Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

A simheuristic for bi-objective stochastic permutation flow shop scheduling problem

Eliana María González-Neira[a*] and Jairo Rafael Montoya-Torres[b]

[a] *Department of Industrial Engineering, Faculty of Engineering, Pontificia Universidad Javeriana, Carrera 7 No. 40-62, Bogotá, D.C., Colombia*
[b] *Research Group in Logistics Systems, Faculty of Engineering, University of La Sabana. Km 7 north highway of Bogotá, D.C., Chía, Cundinamarca, Colombia*

CHRONICLE

Article history:
Received: October 14 2018
Received in revised format: November 19 2018
Accepted: January 7 2019
Available online:
January 8 2019

Keywords:
Stochastic permutation flow shop
Bi-objective
GRASP
Tardiness
Makespan

ABSTRACT

This paper addresses the stochastic permutation flow shop problem (SPFSP) in which the stochastic parameters are the processing times. This allows the modeling of setups and machine breakdowns. Likewise, it is proposed a multi-objective greedy randomized adaptive search procedure (GRASP) coupled with Monte-Carlo Simulation to obtain expected makespan and expected tardiness. To manage the bi-objective function, a sequential combined method is considered in the construction phase of the meta-heuristic. Moreover, the local Search combines 2-optimal interchanges with a Pareto Archived Evolution Strategy (PAES) to obtain the Pareto front. Also, some Taillard benchmark instances of deterministic permutation flow shop problem were adapted in order to include the variation in processing times. Accordingly, two coefficients of variation (CVs) were tested: one depending on expected processing times values defined as twice the expected processing time of a job, and a fixed value of 0.25. Thus, the computational results on benchmark instances show that the variable CV provided lower values of the expected makespan and tardiness, while the constant CV presented higher expected measures. The computational results present insights for further analysis on the behavior of stochastic scheduling problems for a better approach in real-life scenarios at industrial and service systems.

1. Introduction

Scheduling deals with the efficient allocation of a set of resources to the execution of a set of jobs or tasks. One of the most studied scheduling problems is the Flow Shop Problem (FSP). The FSP is characterized by the continuous and uninterrupted flow of n jobs through m machines. Therefore, the flow is unidirectional implying that all jobs follow the same route on the shop floor (Pinedo, 2012), meaning that there are $(n!)^m$ possible solutions. It is demonstrated that this problem for three or more machines is NP-hard (Garey & Johnson, 1977), which means that it is not possible to find optimal solutions for large-sized data sets in a reasonable computational time. For this reason, a lot of heuristics and meta-heuristics approaches have been proposed to solve it. A special case of FSP consists of jobs having the same processing sequence in all machines. This configuration is called Permutation Flow Shop Problem (PFSP) and it reduces the solution space to $n!$ possible outcomes. This kind of sequencing is appropriate in real-world industries in which in-process storage of products is very limited (Ciavotta et al., 2013).

* Corresponding author. Tel: (+57.1) 3208320 ext. 5306
E-mail address: eliana.gonzalez@javeriana.edu.co (E. M. González-Neira)

doi: 10.5267/j.jpm.2019.1.003

A wide variety of researches have been done in the academic literature of FSP. As a result, extensive reviews of the state of the art have been widely studied for single objective deterministic cases (Framinan et al., 2004; Hejazi & Saghafian, 2005; Ruiz & Maroto, 2005; Gupta & Stafford, 2006; Vallada et al., 2008; Pan & Ruiz, 2013), and multi-objective deterministic FSP (Minella et al., 2008; Sun et al., 2011; Yenisey and Yagmahan, 2014). Nevertheless, only two reviews were found in Stochastic FSP (SFSP) and Stochastic PFSP (SPFSP) (Gourgand et al., 2000; González-Neira et al., 2017). This shows that there is less research for SFSP than for deterministic FSP (Framinan & Perez-Gonzalez, 2015). According to (Aytug et al., 2005) the inability of great part of the research on scheduling problems to face with uncertainty was frequently mentioned as a main reason for the low influence of those researches in real-life industrial practice. Moreover, (Li & Ierapetritou, 2008) affirm that having systematic ways to consider the uncertainty is as important as the model itself. Therefore, stochastic scheduling literature has been growing recently.

With the previous ideas in mind, it is essential to notice that actual production systems are subject to different sources of randomness which can be caused by human resources, stochastic processing, setup or release times, machine breakdowns, modifications of due dates, out of stock of raw materials, and so on (Elyasi & Salmasi, 2013). Thus, the two of the most studied elements have been processing times and machine breakdowns. In fact, the occurrence of machine breakdowns and setup times can be modeled by modifying the statistical distributions of processing times in such a way that the possibility of breakdowns and setups be an integral part of processing times (Pinedo, 2012). Therefore, this study will include the processing times as the uncertain parameter, which represents a more realistic scenario that only solving the deterministic case (Juan et al., 2014). Processing times usually are modelled with Log-Normal, Normal, Weibull, Exponential distributions. Nevertheless, Normal distribution is unrealistic because it can produce negative times. Thence, Log-Normal and Weibull distributions are better options to consider (Juan et al., 2014). In fact, besides the non-negative values, Log-normal distribution has another practical characteristic for modelling processing times since it can represent high and low coefficients of variation (CVs) (Baker & Altheimer, 2012).

Although, there is plenty of deterministic FSP literature and some investigations have incorporated uncertainties into this problem, very few researches have addressed the multi-objective SFSP (MOSFSP). The analysis of multiple objective scheduling problems plays an important role in practice, because real life problems are naturally multi-objective problems (Sun et al., 2011; Yenisey & Yagmahan, 2014). Therefore, the aim of this paper is to study a SPFSP with stochastic processing times and minimization of expected makespan and expected tardiness to achieve both good machine utilization and fulfillment of delivery times. According to notation presented by (González-Neira et al., 2017), the problem is noted as $FS\left|prmu, p_{ij}\sim LogN\right|E[Cmax], E\left[\sum T_j\right]$. To solve the problem, a simheuristic approach is proposed using Greedy Randomized Adaptive Search Procedure (GRASP) to deal with the optimization and a Monte-Carlo Simulation to obtain the expected values of the two objective functions.

The remainder of the paper is organized as follows. Section 2 presents the review of related literature. Section 3 describes the integration between GRASP and Monte-Carlo simulation. Computational results are presented in Section 4 and concluding remarks and opportunities for further research are drawn in Section 5.

2. Literature review

Since the current paper solves a SPFSP with multiple objectives using a modification of GRASP meta-heuristic, this section is divided in two main parts. The first part presents a review of current solution approaches for the FSP under uncertainties focusing in multi-objective cases and the second part reviews works related to the application of GRASP for scheduling problems.

2.1 FSP under uncertainties

There is no as much studies in FSP that consider uncertainties in comparison with deterministic FSP. This is demonstrated by the difference in literature reviews of deterministic and uncertain counterparts in literature. For the deterministic FSP there are more than five literature reviews in the last five years, for mentioning (Pan & Ruiz, 2013; Yenisey & Yagmahan, 2014; Arora & Agarwal, 2016; Nagano & Miyata, 2016; Fernandez-Viagas et al., 2017; Rossit et al., 2018) whereas for the stochastic and uncertain counterparts there are only two literature reviews done by (Gourgand et al., 2000; González-Neira et al., 2017) in the last eighteen years. In FSP under uncertainties, mainly, one can find three methods to deal with uncertain parameters: stochastic, robust and fuzzy methods. The most used approach has been the stochastic in which uncertain parameters are modeled with probability distributions. Some examples are works of (Framinan & Perez-Gonzalez, 2015; Lin & Chen, 2015; Qin et al., 2015). The robust approaches models the uncertain parameters with intervals or data sets. In this case there is no need a knowledge of the distribution of data as can be seen in researches of (Ying, 2015; Fazayeli et al., 2016). Finally, in fuzzy methods the uncertain parameter are modeled with fuzzy numbers as showed in (Huang et al., 2012; Behnamian & Fatemi Ghomi, 2014).

Majority of research in uncertain FSP consider only one objective function, being the makespan the most studied one. Specifically, in multi-objective FSP under uncertainties few studies can be find. Those are briefly chronologically explained as follows. First, (Forst, 1995) developed a theorem to minimize the sum of the expected total weighted tardiness and the expected total weighted flowtime having stochastic processing times and common due dates in a PSFP. Second, (Celano et al., 2003) concentrated in a FSP with fuzzy processing times and fuzzy due dates to deal with maximum tardiness and makespan. They proposed an evolutionary algorithm which fitness function is a ratio between the average of fulfillment of due dates and the corresponding makespan. Furthermore, (Temiz & Erol, 2007) dealt with the minimization of fuzzy makespan, fuzzy maximum tardiness and fuzzy total flow time in a FS with uncertain processing times and due dates and proposed a genetic algorithm. (Zhou & Cui, 2008) designed a hybrid multi-objective genetic algorithm and run a simulation as a solution approach to solve a FSP with stochastic processing times and stochastic machine breakdowns.

In addition, (Azadeh et al., 2012) solved a SPFS with two machines. The objective function was weighted makespan plus weighted mean completion time. The authors proposed a simulation model whose results are inputs of an artificial neural network meta-model. The meta-model allowed them to reduce the simulation runs. Then, (Liefooghe et al., 2012) developed an indicator-based evolutionary algorithm to deal with a bi-objective PFSP with stochastic processing times. The objectives were makespan and total tardiness. (Rahmani et al., 2014) considered stochastic processing and release times in a FS to minimize the fuzzy weighted makespan, total flowtime and total tardiness through fuzzy goal programming. They applied Chance-constrained programming to transform the stochastic problem into a deterministic one. A genetic algorithm was developed to solve the deterministic and converted single objective problem. (Mou et al., 2015) developed a hybrid multi-objective evolutionary algorithm with NEH-based insertion method (Nawaz et al., 1983) to make a local improvement in a SPFS inverse scheduling problem. The objectives were to maintain the job sequence as far as possible (hamming distance) and the minimization of adjustment of total completion times and adjustments of processing times. Lastly, (Han et al., 2017) proposed a Non-dominated Sorting Genetic Algorithm II to find robust schedules for makespan and tardiness considering stochastic machine breakdowns.

