Non-dominated sorting genetic algorithm for a bi-objective flexible flow shop problem. A Case Study.

Ibeth Grattz Rodríguez[1], **Jose-Fernando Jimenez**[1], **Eliana María González-Neira**[1], **Nicolás Eduardo Puerto Ordóñez**[2], **Yenny Alexandra Paredes Astudillo**[3], **Juan Pablo Caballero-Villalobos**[1]

[1]Industrial Engineering Department, Pontificia Universidad Javeriana, Colombia
E-mail: grattz.i, j-jimenez, eliana.gonzalez, juan.caballero@javeriana.edu.co

[2]Industrial Engineering Department, Universidad de los Andes, Colombia
E-mail: ne.puerto10@uniandes.edu.co

[3]Institucion Universitaria Politecnico Grancolombiano, Colombia
E-mail: yennyparedes@javeriana.edu.co

Keywords: Scheduling, flexible flow shop, genetic algorithm, non-dominated sorting genetic algorithm, total weighted tardiness, total setup cost.

1. Introduction

Production scheduling is one of the most complex tasks for manufacturing industries. It demands to allocate several productions orders or jobs within the machines aiming to optimize a set of KPIs, such as operating, financial and/or sustainable indicators, among many others. Certainly, the scheduling of this jobs could focus in fulfilling a single objective. However, for the last decades, industries have explored the inclusion of multiple objectives for balancing the best possible overall solution. This paper proposes Non-dominated genetic algorithm, for solving a flexible flow shop problem (FFSP) that minimized the total weighted tardiness and total setup cost.

For solving this problem, this paper focuses in the developing a metaheuristic for a case study regarding the production process of a Colombian Soap Factory. The current research considers the following factors: the total resources available, features related to the setup times of the machines, processing times of the products and the due-date requirements of each customer order. Additionally, it is included an extra factor related with the costs that can be reduced as a consequence of the sequencing of jobs.

The manufacturing environment of the Soap Factory is a FFSP with a set of 19 stages (S), each with a single machine but one that contains two unrelated parallel machines. All jobs have the same processing route starting in stage 1 and finishing in stage 19. Nevertheless, due to customizations, some jobs are allowed to skip some stages, depending on its characteristics. Once each product completes the required operation on each stage, it must be added to the queue of the next stage, where it must wait in a buffer of unlimited capacity to be processed. Finally, there are sequence dependent setup times and the processing times are fixed (including the transportation times between stages).

FFSPs have been studied by multiple authors. Yu et al. (2018) studied a FFSP in which there are unrelated parallel machines. Every machine has special characteristics that allow the processing of certain types of products. In order to find a solution of this problem, the use of a genetic algorithm is proposed, in which the effect of three different mutation operators is studied to increase diversification in every iteration. The objective of this study was to minimize the total tardiness. On the other hand, Rabiee et al. (2014) and Ahonen and de Alvarenga (2017) attempt to minimize the makespan in a FFSP. The first authors implemented a hybrid algorithm using an imperialist competitive algorithm, a simulated annealing, a variable neighborhood search and a genetic algorithm; while the second authors performed a comparison between a simulated annealing and a tabu search to find the solution to the stated problem.

Regarding the objective function, it can be said that a variety of studies focus on a single objective analysis; even though, real implications of scheduling problems generally involve more than one objective (Torkashvand et al., 2017). The above, due to the diversity of jobs that are found in the real industry, the processing requirements of each product, as well as the flexibility in terms of delivery to each customer. Consequently, Lassig et al. (2017) minimized both the tardiness and earliness by applying a multiobjective genetic algorithm, in an FFSP. Alternatively, Talbi (2009) showed that algorithms such as SPEA2 and NSGA2 are proper to approximate the pareto optimal set solutions with an acceptable computational complexity and are relatively simple in terms of its implementation.

Finally, in terms of the inclusion of cost analysis in scheduling problems, Yu and Seif (2016) established a genetic algorithm to provide a solution to a flow shop scheduling problem in which the target is to minimize the total tardiness and the total maintenance costs. Rohaninejad et al. (2015) proposed a tabu search algorithm to minimize the sum of the total tardiness cost, the extra time cost and the setup cost. Nevertheless, this research is made for a job shop scheduling problem. It is also found that the analysis of costs in the objective functions of scheduling problems is rarely studied in the literature, even less for a FFSP like the one proposed for sequencing the jobs in JLS Soap Factory.

To the best of our knowledge, there is no evidence of the study of a multiobjective FFSP that considers: the obtention of pareto solutions of total weighted tardiness and total setups costs, the allowing of skipping stages and the consideration of sequence dependent setup times.

