

CONWIP FLOW SHOP BACKLOG SEQUENCING PROBLEM WITH RANDOM PROCESSING TIMES

Bruna Carvalho da Silva

Universidade Federal Rural do Semi-Árido
Rua Gamaliel Martins Bezerra, 587. Angicos - RN, 59515-000, Brasil
bruna.carvalho@ufersa.edu.br

Marco Aurélio de Mesquita

Escola Politécnica
Universidade de São Paulo
Av. Prof. Luciano Gualberto, 1380. São Paulo - SP, 05508-010, Brasil
marco.mesquita@poli.usp.br

ABSTRACT

This paper addresses the backlog sequencing problem in a flow shop controlled by a ConWiP production control model when processing times are random, and the objective is to minimize the expected makespan. For this problem, we evaluated the performance of two constructive heuristics: a deterministic heuristic applied to mean processing times and a heuristic designed specifically for the stochastic problem. We use simulation to estimate the expected makespan. In general, stochastic heuristic perform better than deterministic heuristic applied to mean processing times. However, this difference does not justify the computational cost of stochastic heuristic, at least within the limits of our experiments. Therefore, a more practical way to deal the stochastic problem is to resolve the deterministic counterpart, since it provides good sequences in reasonable computational time.

KEYWORDS. Backlog sequencing. Stochastic flow shop ConWiP. Optimization and simulation.

Combinational Optimization. Simulation

RESUMO

Este artigo aborda o problema de sequenciamento da fila de entrada de uma linha ConWiP quando os tempos de processamento são aleatórios e o objetivo é minimizar o makespan esperado. Para este problema, avaliamos o desempenho de duas heurísticas construtivas: uma heurística determinística aplicada aos tempos médios de processamento e uma heurística especificamente projetada para o problema estocástico. Usamos simulação para estimar o makespan esperado. Em geral, a heurística estocástica apresenta um desempenho melhor que o da heurística determinística aplicada aos tempos médios de processamento. No entanto, essa diferença não justifica o custo computacional da heurística estocástica, pelo menos dentro dos limites dos nossos experimentos. Portanto, uma maneira mais prática de lidar com o problema estocástico é resolver a contraparte determinística, uma vez que ela fornece boas sequências em tempo computacional razoável.

PALAVRAS CHAVE. Sequenciamento do backlog, Flow shop ConWiP estocástico, Otimização e Simulação.

Otimização Combinatória. Simulação

1. Introduction

Constant Work-in-Process (ConWiP) is a production control model, proposed by [Spearman et al. 1990; 2022] originally for flow shop production environments. According to the theory, we can describe the operation of a ConWiP flow shop can as follows: jobs are organized in the backlog and released to the line whenever a card is available. Otherwise, they remain in the backlog. When a job is completed at the last machine and leaves the line, the card is removed and sent back to the beginning of the line. A card release authorizes the entry of a new job, the first of the backlog if there is one. After entering the line, jobs follow the First Come, First Served (FCFS) rule, maintaining the same sequence. Thus, ConWiP monitors the work-in-process (WiP), keeping the level equal to or below a maximum limit, defined by the work-in-process capacity (WiP Cap), which is equal to the number of cards. Figure 1 illustrates a ConWiP flow shop with m machines, as considered in this article.

The two main decisions for a ConWiP flow shop are: (1) the card count, i.e., to determine the WiP Cap, and (2) job sequencing, i.e., how to sequence jobs in the backlog [Framinan et al. 2003]. Referring to the decisions taken to determine the WiP Cap, several authors have reported their contributions [Hopp and Roof 1998; Tardif and Maaseidvaag 2001; Framinan et al. 2006; Azouz and Pierreval 2019].

In this study, we focus on the backlog sequencing problem. The ConWiP backlog sequencing problem can be divided into two types: static and dynamic. The first considers that all orders are available at the beginning of the sequencing [Herer and Masin 1997; Golany et al. 1999; Luh et al. 2000; Framinan et al. 2001; Zhang and Chen 2001; Cao and Chen 2005; Ajorlou and Shams 2013; Aglan and Durmusoglu 2015]. The second admits that orders arrive randomly over time [Leu 2000; Framinan et al. 2000; Thürer et al. 2017; Bertolini et al. 2020; Diaz and Ardalan 2023].

In the ConWiP flow shop illustrated in Figure 1, the composition of the backlog is known at the beginning of the sequencing, so that the sequencing problem is static. Although some studies have already addressed the static and deterministic version of the backlog sequencing problem in flow shops controlled by ConWiP [Herer and Masin 1997; Zhang and Chen 2001; Framinan et al. 2001; Ajorlou and Shams 2013], the same cannot be said about its stochastic counterpart.

This article addresses the ConWiP flow shop backlog sequencing problem, assuming that the processing times are random variables whose probability distribution is known. As a result, the performance criterion is also a random variable, and the objective is to minimize its expected value. In the study, the objective is to find the sequence that minimizes the expected makespan i.e., expected total time to complete all jobs.

