

Solving a multi-objective manufacturing cell scheduling problem with the consideration of warehouses using a simulated annealing based procedure

Adrián A. Toncovich^{a*}, Daniel A. Rossit^{a,b}, Mariano Frutos^{a,c} and Diego G. Rossit^{a,c}

^aDepartamento de Ingeniería, Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca (8000), Argentina

^bINMABB-UNS-CONICET, Av. Alem 1253, Bahía Blanca (8000), Argentina

^cIISS UNS-CONICET, San Andrés 800, Bahía Blanca (8000), Argentina

CHRONICLE

Article history:

Received January 30 2018

Received in Revised Format

May 25 2018

Accepted June 5 2018

Available online

June 5 2018

Keywords:

Production Scheduling

Flow-shop

Pareto Archived Simulated

Annealing

Multi-objective Optimization

Warehouses

ABSTRACT

The competition manufacturing companies face has driven the development of novel and efficient methods that enhance the decision making process. In this work, a specific flow shop scheduling problem of practical interest in the industry is presented and formalized using a mathematical programming model. The problem considers a manufacturing system arranged as a work cell that takes into account the transport operations of raw material and final products between the manufacturing cell and warehouses. For solving this problem, we present a multiobjective metaheuristic strategy based on simulated annealing, the Pareto Archived Simulated Annealing (PASA). We tested this strategy on two kinds of benchmark problem sets proposed by the authors. The first group is composed by small-sized problems. On these tests, PASA was able to obtain optimal or near-optimal solutions in significantly short computing times. In order to complete the analysis, we compared these results to the exact Pareto front of the instances obtained with augmented ε -constraint method. Then, we also tested the algorithm in a set of larger problems to evaluate its performance in more extensive search spaces. We performed this assessment through an analysis of the hypervolume metric. Both sets of tests showed the competitiveness of the Pareto Archived Simulated Annealing to efficiently solve this problem and obtain good quality solutions while using reasonable computational resources.

© 2019 by the authors; licensee Growing Science, Canada

1. Introduction

The competition faced by manufacturing companies drives the development of innovative methods for better decision making. These methods must be equally fast enough to accomplish highly competitive results. One of the frequent problems tackled by these methods is the scheduling of manufacturing operations. The complexity associated with these problems limits considerably the possibilities of yielding optimal results for real-world cases, investing reasonable computational effort (Błażewicz et al., 2007; Pinedo, 2012; Tirkolaee et al., 2017). Even with these practical limitations, companies need to make continually short, medium and long-term planning and scheduling decisions. As operations scheduling has a significant economic impact on the performance of manufacturing companies, efficient approaches that swiftly provide competitive solutions, i.e., either optimal or “good enough” solutions, to these problems are required (Błażewicz et al., 2007; Pinedo, 2012; Framiñán et al., 2014).

* Corresponding author Tel.: +54 (0291) 4595101 - Ext.: 3222; Fax: +54 (0291) 4595119
E-mail: atoncovi@uns.edu.ar (A. A. Toncovich)

The manufacturing processes include the diverse operations that take part in the transformation from raw materials and basic components to the final products. The general formulation of a given manufacturing scheduling problem can be represented through mathematical programming (Ruiz et al., 2008; Sayadi et al., 2010; Defersha & Chen, 2012). In this work, we specifically address the Flow Shop Scheduling Problem (FSSP), which is a combinatorial optimization scheduling problem in the NP-Complete class (Garey et al., 1976). Given a set of jobs that have to be processed on a set of machines, FSSP can be defined as the problem of finding the best sequence (order) in terms of the relevant objectives in which those jobs should be processed on the machines, considering that this order should be respected for the whole set of machines. The flow shop production setting is appropriate to enhance industrial productivity and, therefore, more than a quarter of real-world production facilities have implemented it (Pan et al., 2011). There are even attempts to extend its benefits to other production processes that, by their very nature, do not have a flow shop configuration such as the case of manufacturing cells, which consist of groups of machines (cells) that process a set of jobs that have common processing characteristics. This allows adopting, within each cell, a similar configuration to a flow shop system. This production strategy is widely used in the industry (Joines et al., 1996; França et al., 2005). However, productivity improvements resulting from the use of this strategy are usually compromised when the cell is not well integrated with the surrounding manufacturing environment. Among other situations, we can mention an inefficient supply of materials required to perform each job or a poor planning in the location of the final products manufactured by the cell. To our knowledge, flow shop optimization literature has not addressed this problem. It is for this reason that, in this work, we conceived the manufacturing cell in a production system, which must provide the necessary materials for the manufacturing process, as well as, receive and store the final products.

In the scheduling research literature, the most popular performance measure used to evaluate the quality of the solutions for the FSSP is the makespan, which is defined as the total completion time required to process all the jobs in the system. Over time, other performance measures have been used and various operational conditions have been added to the basic configuration. For example, Wu et al. (2018) assessed the carbon footprint and the makespan in a multi-objective approach. Another possibility is to consider production and transport costs as it is performed in Ruiz-Torres et al. (2018), where the authors also pursue the minimization of the number of tardy jobs. In Wang and Tang (2017), the authors minimize the makespan, the total tardiness and the total flow time. Han et al. (2017) address a multi objective lot streaming flow shop problem, considering the makespan and the total tardiness as minimization objectives. In our case, the manufacturing system functions as a manufacturing cell environment and incorporates the raw material supply and the final product delivery. This problem considers the handling of the material for the production process, taking into account the supply of initial goods, which is associated with transport operations, and the storage of the final products, that also includes transport operations from the manufacturing cell to the final products warehouse. To the best of our knowledge these transport operations, which are important since they have a direct impact on the system performance, have not been studied in the literature, at least for manufacturing cells. The optimization strategy presented in this work addresses the makespan and the total tardiness in a multi-objective manner.

