the detailed procedure of the proposed accelerations can be explained as follows. Firstly, we calculate e_{ij}, q_{ij}, f_{ij} for sequence Π (i.e. without job σ) according to the following equations:

$$e_{ij} = max\{e_{i,j-1}, e_{i-1,j}\} + p_{i\pi_j}, i = 1 \dots m, j = 1 \dots k-1 \quad (1)$$

$$q_{ij} = max\{q_{i+1,j}, q_{i,j+1}\} + p_{i\pi_j}, i = m \dots 1, j = k-1 \dots 1 \quad (2)$$

$$f_{ij} = \max\{e_{i,j-1}, f_{i-1,j}\} + p_{i\sigma}, i = 1 \dots m, j = 1 \dots k \quad (3)$$

In addition we introduce cp_{ij} as a variable to reproduce the critical path after the insertion of the new job. cp_{ij} is then equals to 1 if there is no idle time between the operation O_{i,π_j} and $O_{i,\pi_{j+1}}$, and 0 otherwise (i.e. when there is no idle time between O_{i,π_j} and O_{i+1,π_j}). Next, job σ is tested in each position j of sequence Π. In each of these positions, we determine the machine $i^{'}$ where the forward and backward critical paths join, i.e. $i^{'} = max_{i=1 \dots m}\{f_{ij} + q_{ij}\}$. Then, we calculate the value of the objective function for all previous insertion. There, the variable cp_{ij} is used to reproduced the critical path. If $cp_{ij} = 1$, the completion time of job in position $j+2$ (which corresponds to job pi_{j+1}) is the actual load of machine i (denoted $Load_i$) plus the processing time $p_{i\pi_{j+1}}$ and the completion time of this job in the other machines is updated. Otherwise, the critical path moves to a higher machine and the completion time of job in position $j+1$ can directly be obtained adding the processing time to the load of previous machine.

4 Computational Results and Conclusions

In this section, we compare the proposed FMF accelerations against previous accelerations proposed in the literature. Three different experimentations have been carried out by implementing the proposed accelerations in the following three different objective functions: Experimentation #1: Total completion time ($Fm|prmu|\sum C_j$); Experimentation #2: Total tardiness ($Fm|prmu|\sum T_j$); and Experimentation #3: Total earliness and tardiness ($Fm|prmu|\sum E_j + \sum T_j$).

The computational results have been developed on two well-known sets of extensive instances with and without due dates, respectively. Results show the excellent performance of the proposed accelerations, regardless the objective function and the benchmark. More specifically, the proposed accelerations clearly outperform each other accelerations proposed in the literature so far. Thereby, in the minimisation of total earliness and tardiness, the

CPU times are reduced in average 75.5% against the method without accelerations and 42.9% against the best so far accelerations found in the literature. For some instances the reduction is around a 90% of the computational time (as compared not using accelerations). In the total tardiness case, the average reduction of computational time is 63.5% against the 44.6% and 45.9% of the PS's and FL's accelerations respectively. Similarly, the reduction in the total completion time is 50.8% against the 37.7% found by the FL's accelerations.

Acknowledgements

This research has been funded by the Spanish Ministry of Science and Innovation, under the project "PROMISE" with reference DPI2016-80750-P.

References

Taillard, E., 1990, "Some efficient heuristic methods for the flow shop sequencing problem", *European Journal of Operational Research*, Vol. 47(1), pp. 65-74.

Vakharia A.J., Wemmerlov U., 1990, "Designing a cellular manufacturing system: a materials flow approach based on operation sequences", *IIE Transactions*, Vol. 22.

Rios-Mercado R.Z. and Bard J.F., 1998, "Heuristics for the flow line problem with setup costs", *European Journal of Operational Research*, Vol. 110, pp. 76-98.

Benavides A.J. and Ritt M., 2016, "Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops", *Computers and Operations Research*, Vol. 66, pp. 160-169.

Naderi B. and Ruiz R., 2010, "The distributed permutation flowshop scheduling problem", *Computers & Operations Research*, Vol. 37, pp. 754-768.

Framinan J.M. and Leisten R., 2008, "Total tardiness minimization in permutation flow shops: A simple approach based on a variable greedy algorithm", *International Journal of Production Research*, Vol. 46, pp. 6479-6498.

Pan Q.K. and Ruiz R., 2010, "An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem", *Omega*, Vol. 44, pp. 41-50.

Fernandez-Viagas V. and Framinan J.M., 2015, "Efficient non-population-based algorithms for the permutation flowshop scheduling problem with makespan minimisation subject to a maximum tardiness", *Computers & Operations Research*, Vol. 64, pp. 86-96.

Fernandez-Viagas V. and Framinan J.M., 2014, "A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem", *International Journal of Production Research*, Vol. 53, pp. 1111-1123.

Pagnozzi F. and Stützle T., 2017, 'Speeding up local search for the insert neighborhood in the weighted tardiness permutation flowshop problem", *Optimization Letters*, Vol. 11(7), pp. 1283-1292.

Storer R.H., Wu S.D. and Vaccari R., 1992, "New search spaces for sequencing problems with application to job shop scheduling", *Management Science*, Vol. 38, pp. 1495-1509.

Fernandez-Viagas V., Ruiz R. and Framinan J.M., 2017, "A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation", *European Journal of Operational Research*, Vol. 257(3), pp.707-721.

Wang L. and Pan Q.-K. and Suganthan P.N. and Wang W.H. and Wang Y.M., 2010, "A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems", *Computers and Operations Research*, Vol. 37(3), pp.509-520.

Fatih Tasgetiren M. and Pan Q.K. and Suganthan P.N. and Buyukdagli O., 2013, "A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem", *Computers and Operations Research*, Vol. 40(7), pp.1729-1743.

Wang B. and Huang K. and Li T., 2018, "Permutation flowshop scheduling with time lag constraints and makespan criterion", *Computers Industrial Engineering*, Vol. 120, pp. 1-14.