2.2 GRASP for scheduling problems

GRASP is a multi-start meta-heuristic developed by (Feo & Resende, 1989). It has the advantage that does not need an initial solution to be constructed, and it is a procedure that does not use memory. Therefore, GRASP is a faster procedure in comparison with other meta-heuristics and has been widely used as a solution approach for different combinatorial problems as: set covering, location, planning and scheduling, quadratic assignment, transportation, among others. For more details, the reader is referred to Festa and Resende (2009).

As a matter of fact, some authors have used GRASP in specific applications to solve scheduling problems. For instance, single machine scheduling problems were studied by (Armentano & Araujo, 2006) and (Niño-Navarrete & Caballero-Villalobos, 2009). Former authors dealt with tardiness minimization while latter authors considered the total weighted tardiness as objective function. (Armentano and de França Filho, 2007) applied GRASP to total tardiness reduction and (Damodaran et al., 2011) minimized makespan, in a parallel machine environment. (Prabhaharan et al., 2006) solved a FSP problem with makespan objective, (Arroyo & de Souza Pereira, 2011) dealt with a multi-objective FSP involving in a first part makespan and maximum tardiness objectives, and in the second part tri-objective function adding the total flowtime. The proposed multi-objective GRASP was implemented under two approaches: scalarizing functions and Pareto dominance. (González-Neira et al., 2016) minimized weighted tardiness and considered the accomplishment of customer importance in a Hybrid FSP (HFSP). (Davoudpour & Ashrafi, 2009) considered a multi-objective function with four delivery-time-based criteria in a HFSP. (Rajkumar et al., 2011) solved a flexible jobshop (JS) scheduling problem with limited resource constraints for the minimization of makespan, maximum workload and total workload. Finally, (Chassaing et al., 2014) proposed a GRASP combined with an evolutionary local search to minimize makespan in a JS with simultaneously satisfaction of hard constraints.

3. Proposed GRASP

GRASP meta-heuristic is a multi-start iterative process that consists in two phases: construction and local search. For the first phase, a greedy function is required to build an initial solution. The greedy function has to be defined in such a way that helps to construct a good initial solution to achieve a certain objective previously defined. This function must be calculated for each job that is not in the solution yet. Next, the elements with best $\alpha\%$ values of that function determine a new set called Restricted Candidate List (RCL). An element that belongs to the RCL is selected at random to be part of the partial solution, and the set of elements that still are not in the solution is updated. This process is executed until the initial solution is completely constructed. As soon as the construction phase is done, local search begins until a local optimum is found. This process is repeated as many times as the number of iterations have been defined (Resende & Ribeiro, 2003).

Special variations of GRASP have been proposed to solve multi-objective problems. Those are combinations of two different strategies for construction phase and two for local search phase (Martí et al., 2015). The authors called these strategies: pure and combined. Pure strategies are those in which only one objective function guides each construction and the entire local search. Instead, combined methods consider more than one objective in each construction or local search. Specifically, sequential combined strategies mix, at each step of one construction or local search, the objectives functions that guide the process. In the case of construction phase of GRASP, a sequential combined method implies that a greedy function is selected at each step of the construction, whereas in the local this type of strategy entails that a different objective function is considered at each movement. In this paper, a GRASP with a sequential combined method for construction and local phases is proposed. It allows considering all the desired objectives in each entire construction and improves the solution until no movement can be done without deteriorating any of the considered objectives.

Due to the stochastic nature of the problem, a simheuristic that couples the GRASP algorithm with Monte-Carlo Simulations is proposed. The construction phase is ordered-sequential, which means that two greedy functions were defined for this problem, one attending to each of the objectives. In other words, the first step of the procedure uses the first greedy function, while the second greedy function is used in the second step; then the third step uses the first greedy function again, while the four step uses the second greedy function, and so on. Once all jobs have been sequenced, the proposed procedure runs a simulation with the number of runs required to give an accurate confidence interval of ±1% around the mean of makespan and tardiness, following the procedure proposed by (Framinan and Perez-Gonzalez, 2015). This solution is saved into the archive of Pareto front solutions. Then, the local search consists in 2-optimal interchanges between jobs. At each interchange the algorithm evaluates deterministically if the new sequence is a promise solution. If it is, then it is carried out a Monte Carlo simulation that also follows the procedure of (Framinan and Perez-Gonzalez, 2015) to estimate precise results of both objectives. Once the simulation is executed, the meta-heuristic evaluates if the solution is effectively Pareto-optimal for the stochastic case (i.e., if the solution should be saved on the archive of Pareto optimal solutions or not, depending on the solutions actually archived). GRASP ends when no interchanges can enter to the archive or until the maximum time is reached. Through preliminary experiments this maximum time was established as the number of jobs × number of machines × 1000 mili-seconds (100 seconds to give accurate stochastic Pareto solutions for instances of 20 jobs with 5 machines and 1000 second for instances of 50 jobs with 20 machines). Fig. 1 presents the pseudo-code of the GRAPS algorithm.

```
PROCEDURE GRASP

START PROCEDURE
          Step 1. Read problem's data
          Step 2. $time$ =0, $archive = \{\emptyset\}$
          WHILE $time \leq maximum\ time$
                    Step 3. $archive \leftarrow$ Construction_Phase
                    Step 4. $archive \leftarrow$ Local_Search_Phase
          END WHILE
          RETURN $archive$
END OF PROCEDURE
```

Fig. 1. GRASP procedure

3.1 Construction phase

For the construction phase two greedy functions were taken into account, one for each objective function. On one side, an adaptation of the dispatching rule called Longest Processing Time (LPT) was applied for makespan objective. On the other side, an adaptation of Modified Due Date (MDD) dispatching rule was implemented for tardiness minimization.

The LPT rule was selected for two reasons. Firstly, LPT was applied in (Nawaz et al., 1983) as part of a more complex solution method for makespan minimization. Also it was used by (Arroyo & de Souza Pereira, 2011) as a greedy function for the same objective in a GRASP meta-heuristic. The adaptation made to the LPT c in the current paper converted it in a dynamic rule that changes at each step of the construction. According to (Arroyo and de Souza Pereira, 2011), LPT indicates that jobs $(j \in \mathbf{J})$ has to be ordered in decreasing values of $\sum_{i \in \mathbf{M}} p_{ij}$ (where p_{ij} are the processing time of job j in machine i). As in PFSP there are m machines, the sum of processing times on all machines of a job is not actually the additional time that the job j will increment the makespan of a partial sequence. The real amount of time that makespan will increase, will be the difference between the makespan with job j on the partial sequence and the makespan before scheduling the job j in the partial schedule. Hence, jobs will be organized in increasing number of that difference that has to be recalculated in each step of the construction phase for all unscheduled jobs. That is, $LPT_j = C_{mj} - C_{mh}$, where C_{mj} is the completion time of job j at last machine m, h is the job scheduled at the position immediately before of the position of job j.

Correspondingly, the adaptation of Modified Due Date (MDD) dispatching rule, implemented by (Molina-Sánchez and González-Neira, 2016) in their proposed GRASP, was selected to handle tardiness objective. In the MDD rule, for single machine environments, the jobs $(j \in \mathbf{J})$ were arranged in increasing order of $\max(due\ date\ of\ job\ j, t + remaining\ work\ of\ job\ j)$. To adapt the rule for this particular problem, the proposed algorithm calculates which would be the completion time of job j if it would be sequenced at time t. Thence the MDD rule is obtained as $\max(d_j, C_{mj})$ at time t (where d_j is the due date of job j).

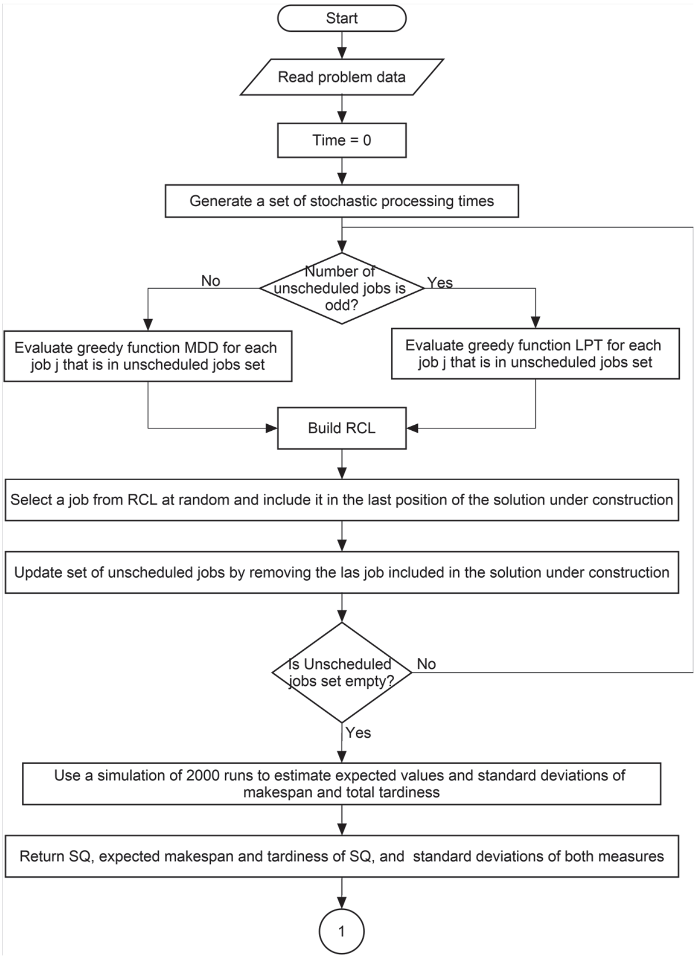
Start
Read problem data
Time = 0
Generate a set of stochastic processing times
Number of unscheduled jobs is odd?
No
Yes
Evaluate greedy function MDD for each job j that is in unscheduled jobs set
Evaluate greedy function LPT for each job j that is in unscheduled jobs set
Build RCL
Select a job from RCL at random and include it in the last position of the solution under construction
Update set of unscheduled jobs by removing the las job included in the solution under construction
Is Unscheduled jobs set empty?
No
Yes
Use a simulation of 2000 runs to estimate expected values and standard deviations of makespan and total tardiness
Return SQ, expected makespan and tardiness of SQ, and standard deviations of both measures
1

Fig. 2. Flow diagram of Construction phase of GRASP

The main idea of the construction procedure is using a different greedy function at each step. As there are two objective functions, the cost functions will be alternated one after another. Therefore, suppose that the procedure begins with LPT for the selection of the job in first position of the sequence, so MDD rule will be used next to choose a job for the second position of the sequence and for the third position, LPT is selected again, and so on. It means that at each step only one utility function is calculated for all remaining jobs.