This paper contributes with a realistic problem, which is new to the literature. For this problem, a mathematical programming formulation is proposed and an appropriate solution methodology, which allows handling efficiently its combinatorial nature, is developed. The solution methodology is a multi-objective metaheuristic procedure that was adapted from the simulated annealing algorithm (Tirkolaei et al., 2016) and is able to generate a set of solutions that provides an approximation to the optimal Pareto front. We conducted a series of experiments in order to evaluate the performance of the proposed technique.

The article's organization is as follows: first, we present the FSSP and detail its configuration. We then formulate a mathematical programming model for the FSSP. Next, we present the Pareto archived simulated annealing procedure and introduce the augmented ϵ -constraint method as a comparative solution method. We then solve a set of small-sized problems using both approaches in order to compare

the results. Subsequently, we apply the Pareto archived simulated annealing algorithm to larger instances to test its performance. Finally, we outline the main conclusions and future research directions.

2. Problem Presentation

The production scheduling problem introduces a series of elements of interest in real production settings known as flexible manufacturing cells (Joines et al., 1996; França et al., 2005; Hendizaeh et al., 2008; Li et al., 2015; Fichera et al., 2017). The target problem is restricted to the analysis of the scheduling activities in a production environment constituted essentially by a set of machines arranged in a cell and two warehouses: one for raw material and another one for final product. Certain transfer operations of either raw materials or final products must be carried out between the warehouses and the manufacturing cell. These operations have an impact on the performance measures considered to optimize the sequencing problem. Naturally, the time required by these operations will depend on the type of product being processed (quantity, weight, volume, and shape of parts, components and final products). Centobelli et al. (2016) provide further insight on the subject material handling in warehouses.

In the next section, we propose a mathematical programming formulation to solve the problem that can be characterized as a skip flow shop, where jobs do not necessarily need to be processed in all the machines of the cell, with some additional features found in several production environments. Then, we present a multi-objective metaheuristic procedure to solve the sequencing problem. Some examples of metaheuristics for scheduling problems can be found in Minella et al. (2011) and in Shoaardebili and Fattahi (2015). For a deeper understanding on the issue, the review of Ghosh et al. (2011) presents a wide perspective of metaheuristics implementations on cellular manufacturing scheduling problems.

The performance measures used to evaluate the quality of the schedules take into account the utilization of production resources and the fulfillment of due dates. In addition, we considered raw material and final product warehouses transportation times in the model. In particular, we take into account two objectives when analyzing the quality of the candidate solutions: the minimization of the completion time of all jobs, or makespan, and the minimization of the total tardiness of all jobs. Fig. 1 shows an example configuration for the case of a manufacturing cell connected to a raw material warehouse and to a final product warehouse.

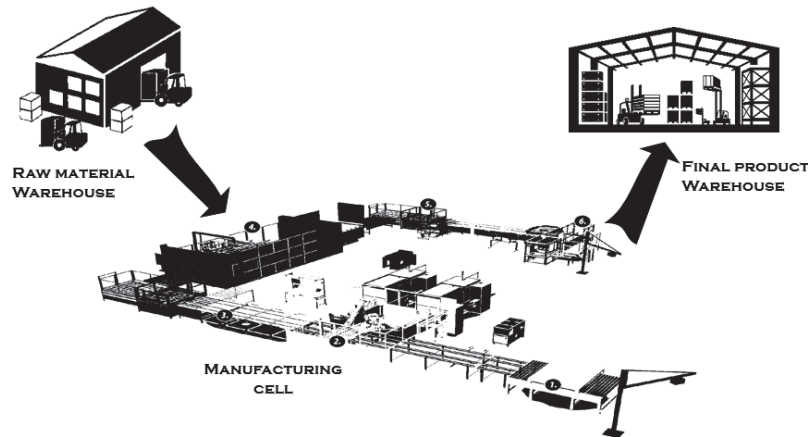


Fig. 1. Example configuration of a manufacturing cell with warehouses

3. Mathematical Formulation

The production configuration considered in this work is a manufacturing cell separated from the raw material warehouse and the final product warehouse. The jobs have a release date, and after their release, the raw materials for producing the job must be transported from the raw material warehouse to the manufacturing cell. Once the raw materials for the job have arrived at the manufacturing cell, the job can start to be processed. The production configuration inside the manufacturing cell is a flow shop setting,

thus all jobs have the same machine sequence. However, some jobs do not require processing in all the machines, namely, they can skip some operations. After a job has finished its processing, it must be transported from manufacturing cell to the final product warehouse. This last transportation time affects the delivery time of the job, since it has to be added to the completion time of the job in the manufacturing cell. As we stated before, the performance criteria considered for assessing the schedules are the makespan and the total tardiness.

Other assumptions made in the formulation of this problem are the following:

- Each machine can process only one job at a time.
- Each job can be processed by only one machine at a time.
- Preemption is not allowed.
- Machines can be idle during the planning horizon.

Sets

- Jobs: $\{j\}$
- Machines: $\{m\}$

Parameters

- $p_{j,m}$: processing time of job j on machine m
- t_j : transport time of job j from the raw material warehouse to the manufacturing cell
- r_j : release date of job j
- d_j : due date of job j
- tm_j : transport time of job j from the manufacturing cell to the final product warehouse
- Ω : a large positive number

Variables

- $C_{j,m}$: completion time of job j on machine m
- T_j : tardiness time of job j
- C_{max} : makespan
- $x_{j',j}$: binary variable, it has a value of 1 if job j' is processed before job j , and 0 otherwise.