2. Proposed solution approach

To solve the stated FFSP an NSGA2 algorithm (Deb et al., 2002) is proposed, using a crossover operator and three mutation operators, which pursue the diversification in the search of solutions throughout the solution space (Peng et al., 2018). Moreover, the application of a complete factorial design is proposed to determine the parameters of the metaheuristic application.

Initially, a population of size N is randomly generated. The population N will serve as parents in the first generation of the algorithm. Each individual of the population is then evaluated in the two objective functions that are going to be minimized. This process is performed in order to compare the solutions and find those that are non-dominated. Once the set of non-dominated solutions are found, they are assigned to a first F1 front. Subsequently, the solutions that were not classified in the F1 front, are compared with each other once again. Then, a second set of non-dominated points are found. This set is again classified in a second F2 front. This procedure is repeated successively until all the Fn fronts are found, and each of the points generated by the individuals of population N are classified.

Additionally, in order to favor the diversity of the solutions found, the density of points surrounding each of the solutions classified in the F fronts is estimated. This procedure is accomplished by calculating the average distance of two points that surround the evaluated solution side by side.

By completing this process, it is possible to order the entire population N of initial parents, first by the non-dominance of the resulting solutions and then, by the density of the points that surround each of the solutions. Afterwards, the selection of the chromosomes that will be crossed is done through a binary tournament. Then, a two-point crossover and three mutation operators named as inverse, insert and swap operation, are used to enhance the diversity of the solutions found.

Finally, the population of chromosomes chosen to be crossed and mutated, and the chromosomes originated from the previous operations, are gathered in a set P, which will be ordered according to the non-dominance of their solutions. This process allows to eliminate the worst individuals until obtaining a population of size N again. The same procedure is performed until the number of iterations defined as a parameter have reached its limit.

A diversity metric is applied to determine the quality of the combination of parameters in terms of the distribution of points along the pareto approach. The equation for the calculation of diversity is shown below:

$$Diversity = \frac{\left(dext1 + dext2 + \sum_{i=1}^{N-1}\left|di - \bar{d}\right|\right)}{\left(dext1 + dext2 + \bar{d}(N-1)\right)} \tag{1}$$

Where $dext1$ and $dext2$ correspond to the euclidean distances of the end points of the generated front, di corresponds to the euclidean distance between two consecutive points and $\bar{d}$ is the average of the euclidean distances of all the points found (Deb et al., 2002). Therefore, it is expected for di to be as close as possible to $\bar{d}$ so as to avoid the concentration of solutions in a single region of the approximated pareto front.

Four parameters are defined in order to analyze their influence in the quality of the solutions found by the algorithm, and consequently, in the result of the diversity metric. These parameters are probability of crossover (*Pcross*), probability of mutation (*Pmut*), initial population N (*P_ini*) and number of iterations (*Iter*).

To estimate the parameters of the algorithm, a value of *Pcross* equals to 0.5 is fixed. As a consequence, this study benefits diversification rather than intensification in the search throughout the solution space. Two levels for each parameter are established for the factorial design, and 30 randomly generated replicates of each combination are completed. *Table 1* shows the factors and their respective levels defined for the experiment.

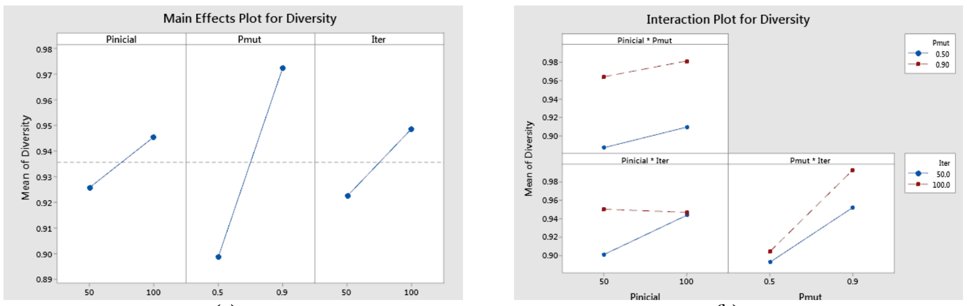
Table 1. Factors and levels of the factorial design.

	Parameters		
Levels	**P_ini**	**Pmut**	**Iter**
Low	50	0.5	50
High	100	0.9	100

The diversity for each set of solutions generated by the combination of parameters is the response variable of the experiment.

3. Results and discussion

According to the results, only the effect of *Pmut* is statistically significant (P-value <0.05), and there are not interactions between parameters. As a result, in order to favor small values for the diversity metric, a combination of [100, 0.5, 0.5, 100] in the parameters *P_ini*, *Pcross*, *Pmut*, and *Iter* respectively, is chosen to be applied for all instances created to evaluate the metaheuristic. Figure 1 shows the main effects plot and the interactions between factors plot.



