# Multi-objective sequence dependent setup times flowshop scheduling: a new algorithm and a comprehensive study

Michele Ciavotta, Rubén Ruiz, Gerardo Minella

Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia, Valencia,
`mciavotta,mgerar@iti.upv.es,rruiz@eio.upv.es`

Although many papers have coped with the flowshop scheduling problem with setups, according to our knowledge, little has been published dealing with the optimization of more than one objective at a time in the presence of sequence dependent setup times.

In this work we tackle the permutation flowshop with sequence dependent setup times considering two well known independent objectives, the Maximum Completion Time, also called *Makespan* ($C_{max}$), and the *Total Weighted Tardiness* ($TWT$). Moreover, the presence of sequence dependent setup times ($SDST$) delays significantly the completion times $c_{j,k}$ of each job $j$ on each machine $k$ and hence the considered objectives.

An efficient algorithm, IPG (*Iterated Pareto Greedy*), has been developed to face this complex scheduling setting. It basically consists of a greedy strategy iteratively applied on an archive of nondominated solutions. The greedy procedure used is a Pareto evolution of the well known NEH heuristic [1] making use of the Pareto relationship to generate, not only one, but a whole set of solutions that do not dominate each other. The IPG can be also seen as a Pareto evolution of the IG (*Iterated Greedy*) algorithm which is a rather new metaheuristic approach which has shown state-of-the-art performance in single objective optimization for the permutation flowshop problem with [2] and without setups [3]. The rationale of the proposed IPG method is very simple. Roughly, it is possible to divide it into four phases. The first phase consists of the *Initialization* where an initial set of good solutions is generated using an heuristic approach. The remaining three phases are iteratively repeated and constitute the bulk of the algorithm. They are: the *Selection* phase where one or more solutions belonging to the current archive of solutions are selected for the following phases. A modified version of the *Crowding Distance Assignment* procedure, originally presented in [4], has been developed in order to carry out the selection process. The *Pareto greedy improvement* phase is then applied on the selected solution and it returns a whole set of solutions which do not dominate each other. During this step, the current solution is disrupted (*Destruction*) removing some jobs from the sequence and a greedy procedure (*Construction*) is applied. The construction phase reinserts the eliminated jobs into partial sequences, similarly to the insertion procedure in the NEH heuristic, returning a hopefully improved nondominated solution set. This set is then joined to the current Pareto archive and dominated elements are discarded. Ultimately, a *Local search* phase is applied on a selected solution to improve the current nondominated set in terms of spread and diversity.

In this work we compare our algorithm against the highest performing approaches (especially those proposed for the multi-objective flowshop problem). In [5], a large number of papers dealing with multi-objective flowshop algorithms has been reviewed and the best performing ones among them have been also re-implemented. Here we first reduced the number of algorithms by means of a preliminary test on a reduced set of instances and, depending on the attained results, we selected the best ten algorithms. Note that such algorithms present also the best results for multi-objective permutation flowshop without setups. Seven of them are specifically designed to tackle flowshop problems while the remaining three are generic multi objective optimization procedures. During the preliminary phase we noted that the algorithm proposed by [6] (MOSA_Varad.) achieved worse positions with respect to the others because it is not able to entirely exploit the available time. Therefore, we decided to implement an improved version of that procedure (which we refer to as MOSA_Varad_M).

In all the experiments presented in this work, we make use of two different sequence dependent setups instance sets based on the original instances of [7] and those presented in [8] and used for the first time in [9]. Each set contains 110 instances with several combinations of the number of jobs $n$ and number of machines $m$. The $n \times m$ combinations are: $\{20, 50, 100\} \times \{5, 10, 20\}$, $200 \times \{10, 20\}$. Setup times are selected to be respectively 50% and 125% of the processing times ($p_{ij}$). As regards the performance measures, the comparison of two different Pareto approximations coming from two algorithms is not straightforward. Two approximation sets $A$ and $B$ can even be incomparable. Recent studies like those of [10], [11] or more recently, [12] are an example of the enormous effort being carried out in order to provide the necessary tools for a better evaluation and comparison of multi-objective algorithms. In particular, measures so-called "Pareto-compliant", see [13; 10], seem to be the most appropriate. Among these, we selected the hypervolume ($I_H$) and the multiplicative unary Epsilon ($I_\varepsilon^1$) indicators which represent the state-of-the-art as far as quality indicators are concerned.

The stopping criterion for all algorithms is given by a time limit depending on the size of the considered instance. The algorithms are stopped after a CPU running time of $n \cdot m/2 \cdot t$ milliseconds, where $t$ is an input parameter. In this way we assign more time to larger instances that are obviously more time consuming. Every algorithm is run 10 different independent times (replicates) on each instance with two different stopping criteria: $t = 150$ and $t = 200$ milliseconds. For every instance, stopping time and replicate we used the same random seed as a common variance reduction technique. A total of 52,800 data points are collected if we consider the 12 algorithms, 220 instances, 10 replicates per instance and two different stopping time criteria. In reality, each data point is an approximated Pareto front containing a set of vectors with the two objective values.