Objective functions (1), minimize the makespan and the total tardiness.

$$\min \begin{cases} f_1 = C_{max} \\ f_2 = \sum_j T_j \end{cases} \quad (1)$$

Precedence Eq. (2) ensures that job j does not start its processing on machine m before it finishes its processing on machine $m - 1$.

$$C_{j,m} \geq C_{j,m-1} + p_{j,m}; \forall j, m > 1 \quad (2)$$

Ordering Eq. (3) requires that job j does not start its processing on machine m before all the previously sequenced jobs j' have been processed.

$$C_{j,m} \geq C_{j',m} + p_{j,m} - (1 - x_{j',j}) \cdot \Omega; \forall m, j' \neq j \quad (3)$$

Logic Eq. (4) enforces that if job j' is sequenced before job j , then the reverse is not valid.

$$x_{j',j} + x_{j,j'} = 1; j \neq j' \quad (4)$$

Raw material supply Eq. (5), each product cannot start its processing, before its release and raw materials arrive at the manufacturing cell.

$$C_{j,m} \geq r_j + t_j + p_{j,m}; m = 1, \forall j \quad (5)$$

Makespan calculation Eq. (6).

$$C_{max} \geq C_{j,m} + tm_j ; \forall j, \forall m \quad (6)$$

Tardiness in Eq. (7), for obtaining the tardiness of job j , it is necessary to consider the transportation time from the manufacturing cell to the final product warehouse.

$$T_j = \max\{0, (C_{j,m} + tm_j) - d_j\} ; \forall j, m = M \quad (7)$$

Non-negativity and binary variables value constraints in Eq. (8).

$$T_j, C_{j,m} > 0; x_{j,j'} \in \{0,1\} ; \forall j, \forall m \quad (8)$$

4. Proposed Solution Approaches to Address the Problem

In this section, we describe the algorithm based on simulated annealing used to solve the problem presented before and introduce the augmented ε -constraint method, used as a comparative solution method. The first algorithm has the structure of a simulated annealing type procedure with an archive of Pareto solutions (PASA, Pareto Archived Simulated Annealing).

Simulated Annealing (SA) is a local search based method developed from an analogy with the phenomenon of annealing to solve complex optimization problems (Kirkpatrick et al., 1983). Local search methods look for the solution with the best value of the chosen criterion in the neighborhood of the current solution, accept this new solution as the current solution, and repeat this step until it is not possible to further improve the solution in the explored neighborhood. By systematically applying this procedure, they obtain, in general, a local optimum of the problem. To avoid getting trapped in a local optimum, a diversifying mechanism should be incorporated with the aim of exploring the entire solution space. In the simulated annealing metaheuristic, the diversifying strategy allows moves, with a certain probability, toward solutions that worsen the current value of the objective function. SA has shown the capability of handling regular flow shop environments (Osman & Potts, 1989; Low, 2005; Vahedi Nouri et al., 2013).

4.1. PASA Procedure

This method, proposed in Engrand and Mouney (1998) uses an aggregation function of the objective functions, along with a non-dominated solutions file system. It is assumed that the objective functions to be minimized, f_p , $p = 1, 2, \dots, P$, are positive. Thus, the problem can be transformed into a mono-objective minimization problem using the following aggregation function:

$$z(S) = \sum_{p=1}^P \ln(f_p(S)), \quad (9)$$

where S is a given sequence. Hence, Eq. (10):

$$\Delta z = z(S_c) - z(S_A) = \sum_{p=1}^P \ln\left(\frac{f_p(S_c)}{f_p(S_A)}\right), \quad (10)$$

represents the average relative variation of the objective functions between the current solution (S_A) and the candidate solution (S_c). If $\Delta z > 0$, S_c deteriorates the relative mean of the set of objective functions. If $\Delta z \leq 0$, S_c improves or maintains the relative mean of the set of objective functions. In the first case the solution S_c is accepted with a probability given by $PA = e^{-\Delta z/T}$, where T is the control parameter that simulates the role of the temperature in the physical process of annealing. The method takes into account a non-dominated solutions file that is managed as follows:

- If at least one of the solutions in the file dominates S_C , it is not added to the file.
- If S_C dominates one or more solutions in the file, S_C is added, replacing the solutions that it dominates.
- If S_C does not dominate, nor is dominated by any file solution, S_C is added without replacing solutions.

In order to obtain an approximation to the entire efficient frontier of non-dominated solutions during the search process, it is necessary to restart the search regularly from one of the archived solutions selected at random.

PASA incorporates the classic parameters of the simulated annealing algorithm:

- T : Control parameter (temperature), a positive real value that varies from an initial higher value, T_0 , to a final lower value, T_f , during the execution of the algorithm.
- N_T : Number of iterations performed by the algorithm for a certain value of T .
- α : function of T , $\alpha = \alpha(T)$, which determines the variation of T . In general: $\alpha(T) = \alpha T$, in practice: $\alpha \in [0.80, 0.99]$.
- N_{stop} : Maximum number of iterations allowed without improvement.

The pseudo-code shown in Fig. 2 is applied to generate a set of potentially efficient (Pareto optimal) solutions.

i. Start

A constructive procedure is applied to generate an initial solution, S_0 .

Evaluate $f_p(S_0), \forall p$.

S_0 is incorporated into the set of efficient solutions: $SES = \{S_0\}, N_{cont} = t = 0, T = T_0$.

ii. Iteration t

A solution is randomly generated in the neighborhood of $S_A, S_C \in V(S_A)$.

Evaluate $f_p(S_C), \forall p$.

Calculate $\Delta z = z(S_C) - z(S_A) = \sum_{p=1}^P \ln \left(\frac{f_p(S_C)}{f_p(S_A)} \right)$.

If $\Delta z \leq 0$, the new solution is accepted: $S_A \leftarrow S_C, N_{cont} = 0$.

Otherwise, S_C is accepted with the following probability: $PA = e^{-\Delta z/T}$.

A random number ξ uniformly distributed in the interval $[0, 1]$ is generated:

$\left\{ \begin{array}{l} \text{If } \xi \leq PA, S_A \leftarrow S_C, N_{cont} = 0, \\ \text{otherwise, } S_A \leftarrow S_A, N_{cont} = N_{cont} + 1. \end{array} \right.$

If applicable, update SES taking into account S_C .