Once one of the greedy functions has been evaluated, the RCL set is defined as the subset of jobs with the best value of $\alpha\%$ of the total range of utility function values obtained. Later, a job is randomly selected of RCL to form part of the partial solution. The procedure continues until all jobs have been scheduled. At this point, a Monte Carlo simulation is carried out to estimate the expected values of makespan and total tardiness of the obtained schedule. Later, the solution is added to the pareto archive using the whole procedure Pareto Archive Evolution Strategy (PAES) proposed by (Knowles and Corne, 2000). Fig. 2 shows the flow diagram of the procedure.

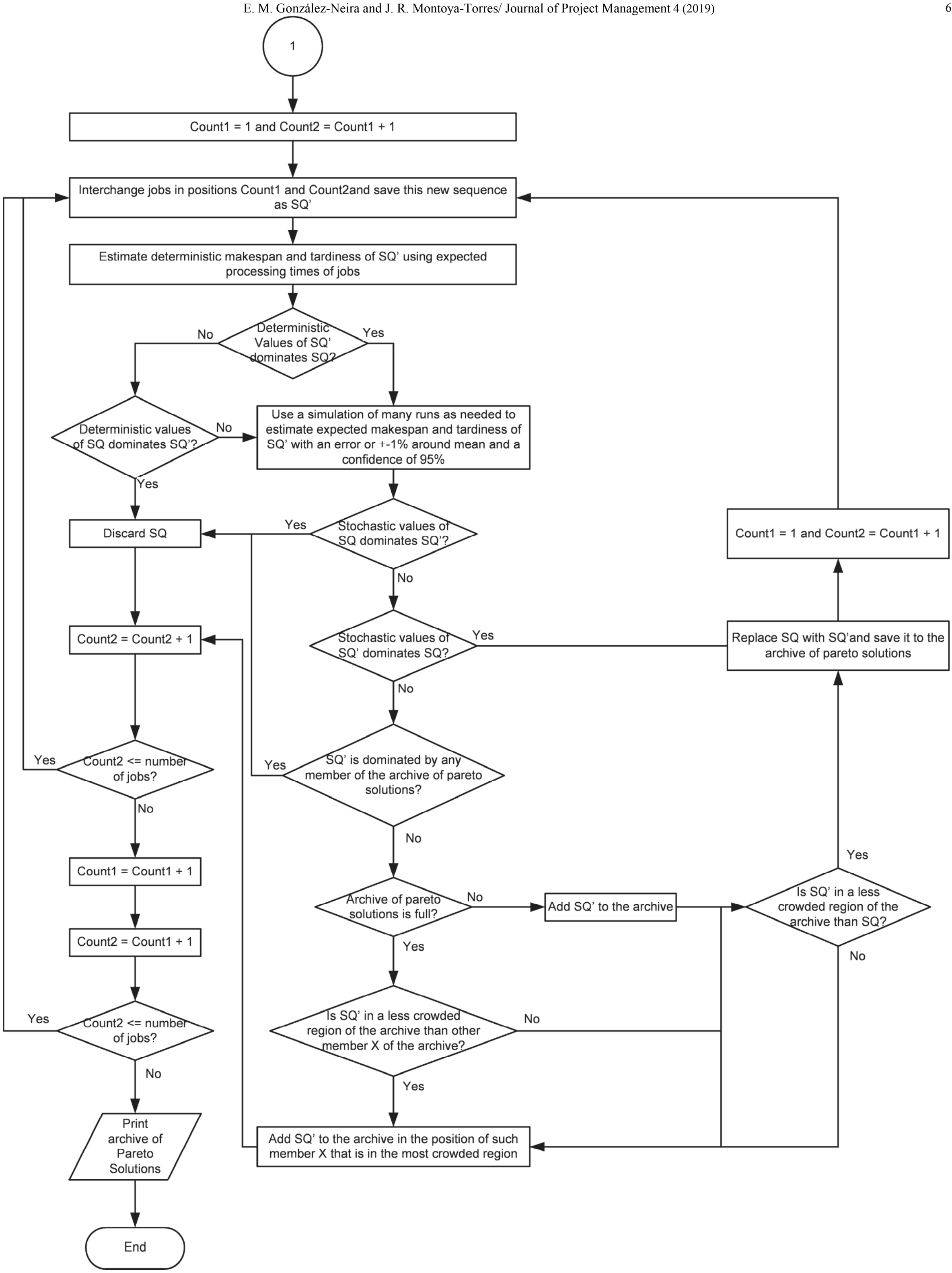
1
Count1 = 1 and Count2 = Count1 + 1
Interchange jobs in positions Count1 and Count2and save this new sequence as SQ'
Estimate deterministic makespan and tardiness of SQ' using expected processing times of jobs
Deterministic Values of SQ' dominates SQ?
No
Yes
Deterministic values of SQ dominates SQ'?
No
Yes
Use a simulation of many runs as needed to estimate expected makespan and tardiness of SQ' with an error or +-1% around mean and a confidence of 95%
Discard SQ
Yes
Stochastic values of SQ dominates SQ'?
No
Count1 = 1 and Count2 = Count1 + 1
Count2 = Count2 + 1
Stochastic values of SQ' dominates SQ?
Yes
Replace SQ with SQ'and save it to the archive of pareto solutions
No
Yes
Count2 <= number of jobs?
No
Yes
SQ' is dominated by any member of the archive of pareto solutions?
No
Yes
Count1 = Count1 + 1
Archive of pareto solutions is full?
No
Add SQ' to the archive
Is SQ' in a less crowded region of the archive than SQ?
Count2 = Count1 + 1
Yes
No
Yes
Count2 <= number of jobs?
Is SQ' in a less crowded region of the archive than other member X of the archive?
No
No
Yes
Print archive of Pareto Solutions
Add SQ' to the archive in the position of such member X that is in the most crowded region
End

Fig. 3. Flow diagram of Construction phase of GRASP

3.2 Implementation of Local Search phase

As proposed by (Molina-Sánchez and González-Neira, 2016), the local search implemented in the current GRASP is based on 2-optimal interchanges, originally designed for a Traveling Salesman

Problem (TSP) (Croes, 1958). To apply it for this scheduling problem, this phase makes an interchange of every two pair of jobs in the sequence given by the construction phase. At each interchange the algorithm evaluates if the deterministic values of makespan and tardiness of the new solution convert it in a promising solution to be added into the archive. If it is a possible good solution, the procedure executes a Monte Carlo simulation of many runs as needed to estimate a precise confidence interval of ±1% around the mean of makespan and tardiness of that sequence SQ', with a confidence of 95%. Next, PAES procedure, proposed by (Knowles and Corne, 2000), is applied to evaluate if SQ' must be included to the Pareto archive and if the initial solution SQ must be updated as SQ'. Fig. 3 presents the flowchart of the algorithm.

4. Computational results

GRASP was tested for the first 60 instances proposed originally by (Taillard, 1993) which were used by (Minella et al., 2008) for multi-objective deterministic optimization and by (Juan et al., 2014) for expected makespan minimization in a stochastic PFSP. There are 10 instances for each of the following combinations of number of jobs with number of machines $\{20 \times 5, 20 \times 10, 20 \times 20, 50 \times 5, 50 \times 10, 50 \times 20\}$. For that reason, lognormal distribution processing times are proposed. The expected values of processing times $E[p_{ij}]$ were the values of processing times p_{ij} of the mentioned benchmark instances. Two CVs were analyzed $\sqrt{2/E[p_{ij}]}$ and 0.25 in order to obtain the variances of processing times. On the one hand, a $CV = \sqrt{2/E[p_{ij}]}$ induces a variation on processing times of jobs and according to (Juan et al., 2014), this formula represents a high CV. On the other hand, a $CV = 0.25$ represents a fixed and equal value of processing times variability and, according to (Framinan and Perez-Gonzalez, 2015), this value is a low CV.

One Pareto front solution was obtained for each combination of instance with CV. Fig. 4 to Fig. 9 show Pareto-optimal solutions for six instances (one of each size). Most of these instances are the same evaluated in the work of (Juan et al., 2014).

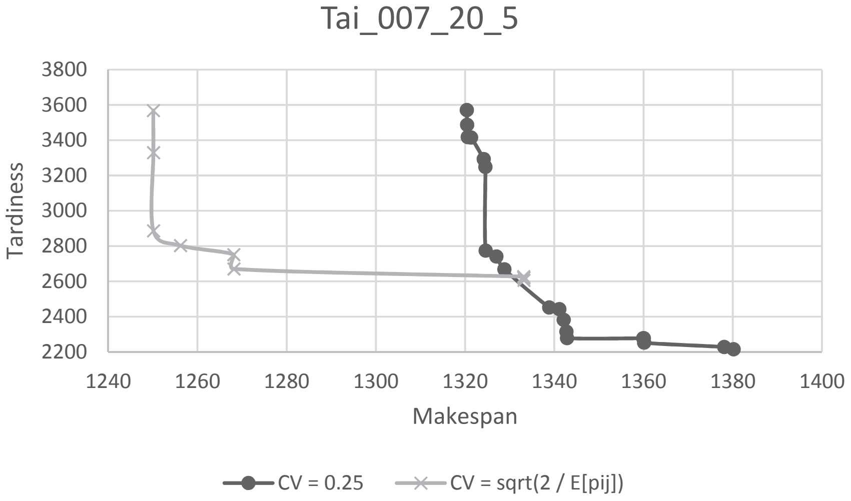

Fig. 4. Pareto fronts of Taillard's instance 7 (20 jobs × 5 machines)

