New results for the multi-objective sequence dependent setup times flowshop problem

Michele Ciavotta · Rubén Ruiz · Gerardo Minella

Although many papers deal with the permutation flowshop scheduling problem with or without setups, according to our knowledge, little has been published tackling multi-objective optimization in presence of sequence dependent setup times. Hence, in this work we cope with this problem considering two pairs of well known independent objectives, the $C_{max} - TWT$ (*Makespan-Total Weighted Tardiness*) and $C_{max} - TFT$ (*Makespan-Total Flowtime*). An effective algorithm, RIPG (*Restarted Iterated Pareto Greedy*), has been developed to face this complex scheduling setting. The RIPG is a Pareto evolution of the IG (*Iterated Greedy*) algorithm, a rather new metaheuristic approach which has shown state-of-the-art performance in single objective optimization for the permutation flowshop problem with [1] and without setups [2]. In essence, it consists of a greedy strategy iteratively applied over an archive of nondominated solutions. The greedy procedure used is an evolution of the well known NEH heuristic [3] and makes use of the Pareto relationship to generate a whole set of nondominated solutions. The rationale of the proposed method is very simple. Roughly, it is possible to divide it into five phases. The first phase is the *Initialization*, where an initial set of good solutions is generated using a heuristic approach. The remaining four 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, are selected for the following steps. 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 over the selected solution and it returns a 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 procedure reinserts the eliminated jobs into partial sequences similarly to the insertion procedure of the NEH heuristic, returning, a hopefully improved, nondominated solution set. This set is then added to the current Pareto archive and dominated elements are discarded. A *Local search* phase is hence applied on a selected solution to enrich the process and to improve the current Pareto set in terms of spread and diversity. Lastly, a *Restart*

Michele Ciavotta, Rubén Ruiz, Gerardo Minella
Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Valencia,
E-mail: {mciavotta,mgerar}@iti.upv.es, rruiz@eio.upv.es

procedure is implemented to prevent the algorithm from getting stuck in local optima. 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 have been reviewed and the best performing ones have been also implemented. Starting from this point, we first reduced the number of methods by means of a preliminary test on a small set of instances and, depending on the results obtained, we selected the ten best algorithms. Notice that such methods also show the best performance for multi-objective permutation flowshop without setups. During this preliminary phase we noted that the algorithm proposed by [6] (MOSA_Varad) achieved worse results with respect to the others, because it is not able to fully exploit the available CPU time. Therefore, we decided to implement an improved version of this method (which we refer to as MOSA_Varad_M). In [7] a promising algorithm (MOIGS) has been recently proposed and we decided to implement and evaluate.

In all the experiments that we present in this work, we make use of two different instance sets (SDST50 and SDST125), presented in [2]. 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 is not straightforward. However, recent studies point out how the so-called "Pareto-compliant" measures (see [8,9]) seem to be the most appropriate. Among these, we selected the hypervolume (I_H) and the multiplicative unary epsilon (I^1_ε) indicators which represent the state-of-the-art as far as quality indicators are concerned (for more details on the application of these measures to multi-objective flowshop problem see [5]).

The stopping criterion for all algorithms is given by a time limit that is not fixed but depends 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 can 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. A total of 114,400 data points are collected (13 tested algorithms $\times$ 220 instances $\times$ 10 replicates per instance $\times$ 2 different stopping time criteria $\times$ 2 pairs of objectives). Actually, each data point is an approximated Pareto front containing a set of vectors with the two objective values.

Table 1 contains average results for (SDST125) instance set and ($C_{max} - TWT$) as objectives. Although each depicted value is attained by means of a very large number of samples, 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 32 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 using both parametric and non-parametric tests consists in improving the soundness of our conclusions. We carried out 16 (8 for each couple of objectives) multi-factor ANOVAs where the type of instance is a controlled factor. The algorithm is another controlled factor with 13 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 14,300 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 16 experiments are performed. In this case, non-parametric Friedman rank-based tests are carried out. Since there are 13 algorithms and 10 different replicates, the results for each instance are ranked between 1 and 130. All these tests proved that IPG widely outperforms all other algorithms for both hypervolume (I_H) and epsilon indicators (I_ε^1). As a consequence, IPG can be considered the state-of-art for this important scheduling problem.

SDST125	Time	150			200	
#	Method	I_H	I_ε^1	Method	I_H	I_ε^1
1	RIPG	1.307	1.077	RIPG	1.322	1.067
2	MOIGS	1.194	1.171	MOIGS	1.207	1.160
3	MOGALS_Arroyo	0.980	1.271	MOGALS_Arroyo	0.993	1.260
4	MOSA_Varad_M	0.939	1.348	MOSA_Varad_M	0.949	1.344
5	MOTS	0.930	1.318	MOTS	0.938	1.312
6	PESAII	0.923	1.267	PESAII	0.936	1.262
7	PESA	0.922	1.264	PESA	0.933	1.259
8	MOSA_Varad	0.853	1.404	MOSA_Varad	0.847	1.408
9	PGA_ALS	0.840	1.354	PGA_ALS	0.842	1.352
10	PILS	0.776	1.410	PILS	0.81	1.385
11	MOGA_Murata	0.737	1.389	MOGA_Murata	0.747	1.384
12	$\varepsilon-$NSGAII	0.679	1.419	$\varepsilon-$NSGAII	0.688	1.416
13	CMOGA	0.669	1.422	CMOGA	0.686	1.413

Table 1 Results for $C_{max} - TWT$ criteria and SDST125 instance set.

References

1. 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), 1143–1159 (2008)
2. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research **177**(3), 2033–2049 (2007)
3. 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), 91–95 (1983)
4. Deb, K.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
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), 451–471 (2008)
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), 772–795 (2005)
7. Framinan, J., Leisten, R.: A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. OR Spectrum **30**, 787–804(18) (2008)
8. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation **3**(4), 257–271 (1999)
9. 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), 117–132 (2003)