$T \leftarrow T + 1$: If T is a multiple of N_T , then $T = \alpha T$, otherwise the value of T is maintained.

If $N_{cont} = N_{stop}$ or $T < T_f$, execution is stopped, otherwise it is continued.

End

Fig. 2. PASA procedure

In the starting stage, the procedure constructs an initial solution by selecting in a random fashion the job that will be added to the sequence. We considered other constructive heuristic procedures in preliminary tests. However, from these tests we concluded that the simulated annealing procedure is competent enough, so it does not require a sophisticated procedure to initiate the algorithm (that can add unnecessary complexity to the solution approach).

Finally, we defined the neighborhood function used to generate a candidate solution as the set of solutions that can be obtained by randomly swapping two jobs in the sequence.

4.2. Augmented ε -constraint Method

In order to find the real Pareto Front of each instance and validate the proposed metaheuristic, we applied the augmented ε -constraint method presented in Mavrotas and Florios (2013). This approach is a novel variation of the original method developed by Haimes (1971), having several advantages not only over the traditional ε -constraint but also over another popular multi-objective approach as it is the weighted sum (Mavrotas, 2009; Mavrotas & Florios, 2013; Rossit et al. 2017). Namely, it guarantees the Pareto optimality of the solution and reduces the computing time by avoiding repeated solutions. We present the AUGMECON. Given a set of P objective functions, i.e., $f_1, f_2, \dots, f_p, \dots, f_P$, that we aim to minimize, the AUGMECON method solves the optimization problem that is described in Eqs. (11-15). These equations were adapted from the ones presented in Mavrotas & Florios (2013), formulated considering a set of objective functions that have to be maximized.

$$\min f_1(x) - \text{eps} \cdot \left(a_2 \frac{S_2}{r_2} + \dots + a_p \frac{S_p}{r_p} + \dots + a_p \frac{S_p}{r_p} \right) \quad (11)$$

subject to

$$f_2(x) + S_2 = \varepsilon_2 \quad (12)$$

...

$$f_p(x) + S_p = \varepsilon_p \quad (13)$$

...

$$f_p(x) + S_p = \varepsilon_p \quad (14)$$

$$x \in X \quad (15)$$

where r_p is the range of the p -th objective function over the efficient set, i.e., the distance between the ideal and the nadir value of the p -th objective function over the Pareto Front. S_p is the slack variable of the constraint associate with the p -th objective function. Analogously to the traditional method ε_p is the parameter for the Right Hand Side (RHS) for the constraint associated with the p -th objective function. Finally, eps is a small positive constant (between 10^{-6} and 10^{-3}). $x \in X$ represents the structural constraints of the problem that are not directly associated with the application of AUGMECON. In the initial version of the AUGMECON the coefficients a_p were all equal to one (Mavrotas, 2009). However, this was modified in the second version, AUGMECON2, presented in Mavrotas and Florios (2013) in which this values are calculated as $a_p = 10^{-(p-2)}$, what results in a lexicographic optimization over the objective functions in case there are alternative optima. Thus, the objective functions are optimized sequentially, i.e., first f_1 , then f_2 and so on. The range of the p -th objective function is divided in q_p equal intervals. Thus, we obtain in total $q_p + 1$ grid points that are used to systematically vary the RHS of the restriction associated with the p -th objective function (ε_p). If r_p is the range of the p -th objective function, then the discretization step is:

$$\text{step} = \frac{r_p}{q_p}. \quad (16)$$

Therefore, the RHS of the corresponding constraint associated with the n -th objective in the t -th iteration is:

$$\varepsilon_{pt} = fmax_p - step_p, \quad (17)$$

where $fmax_p$ is the nadir value for the p -th criteria. At each iteration, the surplus variable corresponding to the innermost objective function is checked. In the bicriteria problem presented in this paper, this is the objective function with $p = 2$. Then we calculate the bypass integer coefficient as:

$$b = \left\lfloor \frac{S_2}{step_2} \right\rfloor. \quad (18)$$

When the surplus variable S_2 is larger than $step_2$ in a t -th iteration, the same solution will be obtained in the next iteration $t+1$ with the only difference of a smaller surplus variable (that will have a value of $S_2 - step_2$). This makes $t+1$ iteration redundant and, therefore, it can be bypassed. The bypass coefficient b actually indicates how many consecutive iterations the procedure can jump ahead.

In integer programming problems, as long as we know the efficient ranges of the optimization criteria and the objective functions have integer values, we can generate the complete set of efficient solutions if we set $q_k \geq r_k$ (Mavrotas & Florios, 2013). For this paper, we obtained the ranges of the objective functions with lexicographic optimization to avoid weakly efficient values.

5. Computational Experimentation

In order to assess the mathematical formulation and the algorithm proposed in this work, we considered instances of two different sizes: the first group of instances involves up to 20 jobs and 20 machines and the second group of instances involves up to 100 jobs and 20 machines. While the first group is appropriate for validating the PASA algorithm through the comparison with the AUGMECON method, the second group of instances shows the capability of the proposed PASA algorithm for solving larger problems. In Table 1, the combinations of number of machines and number of jobs used to generate the instances are detailed.