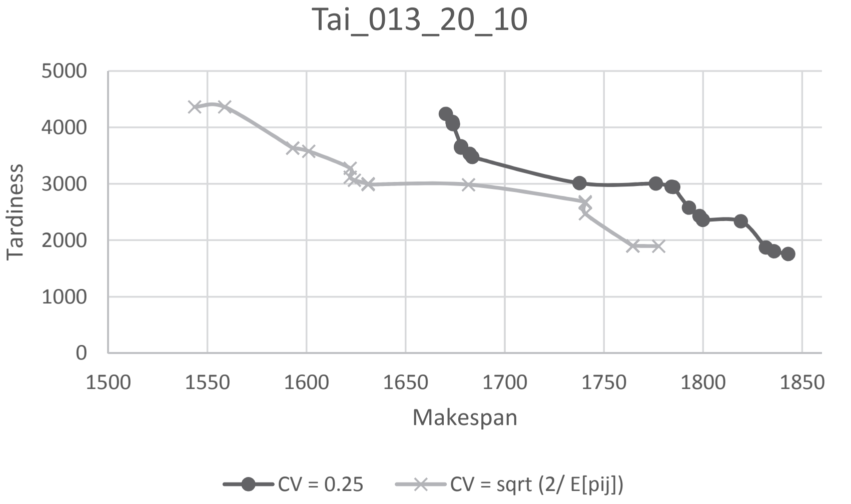

Fig. 5. Pareto fronts of Taillard's instance 13 (20 jobs × 10 machines)

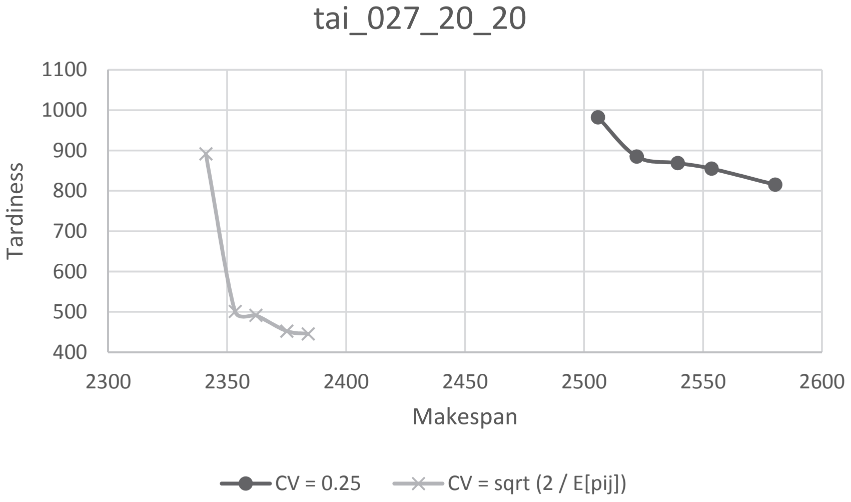

Fig. 6. Pareto fronts of Taillard's instance 27 (20 jobs × 20 machines)

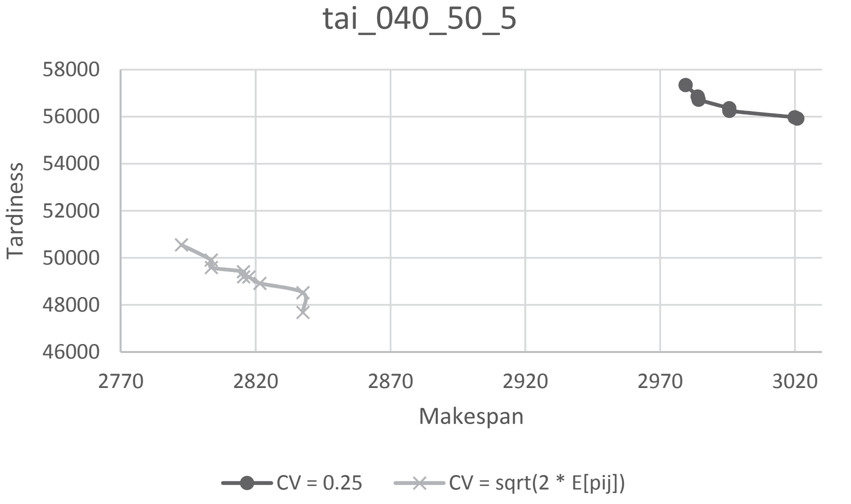

Fig. 7. Pareto fronts of Taillard's instance 40 (50 jobs × 5 machines)

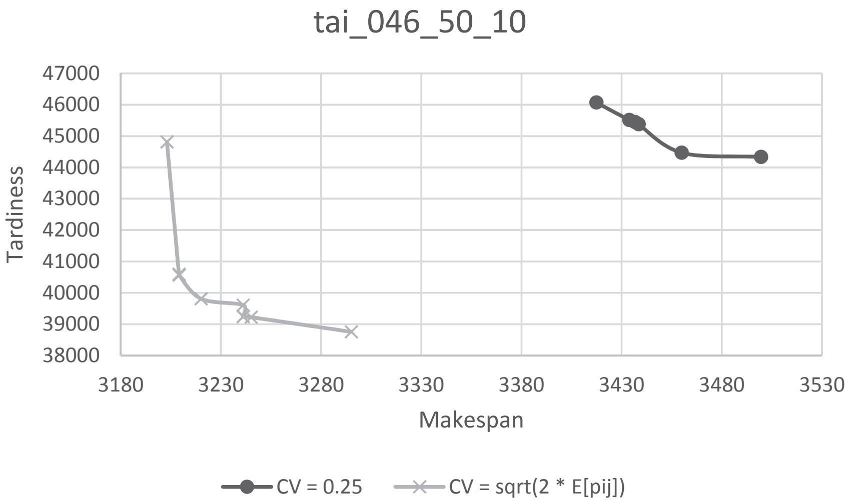

Fig. 8. Pareto fronts of Taillard's instance 46 (50 jobs × 10 machines)

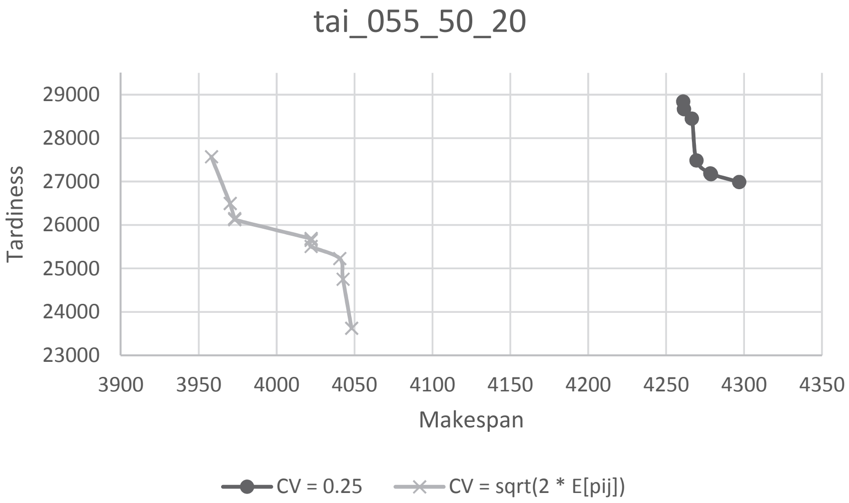

Fig. 9. Pareto fronts of Taillard's instance 55 (50 jobs × 20 machines)

From these results, a first output is that having a variance of processing times that depends directly on mean values (i.e., light gray lines with X marks representing high variance values defined in (Juan et al., 2014) actually give lower expected values of Pareto-optimal solutions than those obtained with a "low" CV = 0.25 traced with dark lines. This is explained by the fact that when $CV = \sqrt{2/E[p_{ij}]}$ the CV decrease as expected processing times increase. For instance, an expected processing time of 10 implies a CV=0.44 whereas an expected processing time of 80 represents a CV= 0.158. It can be confirmed by the standard deviation of both measures, as shown in Tables 1-6 that presents the results of extreme points of Pareto frontier for each tested instance.

Another important output of the current study is a comparison against previous published works. To the best of our knowledge, there exists only one work in SPFS that minimizes expected tardiness and makespan, that was presented by (Liefooghe et al., 2012). Nevertheless, this paper generated their own due dates for each instance, which are unknown, whereas this paper used the adapted Taillard benchamark instances that already have due dates, presented by (Minella et al., 2008). For that reason, first it is presented a set of four measures used in (Karimi et al., 2010; Ebrahimi et al., 2014) that can be taken for future comparisons. These measures are:

- Number of Pareto solutions (NPS): the number of non-dominated points for each instance
- Mean ideal distance (MID): it measures the closeness between Pareto solutions an ideal point (0,0). See Eq. (1).

$$MID = \frac{\sum_{i=1}^{NPS} c_i}{NPS} \qquad c_i = \sqrt{f_{1i}^2 + f_{2i}^2} \tag{1}$$

- Spread of non-dominance solution (SNS): is an indicator of diversity of Pareto points as indicates Eq. (2)

$$SNS = \sqrt{\frac{\sum_{i=1}^{NPS}(MID - c_i)^2}{NPS}} \tag{2}$$

- Rate of achievement to two objectives simultaneously (RAS): it denotes the balance in reaching two objective functions as shown Eq. (3)

$$RAS = \frac{\sum_{i=1}^{NPS}\left(\left(\frac{f_{1i} - F_i}{F_i}\right) + \left(\frac{f_{2i} - F_i}{F_i}\right)\right)}{NPS} \tag{3}$$

Tables 7-8 present the four mentioned indicators results for the 60 instances for both CVs tested. A half of the Pareto fronts had between 5 and 9 points and a 37% of the results more than 10 points. By comparing results for the two CVs evaluated the average MID and RAP are higher for a CV of 0.25 whereas the SNS is lower. This make sense because the CV of the Pareto set solutions when the processing times have a CV of 0.25 is above of the results when a CV of $\sqrt{2/E[p_{ij}]}$ is used.

Finally, in order to give an indicator of the quality of the solutions, a comparison between proposed GRASP and the algorithm presented by (Juan et al., 2014) was done using a relative gap metric for the expected makespan as shown in Eq. (4):

$$GAP = \left(\frac{Minimum\ expected\ Cmax\ of\ GRASP}{Minimum\ expected\ Cmax\ obtained\ by\ \text{Juan et al. (2014)}} - 1\right) * 100\% \tag{4}$$

As shown by the results in Table 9, the minimum expected makespan found by the proposed GRASP in comparison with the values obtained by (Juan et al., 2014) for the all common 14 instances presents an average gap of 3.98%. Although the time limit of execution proposed here is far greater than the used by those authors, very good results were obtained taking into account that the proposed algorithm is solving a multi-objective problem aiming at giving a complete Pareto front.