(a) (b)

Figure 1. (a) Main effects plot for diversity metric. (b) Interaction plot of the parameters studied in the experiment.

The proposed approach was evaluated in thirty instances in order to assess the approximation of the pareto front obtained with the application of the NSGA2 algorithm. Additionally, the metaheuristic is compared with two priority rules: longest processing time of the jobs (LPT) and shortest processing time of the jobs (SPT). For each of the instances evaluated, it is found that the proposed NSGA2 algorithm can significantly improve the results obtained from the use of LPT and SPT priority rules in both the total setup cost and total weighted tardiness. Finally, it is found that each objective function presents a decreasing trend in each generation of the population, which shows the ability of the metaheuristic to find non-dominated solutions in each iteration.

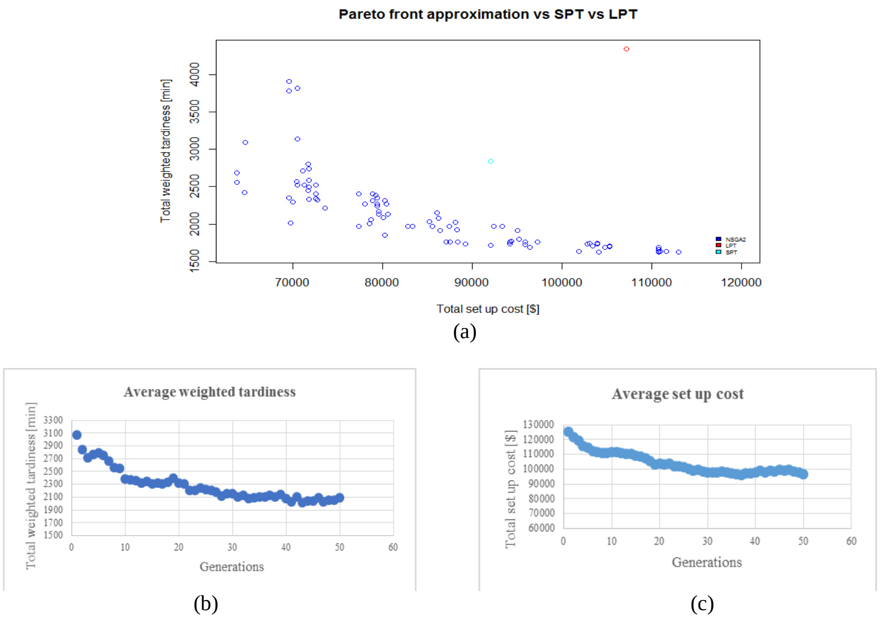


Figure 3. a) Pareto front Approximation, b) total weighted tardiness evolution and c) total setup cost evolution.

References

Ahonen, H. and de Alvarenga, A.G., 2017. “Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: formulation and solution procedures”. The International Journal of Advanced Manufacturing Technology, Vol. 89(1-4), pp.765-777.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.A.M.T., 2002. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. IEEE transactions on evolutionary computation, Vol. 6(2), pp.182-197.

Lassig, L., Mazzer, F., Nicolich, M. and Poloni, C., 2017. “Hybrid flow shop management: multi objective optimisation”. Procedia CIRP, Vol. 62, pp.147-152.

Peng, K., Wen, L., Li, R., Gao, L. and Li, X., 2018. “An effective hybrid algorithm for permutation flow shop scheduling problem with setup time”. Procedia CIRP, Vol. 72, pp.1288-1292.

Rabiee, M., Rad, R.S., Mazinani, M. and Shafaei, R., 2014. “An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines”. The International Journal of Advanced Manufacturing Technology, Vol. 71(5-8), pp.1229-1245.

Rohaninejad, M., Kheirkhah, A.S., Vahedi Nouri, B. and Fattahi, P., 2015. “Two hybrid tabu search–firefly algorithms for the capacitated job shop scheduling problem with sequence-dependent setup cost”. International Journal of Computer Integrated Manufacturing, Vol. 28(5), pp.470-487.

Talbi, E.G., 2009. Metaheuristics: from design to implementation (Vol. 74). John Wiley & Sons.

Torkashvand, M., Naderi, B. and Hosseini, S.A., 2017. “Modelling and scheduling multi-objective flow shop problems with interfering jobs”. Applied Soft Computing, Vol. 54, pp.221-228.

Yu, A.J. and Seif, J., 2016. “Minimizing tardiness and maintenance costs in flow shop scheduling by a lower-bound-based GA”. Computers & Industrial Engineering, Vol. 97, pp.26-40.

Yu, C., Semeraro, Q. and Matta, A., 2018. “A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility”. Computers & Operations Research, Vol. 100, pp.211-229.