Table 1

Detail of the test instances

<i>Instances</i>	<i>No. of jobs</i>	<i>No. of machines</i>
5J_5M	5	5
5J_10M	5	10
10J_5M	10	5
10J_10M	10	10
15J_10M	15	10
15J_15M	15	15
20J_10M	20	10
20J_20M	20	20
50J_15M	50	15
50J_20M	50	20
100J_20M	100	20

As we explained in the mathematical formulation section, several parameters must be set to define each scenario within each instance. In the case of processing times, $p_{j,m}$, the data are obtained from a pseudo-uniform distribution in the interval $[0,100]$. Since the problem addressed in this paper contemplates skipping operations, the zero value is included in the processing time distribution. Furthermore, and as a special feature, the probability of skipping an operation is set at 3%, that is, not all the possible values have the same probability (that is why there is a reference to a pseudo-uniform distribution). This amplifies the impact of skipping operations in the scheduling problem. For the transportation times from the raw material warehouse to the manufacturing cell and from the manufacturing cell to the final product warehouse, we take the data from a uniform distribution in the interval $[10, 20]$. Similarly, we generate the release dates of the jobs from a uniform distribution in the range $[1,100]$. With reference to the due

date of the jobs, we followed the guidelines given in Ruiz and Stützle (2008). However, in that article the due dates are calculated using the following expression: $d_j = r_j + \sum_m p_{j,m} \cdot (1 + random \cdot 3)$, which contemplates all the operations that job j must pass through for its completion plus an additional time considered by the term *random*, which is a random number uniformly distributed in $[0,1]$. In our case, each job has more intervals to consider, namely, the transportation times associated with arriving and leaving the manufacturing cell. Thus, in this paper the due date of job j is calculated as follows: $d_j = r_j + t_{transj} + t_{M,Wj} + \sum_m p_{j,m} \cdot (1 + random \cdot 0.3)$. Moreover, in our due date formulation the constant that multiplies the random number has been reduced from 3 to 0.3 for the small instances (with up to 20 jobs). This produces tighter due dates, giving a higher relevance to the total tardiness with respect to the makespan in the multi-objective evaluation. For each instance, we generated five different sets of parameters (scenarios). For the large instances (more than 50 jobs) the constant that multiplies the random number used in the due date generation was set at 3, respecting the original proposal of the cited paper.

The instances solved using the mathematical programming formulation were addressed using Pyomo (Hart et al., 2011; Hart et al., 2012) as modelling language, and CPLEX 12.5.0 as a solver. The tests were run on a CPU with 8GB of RAM and an I-core 5 processor, and a 64-bit operating system, the run time was limited to 1 hour (3600 seconds). CPLEX could solve instances up to 10 jobs and 10 machines, for the following instance (15 jobs and 10 machines) it could not find a solution with a gap lower than 3%, in average, the gap reported by CPLEX after an hour of computing was larger than 30%. We discarded these solutions. We programmed the instances solved using the simulated annealing metaheuristic in VBA (Visual Basic for Applications) as a modelling language in an MS-Excel setting. The tests were run on a CPU with 8GB of RAM and an I-core 7 processor, with a 64-bit operating system.

5.1. Experimental Design

The intention of the experimental design is to assess the ability of the proposed PASA algorithm to solve the problem at hand (Tirkolaee et al., 2018a; Tirkolaee et al., 2018b; Tirkolaee et al., 2018c; Zokaee et al., 2017; Amiri et al., 2018). For this purpose, we need to solve problems of different sizes. However, given that the PASA algorithm is a metaheuristic, we cannot assure the quality of the results by providing only a set of solutions that approximates the Pareto front in the best possible way. Furthermore, given the novelty of the problem, there is no known real Pareto front for the instances, except for the small ones, and due to its hardness, one cannot expect to calculate practically the optimal front. Thus, only the solutions obtained by PASA are considered, then, it remains to evaluate the precision of the solutions obtained with PASA with an exact method (a mathematical programming formulation solved with CPLEX in this case). In this way, it is possible to evaluate whether a given solution obtained with PASA is a real Pareto-optimal solution or not.

The methodology used to evaluate the set of the non-dominated solutions obtained by PASA, consists of assessing whether they are real non-dominated solutions of the problem, i.e., if they are Pareto optimal. For each solution x , two problems will be solved using the ε -constraint methodology. Each non-dominated solution obtained by PASA is defined by a pair of values of the total tardiness and the makespan objectives $(\sum_j T_j^*, C_{max}^*)$, in such a way that PASA cannot find other solution that improves one of the objectives without worsening the other, i.e., a new solution that dominates x . Therefore, we will evaluate the solution x using the ε -constraint method following a hypothesis test procedure: the hypothesis is that x is Pareto optimal. We will perform the hypothesis test demonstrating that it is not possible to improve it. What is done is to take the value $\sum_j T_j^*$ of x and fixing it as a constraint in the MIP model. Then, we try to minimize in a mono-objective manner the makespan, solving the mathematical programming formulation with CPLEX. If the resulting makespan coincides with C_{max}^* of x , and since CPLEX is an exact method, we verify that this makespan value corresponds to the actual Pareto front. We do the same with the other objective, we fix C_{max}^* as a constraint and we solve the MIP model by minimizing the total tardiness. Similarly if the resulting total tardiness coincides with $\sum_j T_j^*$, the value of total tardiness of x belongs to the real Pareto front. If these both value matches occur, then the solution

x is an optimal Pareto solution of the problem. Conversely, if one (or both) of these matches do not happen, then the exact method found a solution that dominates x and, thus, x is not Pareto optimal for the problem.

6. Results

As a first result of the problem, we present Figure 3, which shows the conflicting nature between the total tardiness and makespan objectives. This figure depicts the solutions obtained by PASA for scenario 3 of instance 50J_20M. We can see that when we minimize the makespan, the total tardiness objective deteriorates. Likewise, the same occurs in the other direction of the relation between objective functions: as the total tardiness reduces, the makespan increases.