From the $12 \cdot 10 = 120$ available Pareto front approximations for each instance, the best non-dominated Pareto front is found and stored. Additionally, a set of best Pareto fronts is stored for each one of the two employed stopping times. These last Pareto fronts are also used for obtaining the reference points for the hypervolume ($I_H$) indicator and used as the reference set in the multiplicative unary Epsilon indicator ($I_\varepsilon^1$).

Table 1 contains average results for the first instance set. Although each depicted value is attained by means of a very large number of data points, it is still necessary to carry out a comprehensive statistical experiment to assess if the observed differences in the average values are statistically significant. A total of 16 different experiments are carried out. We did parametric ANOVA analyses as well as non-parametric Friedman rank-based tests on both quality indicators and for the two different stopping criteria. The utility of showing both parametric as well as non-parametric tests consists in improving the soundness of our conclusions. We carried out eight multi-factor ANOVAS where the type of instance is a controlled factor. The algorithm is another controlled factor with 12 levels. The response variable on each experiment is either the hypervolume or the epsilon indicator. Lastly, there is one set of experiments for each stopping time. All the tests are carried out with confidence level $\alpha = 0.05$. Considering that each experiment contains 13,200 data points, the three main hypotheses of ANOVA: normality, homoscedasticity and independence of the residuals are easily satisfied. To compare results, a second set of eight experiments are performed. In this case, non-parametric Friedman rank-based tests are carried out. Since there are 12 algorithms and 10 different replicates, the results for each instance are ranked between 1 and 120. We performed four different statistical tests on each set of results.

All these tests proved that IPG widely outperforms all other algorithms for both hypervolume ($I_H$) and epsilon indicators ($I_\varepsilon^1$). As a consequence, IPG can be considered the state-of-art for this important scheduling problem.

Table 1: Average quality indicator values for two different termination criteria. Instance group where setup time is 50% that of the processing time.

| | Time | 150 | | | 200 | |
|---|---|---|---|---|---|---|
| # | Method | $I_H$ | $I_\varepsilon^1$ | Method | $I_H$ | $I_\varepsilon^1$ |
| 1 | IPG | 1.197 | 1.119 | IPG | 1.190 | 1.124 |
| 2 | MOGALS_Arroyo | 1.102 | 1.185 | MOGALS_Arroyo | 1.095 | 1.185 |
| 3 | PESAII | 1.071 | 1.187 | PESAII | 1.064 | 1.191 |
| 4 | PESA | 1.066 | 1.198 | PESA | 1.057 | 1.203 |
| 5 | MOSA_Varad_M | 1.019 | 1.308 | MOSA_Varad_M | 1.008 | 1.313 |
| 6 | MOTS | 1.013 | **1.255** | MOTS | 1.000 | **1.263** |
| 7 | PGA_ALS | 0.994 | **1.230** | PGA_ALS | 0.975 | **1.238** |
| 8 | MOSA_Varad | 0.937 | 1.361 | MOSA_Varad | 0.910 | 1.377 |
| 9 | MOGA_Murata | 0.879 | **1.339** | MOGA_Murata | 0.872 | **1.344** |
| 10 | PILS | 0.840 | 1.390 | PILS | 0.868 | 1.375 |
| 11 | $\varepsilon$−NSGAII | 0.818 | **1.349** | CMOGA | 0.814 | **1.357** |
| 12 | CMOGA | 0.814 | 1.355 | $\varepsilon$−NSGAII | 0.814 | **1.349** |

# Bibliography


[1] Nawaz, M., Enscore, J.E.E., Ham, I.: A heuristic algorithm for the $m$ machine, $n$ job flowshop sequencing problem. Omega-International Journal of Management Science **11**(1) (1983) 91–95

[2] Ruiz, R., Stützle, T.: An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. European Journal of Operational Research **87**(3) (2008) 1143–1159

[3] Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research **177**(3) (2007) 2033–2049

[4] Deb, K.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2) (2002) 182–197

[5] Minella, G., Ruiz, R., Ciavotta, M.: A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. INFORMS Journal on Computing **20**(3) (2008) 451–471

[6] Varadharajan, T., Rajendran, C.: A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. European Journal of Operational Research **167**(3) (2005) 772–795

[7] Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research **64**(2) (1993) 278–285

[8] Vallada, E., Ruiz, R., Maroto, C.: Synthetic and real benchmark for complex flow-shop problems. Technical report, Universidad Politecnica de Valencia, Spain (2003)

[9] Ruiz, R., Maroto, C., Alcaraz, J.: Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. European Journal of Operational Research **165**(1) (2005) 34–54

[10] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE, Transactions on Evolutionary Computation **7**(2) (2003) 117–132

[11] Paquete, L.F.: Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: Method and Analysis. PhD thesis, Computer Science Department. Darmstadt University of Technology. Darmstadt, Germany (2005)

[12] Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland (February 2006) revised version.

[13] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation **3**(4) (1999) 257–271