Table 1
Extreme points of Pareto front for instances 20 × 5 from Taillard

Instance	Extreme points of Pareto front									
	$\boldsymbol{CV = 0.25}$					$\boldsymbol{CV = \sqrt{\frac{2}{E[p_{ij}]}}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_001_20_5	1358.67	4135.64	61.67	507.99	40	1319.00	4657.00	0.00	0.00	100
tai_001_20_5	1375.22	3079.86	64.95	475.70		1419.00	2740.00	0.00	0.00	
tai_002_20_5	1431.62	5283.49	60.54	697.71	33	1363.00	3178.00	0.00	0.00	49
tai_002_20_5	1479.60	3548.92	65.16	563.40		1363.00	3178.00	0.00	0.00	
tai_003_20_5	1166.50	4914.33	53.01	598.70	87	1100.01	4349.04	0.17	0.98	27
tai_003_20_5	1252.18	4192.93	62.29	583.54		1127.87	3537.62	1.80	10.20	
tai_004_20_5	1407.29	5781.82	59.59	579.35	100	1307.00	3869.00	0.00	0.00	56
tai_004_20_5	1502.73	3857.78	64.95	570.80		1411.00	3430.00	0.00	0.00	
tai_005_20_5	1316.86	4502.35	60.42	414.00	77	1304.00	3213.01	0.08	0.61	70
tai_005_20_5	1421.49	3725.52	68.71	429.83		1314.02	2738.70	0.46	2.81	
tai_006_20_5	1338.84	6393.78	61.85	566.09	100	1214.00	4083.00	0.00	0.00	61
tai_006_20_5	1410.73	5230.85	66.55	489.73		1303.00	3421.00	0.00	0.00	
tai_007_20_5	1320.39	3570.65	60.93	555.68	64	1250.10	3568.20	1.84	17.17	26
tai_007_20_5	1380.18	2214.87	69.80	576.04		1333.16	2606.39	1.97	18.41	
tai_008_20_5	1309.38	4722.08	55.72	575.84	53	1239.00	4685.00	0.00	0.00	27
tai_008_20_5	1318.05	4600.15	56.48	531.80		1262.00	3964.12	0.00	1.23	
tai_009_20_5	1338.81	3695.49	56.19	532.19	66	1253.02	3616.02	0.36	0.36	45
tai_009_20_5	1352.21	3231.47	59.97	583.91		1286.09	2921.39	1.00	4.38	
tai_010_20_5	1219.73	4133.73	54.01	493.81	89	1149.95	4243.23	2.76	26.90	100
tai_010_20_5	1273.25	3745.13	58.04	462.80		1201.42	3205.36	1.83	5.76	

Table 2
Extreme points of Pareto front for instances 20 × 10 from Taillard

Instance	Extreme points of Pareto front									
	$CV = 0.25$					$CV = \sqrt{\frac{2}{E[p_{ij}]}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_011_20_10	1779.55	1110.84	68.08	241.66	115	1652.00	861.00	0.00	0.00	58
tai_011_20_10	1906.21	93.75	72.09	114.95		1756.00	97.00	0.00	0.00	
tai_012_20_10	1844.99	3067.12	64.58	437.55	99	1793.00	2767.00	0.00	0.00	21
tai_012_20_10	1881.02	2257.69	65.12	424.19		1823.00	2372.00	0.10	0.10	
tai_013_20_10	1670.31	4239.86	62.88	562.12	149	1543.49	4362.96	2.04	8.15	50
tai_013_20_10	1843.00	1756.26	72.98	319.84		1777.59	1895.51	1.29	7.76	
tai_014_20_10	1561.53	2307.54	57.58	260.99	45	1458.32	1974.36	1.38	2.01	44
tai_014_20_10	1661.79	2189.16	66.74	282.60		1514.00	1555.00	0.00	0.00	
tai_015_20_10	1587.82	1740.18	60.12	258.85	149	1471.01	1846.14	0.23	1.95	74
tai_015_20_10	1651.55	809.84	64.70	218.76		1567.01	1027.14	0.28	1.73	
tai_016_20_10	1543.71	4525.97	57.29	395.47	102	1489.00	3931.94	0.00	8.23	46
tai_016_20_10	1680.66	3688.99	65.62	428.28		1669.00	2920.16	0.00	3.03	
tai_017_20_10	1614.88	2162.61	63.80	273.57	100	1533.48	1519.96	1.88	3.75	38
tai_017_20_10	1675.42	905.65	65.58	266.99		1595.00	1308.01	0.00	0.22	
tai_018_20_10	1742.85	2794.03	68.22	323.45	78	1606.23	1190.21	2.05	2.26	62
tai_018_20_10	1804.29	1322.98	72.11	202.58		1710.23	916.76	1.31	2.39	
tai_019_20_10	1744.73	3113.06	67.04	553.25	137	1668.00	3394.00	0.00	0.00	57
tai_019_20_10	1878.81	2005.47	74.32	366.24		1706.54	2771.40	1.39	14.48	
tai_020_20_10	1777.35	2854.88	64.52	452.30	84	1619.00	2678.00	0.00	0.00	96
tai_020_20_10	1793.53	2838.54	65.48	474.24		1776.00	2481.00	0.00	0.00	

Table 3
Extreme points of Pareto front for instances 20 × 20 from Taillard

Instance	Extreme points of Pareto front									
	$CV = 0.25$					$CV = \sqrt{\frac{2}{E[p_{ij}]}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_021_20_20	2550.74	2245.27	80.41	265.37	53	2385.60	2316.31	1.34	4.46	91
tai_021_20_20	2610.89	1239.30	74.37	180.39		2456.13	461.64	1.95	5.34	
tai_022_20_20	2315.87	2806.43	69.69	316.03	280	2285.00	1587.00	0.00	0.00	59
tai_022_20_20	2471.83	766.89	76.37	223.72		2356.01	745.05	0.24	1.40	
tai_023_20_20	2567.13	1185.97	77.92	334.74	141	2381.02	1664.02	0.34	0.39	120
tai_023_20_20	2776.98	752.86	89.72	243.55		2455.87	710.75	1.94	3.89	
tai_024_20_20	2538.54	767.48	78.28	178.87	179	2406.00	450.00	0.00	0.00	99
tai_024_20_20	2538.54	767.48	78.28	178.87		2411.00	155.00	0.00	0.00	
tai_025_20_20	2526.98	6135.70	75.01	522.11	229	2325.01	6366.01	0.30	0.56	212
tai_025_20_20	2657.58	4586.34	81.03	628.94		2493.00	4039.00	0.00	0.00	
tai_026_20_20	2436.41	5166.45	71.74	443.56	119	2356.00	5265.58	0.02	1.34	85
tai_026_20_20	2509.23	4306.78	77.41	428.12		2422.59	3215.52	1.33	8.01	
tai_027_20_20	2505.86	982.06	71.54	228.17	85	2341.00	892.00	0.18	0.00	223
tai_027_20_20	2580.35	815.25	75.18	197.10		2384.01	445.62	0.28	3.22	
tai_028_20_20	2441.90	4701.82	74.60	308.89	400	2229.00	5249.00	0.02	0.07	206
tai_028_20_20	2574.14	1177.32	76.13	315.22		2353.00	2127.47	0.00	1.86	
tai_029_20_20	2464.23	3732.03	73.90	315.11	148	2304.70	2161.54	1.48	4.18	97
tai_029_20_20	2596.52	2349.27	79.82	275.66		2330.54	1690.78	1.27	6.01	
tai_030_20_20	2406.86	1729.20	74.21	185.91	400	2247.00	1210.00	0.00	0.00	201
tai_030_20_20	2573.53	4.85	83.38	24.74		2331.00	0.00	0.00	0.00	

Table 4
Extreme points of Pareto front for instances 50 × 5 from Taillard

Instance	Extreme points of Pareto front									
	$CV = 0.25$					$CV = \sqrt{\frac{2}{E[p_{ij}]}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_031_50_5	3017.31	55025.90	102.01	2512.21	250	2844.00	45636.30	0.00	42.62	250
tai_031_50_5	3052.18	54344.50	108.80	2459.58		2922.00	44431.20	0.00	45.86	
tai_032_50_5	3080.51	54509.80	88.84	2554.16	250	2949.66	48307.80	1.99	61.88	250
tai_032_50_5	3144.75	49361.70	98.08	2242.35		2969.57	46522.50	2.08	86.08	
tai_033_50_5	2815.07	45551.90	86.87	2128.25	250	2802.32	49235.60	2.81	104.99	250
tai_033_50_5	2836.77	43368.20	87.88	1889.76		2860.05	44940.90	2.37	89.44	
tai_034_50_5	3124.87	56994.20	90.92	2312.64	250	2888.70	54918.40	1.61	65.12	250
tai_034_50_5	3264.66	54884.20	103.16	2299.14		2933.07	52058.10	2.52	37.80	
tai_035_50_5	3111.37	56182.30	101.30	2427.76	250	3249.00	63716.00	0.00	0.00	250
tai_035_50_5	3279.78	52780.10	110.39	2539.06		3373.00	61995.60	0.00	1.41	
tai_036_50_5	3109.00	54355.20	92.75	2375.46	250	3010.00	52585.00	0.00	1.79	250
tai_036_50_5	3195.50	52217.50	98.75	2410.74		3063.01	50374.10	0.38	3.58	
tai_037_50_5	3046.32	56747.80	90.20	2335.03	250	2909.00	55132.00	0.06	8.96	250
tai_037_50_5	3107.68	54638.50	92.54	2434.74		2934.56	52938.30	1.27	30.17	
tai_038_50_5	2991.67	51714.00	93.73	2140.00	250	2874.05	51871.50	2.89	56.57	250
tai_038_50_5	3053.54	49790.00	95.09	2232.89		2969.34	47442.40	1.56	51.29	
tai_039_50_5	2732.52	44245.80	81.48	2082.15	250	2630.00	45358.10	0.14	45.36	250
tai_039_50_5	2748.91	43643.60	81.34	1990.38		2675.06	43856.00	0.62	29.25	
tai_040_50_5	2979.46	57345.90	84.66	2342.33	250	2792.43	50553.80	1.97	11.65	250
tai_040_50_5	3020.85	55919.90	86.94	2258.73		2675.06	43856.00	0.62	29.25	