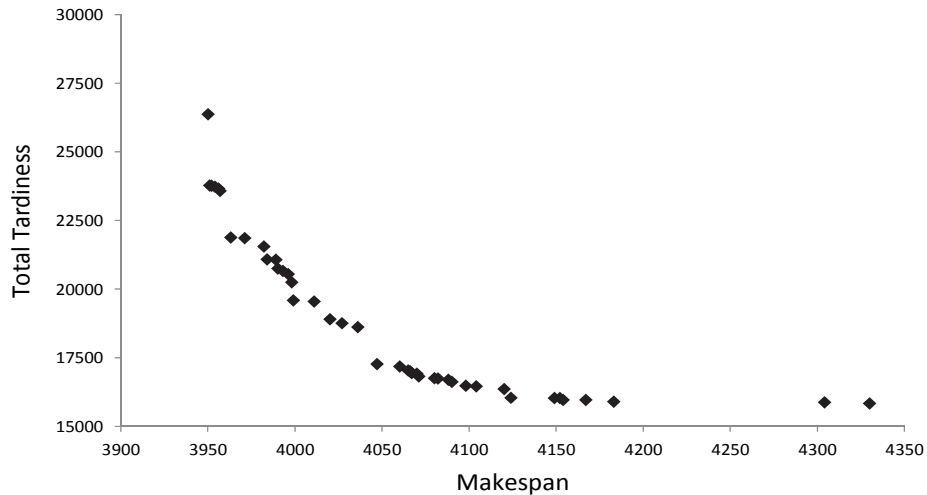


Fig. 3. Conflict between objectives - Objective function representation for scenario 3 of 50J_20M instance

The first stage of the evaluation consists of solving the problem with PASA and then evaluating if these solutions belong to the real Pareto front of the problem. To do this, we assess the PASA's solutions using the procedure described in the experimental design. In Table 2, we present the summary of the results of these assessments (we detail the information of each of the solutions in the Appendix Table). Table 2 shows for each of the scenarios tested for each instance: the average and the maximum number of solutions obtained with PASA; the percentage proportion of the PASA solutions that were verified to be Pareto optimal solutions of the real problem; and the number of solutions obtained with the augmented ε -constraint method, which represents the complete Pareto Front. In all the scenarios the PASA algorithm manages to find optimal Pareto solutions.

As we show in Table 2, the PASA algorithm was able to effectively address the problem for small instances in a very stable manner since the difference between the maximum number of found solutions and the average number in most of the cases is the same.

In the next stage, we evaluate the PASA algorithm performance for solving in the larger instances, i.e., the ones that have from 15 to 100 jobs and from 10 to 20 machines. For this purpose, we studied the hypervolume indicator obtained in the experimentation. We considered the average hypervolume indicator of the 30 runs that were completed for each scenario. Moreover, we presented the standard deviation and the coefficient of variation to provide a level of representation of the variability of the algorithm in the solution process. We depict all these results in Table 3. The average coefficient of variation for all the runs performed (7 instances \times 5 scenarios \times 30 runs per scenario: 1050) is 6.53%. Thus, the PASA algorithm is stable, obtaining always very similar solutions in general terms.

Table 2
Results of Pareto efficiency tests for PASA's solutions

Instance	Scenario	PASA		MIP % No. Solutions	AUGMENTED ϵ - CONSTRAINT No.
		Average	Max		
5J_5M	1	2	2	100	2
	2	8	8	100	8
	3	5	5	100	5
	4	7	7	100	7
	5	2	2	100	2
5J_10M	1	5	5	100	5
	2	3	3	100	3
	3	3	3	100	3
	4	5	5	100	5
	5	5	5	100	5
10J_5M	1	9	9	100	9
	2	5	5	100	5
	3	8	8	100	8
	4	5	6	100	6
	5	5	5	100	5
10J_10M	1	15	16	100	16
	2	7	7	100	7
	3	1	1	100	1
	4	4	5	100	5
	5	7	7	100	7

Table 3
Hypervolume analysis for the larger instances

Instance	Scenario	Hypervolume		
		Mean	Standard deviation	Coefficient of variation
15J_10M	1	8366721	235134	2,81%
	2	8489998	292030	3,44%
	3	8121540	254456	3,13%
	4	8013598	147072	1,84%
	5	8302312	292760	3,53%
15J_15M	1	9793596	1435394	14,66%
	2	10873222	195376	1,80%
	3	10171911	146859	1,44%
	4	11080475	297443	2,68%
	5	10752697	278537	2,59%
20J_10M	1	17064523	299007	1,75%
	2	17037822	300472	1,76%
	3	14484202	385328	2,66%
	4	16806410	631349	3,76%
	5	17470007	749474	4,29%
20J_20M	1	27545762	1129725	4,10%
	2	28345403	818171	2,89%
	3	30853641	2593627	8,41%
	4	28185167	748659	2,66%
	5	28885397	1026750	3,55%
50J_15M	1	142316697	10403149	7,31%
	2	83728978	11005042	13,14%
	3	116667840	9121458	7,82%
	4	106268432	8413715	7,92%
	5	127250867	11206569	8,81%
50J_20M	1	106077747	12131797	11,44%
	2	96180253	15490816	16,11%
	3	104781374	14374264	13,72%
	4	115853557	11161809	9,63%
	5	95798936	16476075	17,20%
100J_20M	1	1336150972	106412672	7,96%
	2	1425983428	86453796	6,06%
	3	1310145664	119455114	9,12%
	4	1276865560	131883613	10,33%
	5	1311489759	109568141	8,35%
<i>Average</i>				6,53%

When comparing the coefficients of variation of the scenarios for a given instance, we observed that there were variations between scenarios of the same problem size in terms of machines and jobs. This may be due to the fact that the data set of one scenario is more difficult to solve than the others, as it is explained in Vallada et al. (2015). Related to this, in order to avoid the case where the only tested scenario is an atypical tricky one, we evaluate more than one scenario for each instance. Nevertheless, the interesting issue to analyze is the convergence of the algorithm regarding the size of the problem, in order to do that we present Table 4.

Table 4

Hypervolume average coefficient of variation for each instance

<i>Instance</i>	<i>Coefficient of variation</i>
15J_10M	2,95%
15J_15M	4,63%
20J_10M	2,84%
20J_20M	4,32%
50J_15M	9,00%
50J_20M	13,62%
100J_20M	8,37%

Table 4 shows the average coefficient of variation for the 5 scenarios of each instance. As the size of the instance increases in terms of the number of machines and jobs, the coefficient of variation also tends to increase. This becomes more evident when passing from the instances of 20 jobs to the instances of 50 jobs. We can explain this behavior by the fact that increasing the size of the problem considerably increases the size of the space of feasible solutions. Thus, the algorithm does not always find the same solutions. However, this increase in the coefficient of variation appears to be bounded, since, for the largest instance of 100 jobs and 20 machines, it decreases with respect to the instances of 50 jobs and 20 machines.