Table 5
Extreme points of Pareto front for instances 50 × 10 from Taillard

Instance	Extreme points of Pareto front									
	$CV = 0.25$					$CV = \sqrt{\frac{2}{E[p_{ij}]}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_041_50_10	3451.08	43054.50	83.56	2252.72	500	3169.05	37112.10	2.91	21.57	500
tai_041_50_10	3502.08	38831.50	87.66	2176.92		3179.18	36600.60	2.32	28.23	
tai_042_50_10	3227.38	40206.60	82.49	2245.31	500	3089.88	37631.30	1.93	78.90	500
tai_042_50_10	3287.33	37926.20	84.04	2031.55		3172.90	36626.20	1.82	69.97	
tai_043_50_10	3478.99	50000.90	94.26	2196.55	500	3100.73	41362.20	3.50	114.29	500
tai_043_50_10	3507.58	47828.70	99.85	2185.42		3172.90	36626.20	1.82	69.97	
tai_044_50_10	3432.43	40421.60	86.83	1943.25	500	3239.56	35796.80	2.27	66.35	500
tai_044_50_10	3461.03	38517.80	88.09	2047.67		3295.55	34843.50	2.13	63.45	
tai_045_50_10	3401.79	42489.70	82.51	2279.91	500	3150.46	38337.00	3.10	37.50	500
tai_045_50_10	3405.70	42320.40	81.96	2318.29		3308.46	36471.30	3.32	43.21	
tai_046_50_10	3417.39	46068.00	90.67	2230.09	500	3203.01	44813.90	0.10	33.27	500
tai_046_50_10	3499.60	44332.10	89.98	2161.25		3295.01	38756.20	0.39	23.75	
tai_047_50_10	3512.31	48186.10	93.76	2088.37	500	3329.48	42513.20	2.00	54.52	500
tai_047_50_10	3545.39	46159.10	100.28	2016.07		3344.43	41699.70	1.90	38.11	
tai_048_50_10	3463.88	50404.00	87.20	2431.64	500	3269.32	46927.00	1.30	6.95	500
tai_048_50_10	3518.80	47209.00	93.62	2228.04		3416.00	42581.00	0.00	2.19	
tai_049_50_10	3296.89	53560.70	85.95	2143.20	500	3140.06	47557.70	2.48	49.23	500
tai_049_50_10	3410.00	48188.00	85.47	2232.06		3199.00	44437.10	0.00	17.98	
tai_050_50_10	3583.25	50673.10	95.23	2267.26	500	3292.59	43180.80	2.12	19.09	500
tai_050_50_10	3640.52	49181.30	102.58	2218.34		3392.03	40827.40	0.44	9.71	

Table 6
Extreme points of Pareto front for instances 50 × 20 from Taillard

Instance	Extreme points of Pareto front									
	$CV = 0.25$					$CV = \sqrt{\frac{2}{E[p_{ij}]}}$				
	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)	E[Cmax]	E[Tardiness]	Deviation Cmax	Deviation Tardiness	Execution time (s)
tai_051_50_20	4516.86	36035.60	96.38	2170.92	1000	4028.18	27708.80	1.17	6.51	1000
tai_051_50_20	4560.83	33083.90	98.57	2056.13		4140.40	24676.90	2.47	17.60	
tai_052_50_20	4339.36	37035.30	93.57	2493.43	1000	3975.94	27483.20	2.95	71.30	1000
tai_052_50_20	4387.87	32481.10	99.99	2141.07		4117.62	23872.10	1.58	53.12	
tai_053_50_20	4224.03	37696.80	88.80	2119.18	1000	3930.89	31939.80	3.86	83.02	1000
tai_053_50_20	4229.46	35606.40	90.94	1927.42		4159.46	26952.60	1.97	53.59	
tai_054_50_20	4439.46	36035.60	96.00	2075.96	1000	3978.01	29040.10	0.20	2.23	1000
tai_054_50_20	4451.88	34169.90	95.21	1917.38		4020.01	24036.10	0.40	2.81	
tai_055_50_20	4260.93	28836.10	88.94	1786.93	1000	3958.07	27571.30	0.67	40.66	1000
tai_055_50_20	4296.92	26987.80	97.85	1674.24		4048.06	23624.00	0.77	39.62	
tai_056_50_20	4311.50	30248.50	91.02	2073.96	1000	3904.29	20658.80	1.27	25.43	1000
tai_056_50_20	4356.17	27777.10	94.97	1935.34		4035.12	18921.30	0.88	15.87	
tai_057_50_20	4307.21	39668.90	91.49	1818.78	1000	3889.59	27351.60	1.98	55.64	1000
tai_057_50_20	4312.78	35990.20	97.37	1795.90		4012.02	24112.90	0.58	29.06	
tai_058_50_20	4319.57	34607.60	90.70	2102.56	1000	4023.93	25084.50	1.86	6.36	1000
tai_058_50_20	4366.66	31160.80	94.99	2156.65		4093.68	23059.90	2.46	32.47	
tai_059_50_20	4429.92	38508.30	92.43	1950.62	1000	4058.63	29175.40	2.92	20.57	1000
tai_059_50_20	4498.94	36064.20	101.14	1849.70		4073.04	26517.10	2.35	34.32	
tai_060_50_20	4420.73	44593.00	95.17	1914.63	1000	4007.22	38929.60	2.75	77.29	1000
tai_060_50_20	4456.07	42518.40	97.21	1765.02		4151.12	33718.40	2.56	71.94	

Table 7
Multi-objective indicators for instances of 20 jobs

Instance	$CV = 0.25$				$CV = \sqrt{\frac{2}{E[p_{ij}]}}$			
	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS
tai_001_20_5	4	3709.02	459.65	1.52	11	3824.70	601.61	1.62
tai_002_20_5	7	4126.28	595.63	1.67	1	3457.96	0.00	1.33
tai_003_20_5	7	4677.75	238.56	2.76	5	4012.90	300.12	2.47
tai_004_20_5	12	5034.57	559.16	2.32	6	3878.09	136.72	1.70
tai_005_20_5	13	4323.38	270.58	2.05	5	3227.55	191.11	1.26
tai_006_20_5	10	5896.34	364.74	3.22	5	3896.66	227.13	1.94
tai_007_20_5	19	3042.85	450.91	1.04	9	3150.14	307.75	1.25
tai_008_20_5	3	4838.26	58.03	2.55	6	4515.80	220.27	2.47
tai_009_20_5	4	3665.67	193.52	1.54	11	3530.78	191.80	1.61
tai_010_20_5	9	4072.71	136.44	2.13	12	3888.90	265.99	2.16
tai_011_20_10	20	1948.37	63.87	3.82	9	1775.44	55.37	6.45
tai_012_20_10	7	3311.95	264.16	0.47	4	3085.51	143.06	0.37
tai_013_20_10	18	3492.50	645.76	0.73	20	3389.89	628.64	0.77
tai_014_20_10	4	2766.15	15.49	0.43	8	2346.67	131.76	0.21
tai_015_20_10	9	2119.09	206.90	0.35	9	2076.75	228.71	0.31
tai_016_20_10	16	4542.95	183.34	1.67	7	3999.29	288.26	1.43
tai_017_20_10	13	2275.09	260.24	0.26	7	2099.68	51.87	0.10
tai_018_20_10	8	2770.07	336.47	0.28	4	1940.16	46.95	0.63
tai_019_20_10	9	3160.19	311.37	0.45	7	3613.31	193.75	0.90
tai_020_20_10	3	3359.12	3.33	0.59	4	3115.67	43.42	0.57
tai_021_20_20	9	3134.43	175.11	0.51	14	2896.18	352.69	1.40
tai_022_20_20	18	2937.92	348.16	0.81	9	2561.90	105.62	1.28
tai_023_20_20	7	2827.55	48.76	2.21	9	2751.33	170.93	1.24
tai_024_20_20	1	2652.02	0.00	2.31	2	2431.85	22.45	9.45
tai_025_20_20	15	6134.62	381.95	1.14	12	5576.27	844.77	1.08
tai_026_20_20	7	5456.12	240.22	0.98	12	4874.17	618.88	0.78
tai_027_20_20	5	2689.43	12.24	1.90	5	2433.40	40.74	3.54
tai_028_20_20	19	3899.55	733.04	0.41	12	4700.24	1017.09	0.81
tai_029_20_20	11	3881.83	296.65	0.20	11	3011.21	120.10	0.22
tai_030_20_20	16	2664.00	186.16	98.09	4	2362.01	128.79	58288.87

Table 8
Multi-objective indicators for instances of 50 jobs

Instance	$CV = 0.25$				$CV = \sqrt{\frac{2}{E[p_{ij}]}}$			
	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS
tai_031_50_5	10	54744.49	262.99	17.01	4	45085.65	556.54	14.62
tai_032_50_5	19	52430.33	2020.12	15.86	5	47217.24	703.79	14.90
tai_033_50_5	8	44586.99	939.35	14.75	8	46318.39	1356.09	15.44
tai_034_50_5	13	56149.84	781.33	16.60	13	53600.74	915.54	17.34
tai_035_50_5	15	54464.11	1041.25	16.15	7	62925.49	609.37	17.96
tai_036_50_5	7	53388.46	774.98	15.91	5	51268.07	1025.73	15.86
tai_037_50_5	17	55639.95	689.72	17.10	13	53976.17	826.30	17.46
tai_038_50_5	10	51018.31	671.63	15.86	20	49628.06	1413.94	16.07
tai_039_50_5	4	44039.57	253.98	15.04	14	44846.92	425.46	15.95
tai_040_50_5	7	56569.64	513.17	17.85	10	49090.47	1013.92	16.39
tai_041_50_10	13	41227.30	1741.73	10.81	5	36956.10	192.97	10.59
tai_042_50_10	11	39457.79	743.64	11.11	8	37190.88	364.34	10.80
tai_043_50_10	7	48768.14	690.29	12.91	6	40383.12	916.54	11.85
tai_044_50_10	7	39475.37	668.66	10.41	7	35622.30	389.00	9.86
tai_045_50_10	2	42541.44	119.11	11.46	9	37372.41	731.92	10.69
tai_046_50_10	6	45330.49	663.47	12.11	9	40336.50	1828.32	11.44
tai_047_50_10	6	47264.53	875.79	12.35	3	42258.57	406.38	11.63
tai_048_50_10	11	49289.57	1463.17	13.13	6	43836.09	1731.75	12.25
tai_049_50_10	13	50609.13	1371.00	14.12	13	46570.02	794.22	13.72
tai_050_50_10	6	50411.69	547.58	12.94	6	41982.15	819.35	11.49
tai_051_50_20	5	34340.79	1208.01	6.51	15	26378.77	1202.71	5.38
tai_052_50_20	8	34026.06	1420.07	6.75	8	25137.24	1281.16	5.14
tai_053_50_20	5	37408.45	879.20	7.80	13	29789.08	1770.35	6.30
tai_054_50_20	5	34922.51	783.07	6.79	4	27021.74	2526.82	5.70
tai_055_50_20	7	28150.22	782.00	5.51	12	26057.73	964.80	5.43
tai_056_50_20	8	29700.86	896.37	5.78	3	20385.25	907.32	4.07
tai_057_50_20	4	37336.93	1721.26	7.60	9	26018.79	1446.67	5.54
tai_058_50_20	8	32407.50	1515.20	6.39	15	24280.68	654.58	4.88
tai_059_50_20	11	37631.00	950.84	7.38	6	27707.28	989.37	5.73
tai_060_50_20	4	43282.51	1019.50	8.71	15	37094.28	1855.73	8.12

Table 9

GAP for Minimum makespan when $CV = \sqrt{2/E[p_{ij}]}$ in comparison with (Juan et al., 2014)

Instance	Minimum E[Cmax] of proposed GRASP	Best E[Cmax] by Juan et al. (2014)	GAP%
tai_007_20_5	1250.10	1248.10	0.16
tai_009_20_5	1253.02	1246.80	0.49
tai_010_20_5	1149.95	1126.60	2.07
tai_011_20_10	1652.00	1599.60	3.27
tai_013_20_10	1543.49	1511.90	2.09
tai_027_20_20	2341.00	2288.40	2.30
tai_036_20_20	3010.00	2834.40	6.19
tai_040_50_5	2792.43	2783.70	3.13
tai_044_50_10	3239.56	3079.00	5.21
tai_045_50_10	3150.46	3017.60	4.40
tai_046_50_10	3203.01	3036.40	5.48
tai_047_50_10	3329.48	3145.10	5.86
tai_052_50_20	3975.94	3733.30	6.50
tai_055_50_20	3958.07	3645.50	8.57

Table 10

Analysis of Variance of ration Tardiness/Cmax

Source	DF	AdjSS	AdjMS	F-Value	P-Value
A	1	6864,35	6864,35	8620,17	0,000
B	2	1434,92	717,46	900,98	0,000
C	1	41,81	41,81	52,51	0,000
D	1	9,85	9,85	12,37	0,001
A×B	2	625,42	312,71	392,70	0,000
A×C	1	2,01	2,01	2,53	0,113
A×D	1	4,06	4,06	5,10	0,025
B×C	2	0,65	0,33	0,41	0,664
B×D	2	3,19	1,59	2,00	0,138
C×D	1	0,12	0,12	0,15	0,697
A×B×C	2	0,04	0,02	0,02	0,976
A×B×D	2	7,12	3,56	4,47	0,013
A×C×D	1	0,33	0,33	0,41	0,523
B×C×D	2	0,12	0,06	0,08	0,925
A×B×C×D	2	0,00	0,00	0,00	0,999
Error	216	172,00	0,80		
Total	**239**	**9166,00**			

Additionally, an Analysis of Variance (ANOVA) was conducted to determine the effects of four factors on the ratio between tardiness and makespan values (5) at the two extreme points of the frontiers. The factors analyzed were: A: number of jobs, B: number of machines, C: objective function which extreme point value is minimum, and D: coefficient of variation. There was a total of 10 observations (one for each size of instance) per treatment. The ANOVA presented a $R_{adj}^2 = 97.92\%$ indicating that the main effects of all the analyzed factors significantly explain the ratio between tardiness and makespan, with a significance of 0.01 (Table 10). Assumptions of ANOVA that are normality, equal variances and independence were fulfilled with p-values of 0.017, 0.228 and 0.301 respectively.

$$Ratio = \frac{E[Tardiness]\ of\ solution\ X}{E[Cmax]\ of\ solution\ X} \tag{5}$$

As shown in Fig. 10, the largest main effect on the ratio measure is the number of jobs, showing an important increment of 600% between 20 and 50 jobs. In contrast, as the number of machines grows, the ratio reduces significantly. Additionally, although the effects of the extreme point and CV are less in amount in comparison to the number of jobs and number of machines, they are also statistically significant. The extreme point in which Cmax is minimum, represents a greater ratio of E[Tardiness]/E[Cmax] than the one in which Tardiness is minimum. It means that expected tardiness reduces in a higher proportion than the increase of expected Cmax. Finally, the CV has the lowest effect on the ratio of expected measures, nonetheless it is also significant. However, no final conclusion can be drawn because the first CV is not fixed (i.e., it depends on processing times), while the second CV is set to be fixed at the same value (0.25) for all jobs.

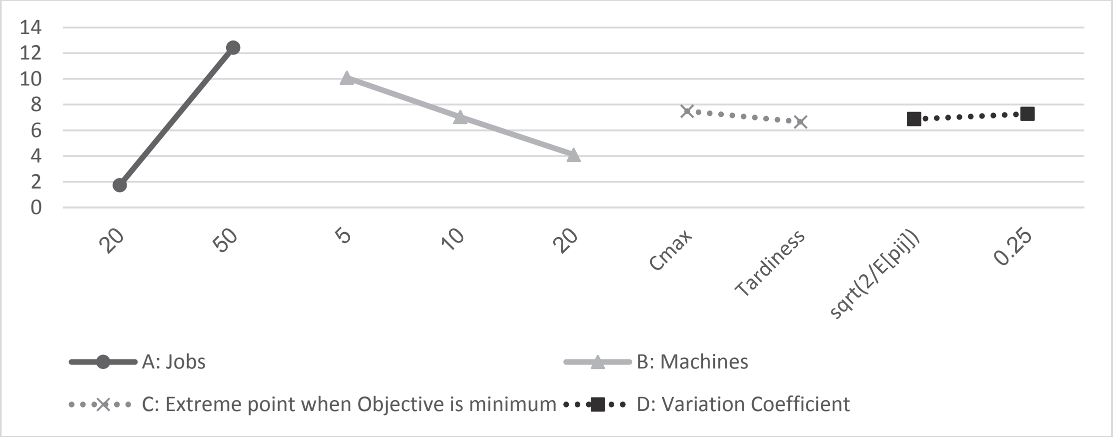

Fig. 10. Main effects plots for ratio E[Tardiness]/E[Cmax]

According to Fig. 11, the ratio E[Tardiness]/E[Cmax] increases from 20 to 50 jobs and decreases as the number of machines augments. It means that Cmax expands in a greater proportion than Tardiness when there are more machines.

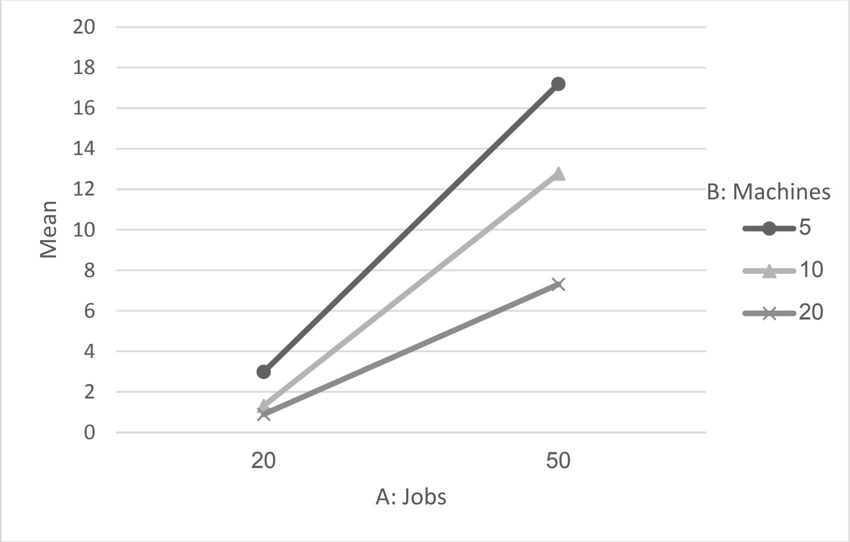

Fig. 11. Jobs-Machines Interaction plot for ratio E[Tardiness]/E[Cmax]

5. Conclusions and perspectives

In this paper it is proposed a GRASP meta-heuristic procedure hybridized with a Monte Carlo Simulation and a Pareto Archived Evolution Strategy (PAES) algorithm to obtain a complete set of Pareto-optimal solutions for a bi-objective stochastic permutation flowshop scheduling problem (SPFS). Objective functions were the expected makespan and the expected total tardiness. The proposed GRASP used a sequential combined method in both construction and local search phases to consider the two objective functions. The local search was based on 2-opt interchanges hybridized with PAES algorithm allowing obtaining a Pareto front. In order to apply PAES, the GRASP first evaluate if the deterministic new solution is a promising one; if so, a Monte Carlo Simulation is used to estimate expected values and evaluate if solution must be added to the archive of Pareto front.

Our algorithm was tested on first 60 Taillard's benchmark instances. Two different CVs were considered: one with fixed value of CV=0.25 defined of a "low" nature according to (Framinan and Perez-Gonzalez, 2015), and another with variable value of $CV = \sqrt{2/E[p_{ij}]}$, defined as of high variability according to Juan et al. (Juan et al., 2014). Results showed that for all instances the expected objectives are lower when the $CV = \sqrt{2/E[p_{ij}]}$ and far higher when CV=0.25. Finally, we compared the lowest expected makespan against previous works from the literature and obtained an average percentage deviation of 3.98%.

Besides, a statistical analysis was carried out to analyze the behavior of the ratio between expected tardiness and expected makespan at each extreme solution point of the frontier. Results showed that the number of jobs affects the ratio the most ascending and the number of machines inversely, i.e. the higher the number of jobs, the higher the ratio, but the higher the number of machines, the lower ratio. Moreover, if the extreme point is one in which the makespan is minimum the ratio indicator is greater than it in which the tardiness is minimum.