7. Conclusions

In this work, we have studied a particular case of the FSSP (Flow Shop Scheduling Problem). This problem takes into account the material handling for the productive process considering both the supply of initial goods, with its associated transport operations, and the store of the final products, that also includes transport operations from the manufacturing cell to the final products warehouse. We have proposed a mathematical formulation for this realistic problem and presented a multi-objective simulated annealing algorithm (PASA). We have assessed the PASA solutions quality not only in terms of convergence, with a hypothesis test, but also in terms of diversification, with a comparison to the complete Pareto front obtained with the augmented ϵ -constraint method. These assessments have indicated that the proposed PASA algorithm was capable of addressing this problem, having a high performance on both convergence and Pareto front coverage.

We also addressed problems of larger size using the Pareto archived simulated annealing algorithm. For this set of problems, we observed that the results obtained by means of the algorithm were stable in terms of precision and the solutions were obtained in practical computational times.

Future research will consider larger instances of study and different performance measures with the aim of developing an accurate and fast method for solving this kind of scheduling problems.

References

- Amiri, M., Sadjadi, S. J., Tavakkoli-Moghaddam, R., & Jabbarzadeh, A. (2018). An integrated approach for facility location and supply vessel planning with time windows. *Journal of Optimization in Industrial Engineering*.
- Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling: from theory to applications*. Berlin Heidelberg: Springer Science & Business Media.

- Centobelli, P., Converso, G., Murino, T., & Santillo, L. (2016). Flow shop scheduling algorithm to optimize warehouse activities. *International Journal of Industrial Engineering Computations*, 7(1), 49-66.
- Defersha, F. M., & Chen, M. (2012). Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. *International Journal of Advanced Manufacturing Technology*, 62(1), 249-265.
- Engrand, P., & Mouney, X. (1998). Une méthode originale d'optimisation multiobjectif. Tech. Rep. HT-14/97/035/A (98NJ00005), EDF-DER Electricité de France, Direction des Études et Recherches, Clamart, France.
- Fichera, S., Costa, A., & Cappadonna, F. (2017). Heterogeneous workers with learning ability assignment in a cellular manufacturing system. *International Journal of Industrial Engineering Computations*, 8(4), 427-440.
- Framiñán, J. M., Leisten, R., & García, R. R. (2014). *Manufacturing scheduling systems*. Berlin: Springer.
- França, P. M., Gupta, J. N., Mendes, A. S., Moscato, P., & Veltink, K. J. (2005). Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers and Industrial Engineering*, 48(3), 491-506.
- Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117-129.
- Ghosh, T., Sengupta, S., Chattopadhyay, M., & Dan, P. (2011). Meta-heuristics in cellular manufacturing: A state-of-the-art review. *International Journal of Industrial Engineering Computations*, 2(1), 87-122.
- Haimes, Y. Y. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on Systems, Man, and Cybernetics*, 1(3), 296-297.
- Han, Y., Gong, D., Jin, Y., & Pan, Q. (2017). Evolutionary Multiobjective Blocking Lot-Streaming Flow Shop Scheduling With Machine Breakdowns. *IEEE Transactions on Cybernetics*.
- Hart, W. E., Laird, C., Watson, J. P., & Woodruff, D. L. (2012). *Pyomo—optimization modeling in python*. Vol. 67. Springer Science & Business Media.
- Hart, W. E., Watson, J. P., & Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3), 219-260.
- Joines, J. A., Culbreth, C. T., & King, R. E. (1996). Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transactions*, 28(1), 69-85.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Li, Y., Li, X., & Gupta, J. N. (2015). Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. *Expert Systems with Applications*, 42(3), 1409-1417.
- Low, C. (2005). Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers and Operations Research*, 32(8), 2013-2025.
- Mavrotas, G. (2009). Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2), 455-465.
- Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ϵ -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18), 9652-9669.
- Minella, G., Ruiz, R., & Ciavotta, M. (2011). Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems. *Computers and Operations Research*, 38(11), 1521-1533.
- Osman, I. H., & Potts, C. N. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega*, 17(6), 551-557.
- Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 181(12), 2455-2468.
- Pinedo, M. (2012). *Scheduling: theory, algorithms, and systems*. New Jersey: Prentice-Hall.