From the results obtained in this paper, several lines of future research can be highlighted. In the first instance, an interesting line is to carry out a comparison of many other different CVs and probability distributions to determine their effect on results of objective functions. In addition, regarding the proposed meta-heuristic, other greedy functions and local search procedures can be evaluated. Lastly, this work can be extended for other performance metrics (objective functions) such as flowtime and earliness and apply this algorithm to other environments such as flexible flow shop and job shop problems.

References

Armentano, V. A., & Araujo, O. C. B. de. (2006). Grasp with memory-based mechanisms for minimizing total tardiness in single machine scheduling with setup times. *Journal of Heuristics*, *12*(6), 427–446.

Armentano, V. A., & de França Filho, M. F. (2007). Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach. *European Journal of Operational Research*, *183*(1), 100–114.

Arora, D., & Agarwal, G. (2016). Meta-heuristic approaches for flowshop scheduling problems: a review. *International Journal of Advanced Operations Management*, *8*(1), 1.

Arroyo, J. E. C., & de Souza Pereira, A. A. (2011). A GRASP heuristic for the multi-objective permutation flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *55*(5–8), 741–753.

Aytug, H., Lawley, M. a., McKay, K., Mohan, S., & Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal* *of Operational Research*, *161*(1), 86–110.

Azadeh, A., Jeihoonian, M., Shoja, B. M., & Seyedmahmoudi, S. H. (2012). An integrated neural network–simulation algorithm for performance optimisation of the bi-criteria two-stage assembly flow-shop scheduling problem with stochastic activities. *International Journal of Production Research*, *50*(24), 7271–7284.

Baker, K. R., & Altheimer, D. (2012). Heuristic solution methods for the stochastic flow shop problem. *European Journal of Operational Research*, *216*(1), 172–177.

Behnamian, J., & Fatemi Ghomi, S. M. T. (2014). Multi-objective fuzzy multiprocessor flowshop scheduling. *Applied Soft Computing*, *21*, 139–148.

Celano, G., Costa, A., & Fichera, S. (2003). An evolutionary algorithm for pure fuzzy flowshop scheduling problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *11*(06), 655–669.

Chassaing, M., Fontanel, J., Lacomme, P., Ren, L., Tchernev, N., & Villechenon, P. (2014). A GRASP×ELS approach for the job-shop with a web service paradigm packaging. *Expert Systems with Applications*, *41*(2), 544–562.

Ciavotta, M., Minella, G., & Ruiz, R. (2013). Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. *European Journal of Operational Research*, *227*(2), 301–313.

Croes, G. A. (1958). A method for solving traveling-salesman Problems. *Operations Research*, *6*(6), 791–812.

Damodaran, P., Vélez-Gallego, M. C., & Maya, J. (2011). A GRASP approach for makespan minimization on parallel batch processing machines. *Journal of Intelligent Manufacturing*, *22*(5), 767–777.

Davoudpour, H., & Ashrafi, M. (2009). Solving multi-objective SDST flexible flow shop using GRASP algorithm. *The International Journal of Advanced Manufacturing Technology*, *44*(7–8), 737–747.

Ebrahimi, M., Fatemi Ghomi, S. M. T. M. T., & Karimi, B. (2014). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling*, *38*(9–10), 2490–2504.

Elyasi, A., & Salmasi, N. (2013). Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming. *Mathematical and Computer Modelling*, *57*(5–6), 1154–1164.

Fazayeli, M., Aleagha, M.-R., Bashirzadeh, R., & Shafaei, R. (2016). A hybrid meta-heuristic algorithm for flowshop robust scheduling under machine breakdown uncertainty. *International Journal of Computer Integrated Manufacturing*, *29*(7), 709–719.

Feo, T. A., & Resende, M. G. . (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, *8*(2), 67–71.

Fernandez-Viagas, V., Ruiz, R., & 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*, *257*(3), 707–721.

Festa, P., & Resende, M. G. C. (2009). An annotated bibliography of GRASP-Part II: Applications. *International Transactions in Operational Research*, *16*(2), 131–172.

Forst, F. G. (1995). Bicriterion stochastic scheduling on one or more machines. *European Journal of Operational Research*, *80*(2), 404–409.

Framinan, J. M., Gupta, J. N. D., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, *55*(12), 1243–1255.

Framinan, J. M., & Perez-Gonzalez, P. (2015). On heuristic solutions for the stochastic flowshop scheduling problem. *European Journal of Operational Research*, *246*(2), 413–420.

Garey, M. R., & Johnson, D. S. (1977). Two-processor scheduling with start-times and deadlines. *SIAM Journal on Computing*, *6*(3), 416–426.

González-Neira, E. M., García-Cáceres, R. G., Caballero-Villalobos, J. P., Molina-Sánchez, L. P., & Montoya-Torres, J. R. (2016). Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria. *Computers & Industrial Engineering*, *101*, 128–144.

González-Neira, E. M., Montoya-Torres, J. R., & Barrera, D. (2017). Flow-shop scheduling problem under uncertainties: Review and trends. *International Journal of Industrial Engineering Computations*, *8*(4), 399–426.

Gourgand, M., Grangeon, N., & Norre, S. (2000). A review of the static stochastic flow-shop scheduling problem. *Journal of Decision Systems*, *9*(2), 1–31.

Gupta, J. N. D., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, *169*(3), 699–711.

Han, Y., Gong, D., Jin, Y., & Pan, Q. (2017). Evolutionary Multiobjective Blocking Lot-Streaming Flow Shop Scheduling With Machine Breakdowns. *IEEE Transactions on Cybernetics*, 1–14.

Huang, C.-S., Huang, Y.-C., & Lai, P.-J. (2012). Modified genetic algorithms for solving fuzzy flow shop scheduling problems and their implementation with CUDA. *Expert Systems with Applications*, *39*(5), 4999–5005.

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *46*, 101–117.

Karimi, N., Zandieh, M., & Karamooz, H. R. (2010). Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert Systems with Applications*, *37*(6), 4024–4032.

Knowles, J. D., & Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, *8*(2), 149–172.

Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, *32*(4–5), 715–727.

Liefooghe, A., Basseur, M., Humeau, J., Jourdan, L., & Talbi, E.-G. (2012). On optimizing a bi-objective flowshop scheduling problem in an uncertain environment. *Computers & Mathematics with Applications*, *64*(12), 3747–3762.

Lin, J. T., & Chen, C.-M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory*, *51*, 100–114.

Martí, R., Campos, V., Resende, M. G. C., & Duarte, A. (2015). Multiobjective GRASP with Path Relinking. *European Journal of Operational Research*, *240*(1), 54–71.

Minella, G., Ruiz, R., & Ciavotta, M. (2008). A Review and Evaluation of Multiobjective Algorithms for the Flowshop Scheduling Problem. *INFORMS Journal on Computing*, *20*(3), 451–471.

Molina-Sánchez, L. P., & González-Neira, E. M. (2016). GRASP to minimize total weighted tardiness in a permutation flow shop environment. *International Journal of Industrial Engineering Computations*, *7*(1), 161–176.

Mou, J., Li, X., Gao, L., & Yi, W. (2015). An effective L-MONG algorithm for solving multi-objective flow-shop inverse scheduling problems. *Journal of Intelligent Manufacturing*.

Nagano, M. S., & Miyata, H. H. (2016). Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. *The International Journal of Advanced Manufacturing Technology*, *86*(5–8), 2161–2174.

Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95.

Niño-Navarrete, A. M., & Caballero-Villalobos, J. P. (2009). Evaluación de funciones de utilidad de GRASP en la programación de producción para minimizar la tardanza total ponderada en una máquina. *Revista Ingeniería*, *14*(2), 51–58.

Pan, Q.-K., & Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. *Computers & Operations Research*, *40*(1), 117–128.

Pinedo, M. L. (2012). *Scheduling: Theory, algorithms and systems*. *Springer* (4th ed., Vol. 4). New York: Springer Science & Business Media.

Prabhaharan, G., Khan, B. S. H., & Rakesh, L. (2006). Implementation of grasp in flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *30*(11–12), 1126–1131.

Qin, W., Zhang, J., & Song, D. (2015). An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. *Journal of Intelligent Manufacturing*.

Rahmani, D., Ramezanian, R., & Mehrabad, M. S. (2014). Multi-objective flow shop scheduling problem with stochastic parameters: fuzzy goal programming approach. *International Journal of Operational Research*, *21*(3), 322–340.

Rajkumar, M., Asokan, P., Anilkumar, N., & Page, T. (2011). A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, *49*(8), 2409–2423.

Resende, M. C., & Ribeiro, C. (2003). Greedy Randomized Adaptive Search Procedures. In F. Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics SE - 8* (Vol. 57, pp. 219–249). Springer US.

Reza Hejazi, S., & Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research*, *43*(14), 2895–2929.

Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The Non-Permutation Flow-Shop scheduling problem: A literature review. *Omega*, *77*, 143–153.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *165*(2), 479–494.

Sun, Y., Zhang, C., Gao, L., & Wang, X. (2011). Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *The International Journal of Advanced Manufacturing Technology*, *55*(5–8), 723–739.

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

Temiz, I., & Erol, S. (2007). Multiobjective genetic algorithm for fuzzy flowshop scheduling problem (Bulanik akiş tipi çizelgeleme problemi için çok amaçli genetik algoritma). *Journal of the Faculty of Engineering and Architecture of Gazi University*, *22*(4), 855–862. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-38549127324&partnerID=tZOtx3y1

Vallada, E., Ruiz, R., & Minella, G. (2008). Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers & Operations Research*, *35*(4), 1350–1373.

Yenisey, M. M., & Yagmahan, B. (2014). *Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends*. *Omega* (Vol. 45). Pergamon Press.

Yenisey, M. M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, *45*(C), 119–135.

Ying, K. (2015). Scheduling the two-machine flowshop to hedge against processing time uncertainty. *Journal of the Operational Research Society*, *66*(9), 1413–1425.

Zhou, Q., & Cui, X. (2008). Research on multiobjective flow shop scheduling with stochastic processing times and machine breakdowns. In *2008 IEEE International Conference on Service Operations and Logistics, and Informatics* (Vol. 22, pp. 1718–1724). IEEE.