- Rossit, D. G., Tohmé, F. A., Frutos, M., & Broz, D. R. (2017). An application of the augmented ϵ -constraint method to design a municipal sorted waste collection system. *Decision Science Letters*, 6(4), 323-336.
- Ruiz, F., Şerifoğlu, R. S., & Urlings, T. (2008). Modeling realistic hybrid flexible flowshop scheduling problems. *Computers and Operations Research*, 35(4), 1151-1175.
- Ruiz, R., & Stützle, T. (2008). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3), 1143-1159.
- Ruiz-Torres, A. J., Paletta, G., Mahmoodi, F., & Ablanedo-Rosas, J. H. (2018). Scheduling assemble-to-order systems with multiple cells to minimize costs and tardy deliveries. *Computers & Industrial Engineering*, 115, 290-303.
- Sayadi, M., Ramezani, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, 1(1), 1-10.
- Shoardebili, N., & Fattahi, P. (2015). Multi-objective meta-heuristics to solve three-stage assembly flow shop scheduling problem with machine availability constraints. *International Journal of Production Research*, 53(3), 944-968.
- Tirkolae, E., Alinaghian, M., Bakhshi Sasi, M., & Seyyed Esfahani, M. M. (2016). Solving a robust capacitated arc routing problem using a hybrid simulated annealing algorithm: a waste collection application. *Journal of Industrial Engineering and Management Studies*, 3(1), 61-76.
- Tirkolae, E. B., Alinaghian, A., Hosseinabadi, A. A. R., Sasi, M. B., Sangaiah, A. K. (2018a). An improved ant colony optimization for the multi-trip Capacitated Arc Routing Problem, *Computers & Electrical Engineering*, Available online 13 February 2018.
- Tirkolae, E. B., Goli, A., Bakhshi, M., & Mahdavi, I. (2017). A robust multi-trip vehicle routing problem of perishable products with intermediate depots and time windows. *Numerical Algebra, Control & Optimization*, 7(4), 417-433.
- Tirkolae, E. B., Hosseinabadi, A. A. R., Soltani, M., Sangaiah, A. K., & Wang, J. (2018b). A Hybrid Genetic Algorithm for Multi-Trip Green Capacitated Arc Routing Problem in the Scope of Urban Services. *Sustainability*, 10(5), 1-21
- Tirkolae, E. B., Mahdavi, I., & Esfahani, M. M. S (2018c). A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. *Waste Management*, Available online 26 March 2018.
- Vahedi Nouri, B., Fattahi, P., & Ramezani, R. (2013). Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities. *International Journal of Production Research*, 51(12), 3501-3515.
- Vallada, E., Ruiz, R., & Framinan, J. M. (2015). New hard benchmark for flowshop scheduling problems minimising makespan. *European Journal of Operational Research*, 240(3), 666-677.
- Wang, X., & Tang, L. (2017). A machine-learning based memetic algorithm for the multi-objective permutation flowshop scheduling problem. *Computers & Operations Research*, 79, 60-77.
- Wu, X., Shen, X., & Cui, Q. (2018). Multi-Objective Flexible Flow Shop Scheduling Problem Considering Variable Processing Time due to Renewable Energy. *Sustainability*, 10(3), 841.
- Zokae, S., Jabbarzadeh, A., Fahimnia, B., & Sadjadi, S. J. (2017). Robust supply chain network design: an optimization model with real world application. *Annals of Operations Research*, 257(1-2), 15-44.

Appendix

Instance	Scenario	Solution	PASA		MIP	
			Cmax	Total Tardiness	Cmax	Total Tardiness
5J_5M	1	1	510	460	510	460
		2	554	454	554	454
	2	1	492	609	492	609
		2	539	398	539	398
		3	568	277	568	277

		4	553	281	553	281
		5	488	623	488	623
		6	541	343	541	343
		7	532	402	532	402
		8	511	526	511	526
		9	501	565	501	565
		1	479	570	479	570
	3	2	480	534	480	534
		3	517	434	517	434
		4	486	503	486	503
		5	514	450	514	450
		1	621	580	621	580
		2	707	517	707	517
	4	3	607	640	607	640
		4	652	573	652	573
		5	610	594	610	594
		6	608	607	608	607
		7	670	538	670	538
	5	1	581	628	581	628
		2	591	565	591	565
		1	929	465	929	465
		2	863	546	863	546
	1	3	860	626	860	626
		4	899	531	899	531
		5	858	635	858	635
		1	790	664	790	664
	2	2	835	596	835	596
		3	886	568	886	568
		1	799	353	799	353
	3	2	847	235	847	235
		3	795	394	795	394
		1	719	636	719	636
		2	925	443	925	443
	4	3	743	501	743	501
		4	746	464	746	464
		5	735	606	735	606
		1	869	402	869	402
		2	818	525	818	525
	5	3	855	479	855	479
		4	835	489	835	489
		5	804	588	804	588
		6	895	333	895	333
		1	844	1918	844	1918
		2	813	2332	813	2332
		3	817	2243	817	2243
		4	806	2333	806	2333
	1	5	819	2042	819	2042
		6	821	2006	821	2006
		7	829	1996	829	1996
		8	845	1897	845	1897
		9	833	1923	833	1923
		1	791	1788	791	1788
		2	776	1807	776	1807
	2	3	827	1778	827	1778
		4	899	1763	899	1763
		5	873	1777	873	1777
		1	871	2069	871	2069
		2	873	2054	873	2054
	3	3	944	1970	944	1970
		4	902	1980	902	1980
		5	887	2043	887	2043
		6	851	2158	851	2158
		7	849	2374	849	2374

		8	956	1968	956	1968
		1	766	1663	766	1663
		2	751	1743	751	1743
	4	3	763	1701	763	1701
		4	771	1626	771	1626
		5	775	1586	775	1586
		1	774	2145	774	2145
		2	747	2252	747	2252
	5	3	753	2169	753	2169
		4	749	2186	749	2186
		5	755	2148	755	2148
		1	990	1898	990	1898
		2	996	1852	996	1852
		3	1088	1662	1088	1662
		4	982	1953	982	1953
		5	988	1917	988	1917
		6	1035	1842	1035	1842
		7	975	2042	975	2042
	1	8	1011	1849	1011	1849
		9	1095	1651	1095	1651
		10	1047	1823	1047	1823
		11	1076	1758	1076	1758
		12	1046	1828	1046	1828
		13	987	1930	987	1930
		14	1080	1751	1080	1751
		15	1050	1779	1050	1779
		1	1206	2188	1206	2188
		2	1158	2200	1158	2200
		3	1126	2310	1126	2310
	2	4	1131	2303	1131	2303
		5	1145	2276	1145	2276
		6	1093	2313	1093	2313
		7	1155	2228	1155	2228
	3	1	1058	1502	1058	1502
		1	1089	1504	1089	1504
		2	1040	1531	1040	1531
	4	3	1010	1756	1010	1756
		4	1037	1589	1037	1589
		1	1056	2288	1056	2288
		2	1028	2642	1028	2642
		3	1060	2202	1060	2202
	5	4	1155	2136	1155	2136
		5	1033	2402	1033	2402
		6	1051	2368	1051	2368
		7	1111	2193	1111	2193

10J_10M



© 2019 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).