To do this, we adapted and tested two constructive heuristics that show reasonable solutions for permutational flow shop scheduling problem (PFSP) with random processing times in our problem. The first heuristic is a deterministic heuristic (applied to mean processing times), the second is a heuristic specifically designed for the stochastic problem. While heuristics were used to find good sequences, a simulation procedure was used to assess the quality of the solutions from estimates of the expected makespan. Both heuristics are tested on different problems and coefficients of variation.

The remainder of this paper is organized as follows: Section 2 presents the background. In Section 3, we describe the problem, the solution methods, and the testbed and design of experiments. Section 4 offers and discusses the results. Finally, we conclude the paper and points out future research opportunities in Section 5.

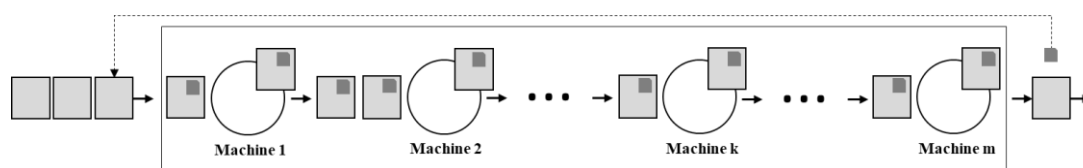


Figure 1: ConWiP flow shop with m machines.

2. Background

2.1 Backlog sequencing problem in ConWiP systems with deterministic processing times

One of the first published studies that addressed the static and deterministic version of the backlog sequencing problem on a ConWiP-based production system was [Herer and Masin 1997]. In this study, the authors propose a linear programming model to set the order of the backlog on a ConWiP flow shop. Its objective is to minimize the total cost, which includes the finished goods inventory holding cost, backorder cost, work-in-process inventory holding cost, and overtime cost. The authors do not propose an efficient algorithm to solve their model.

In the sequence, other studies have also addressed the backlog sequencing problem on ConWiP-based flow shops. [Zhang and Chen 2001] developed an integer nonlinear programming model to determine the backlog sequence and lot sizes. The goal is to minimize the set-up cost and the cost associated with the workload difference between adjacent batches at the bottleneck machines. Different from the model developed in [Herer and Masin 1997], [Zhang and Chen 2001] considered the lot sizes and unbalanced workload at the bottleneck machines in the development of their model.

[Framinan et al. 2001] evaluate the performance of well-known permutation flow shop scheduling problem (PFSP) heuristics when applied to the backlog-sequencing problem on a ConWiP-based flow shop, to minimize the makespan. The results of the experiment reveal that the NEH heuristic [Nawaz et al. 1983] is the most suitable among the tested heuristics. In this study, the authors also present a dispatching rule named "Centre Dispatching Rule" that is particularly suitable when the number of cards (or WiP Cap) is low.

[Ajlou and Shams 2013] present a nonlinear mixed integer programming model to find the optimal WiP level as well as backlog sequence simultaneously to minimize the makespan. An Artificial Bee Colony (ABC) metaheuristic is used to determine the backlog sequence, while the WiP level is maintained fixed. They run ABC varying the WiP level from 1 to the size of the backlog to find the optimal backlog sequence in each case. So, the backlog sequence with the lowest WiP level associated with the minimum makespan value is chosen. The results are compared with the non-traditional Genetic Algorithm (GA) approach. Although it finds the same solution, the computational time required to run GA is significantly higher.

In addition to the studies mentioned, the backlog sequencing problem has been addressed on different ConWiP-based production systems such as manufacturing cells [Golany et al. 1999], job shop [Luh et al. 2000], assembly [Cao and Chen 2005] and hybrid manufacturing system [Aglan and Durmusoglu 2015].

While the static and deterministic version of the backlog sequencing problem on ConWiP-based production systems has been studied under different additional assumptions, we cannot say the same about its stochastic counterpart. Despite this, there are reports of some advances for the stochastic flow shop scheduling problem (FSP). The next section presents a brief review of the literature on the stochastic version of FSP and PFSP when the processing times are random, and the objective is to minimize the expected makespan.

2.2 Stochastic version of the FSP and the PFSP with expected makespan objective

The stochastic flow shop scheduling problem (SFSP) consists of n jobs that must be processed by m machines in series, where the processing time of each job j on each machine k is not a constant value but is a random variable, P_{jk} , whose probability distribution is known. When we assume that the job sequence is the same on all machines, we have the stochastic permutation flow shop scheduling problem (SPFSP). Other assumptions generally assumed include all jobs and machines continuously available from instant zero, there are no set-up times, machines can process at most one job at a time, no preemption allowed, there is unlimited storage buffer between machines, etc., defined, for example, in [Pinedo 2016] and [Baker and Trietsch 2019].

In this literature, an important result is due to Talwar (1967). For the FSP with $m = 2$ and exponential processing times, [Talwar 1967] assumed that the expected makespan is minimized when jobs are sequenced according to the non-increasing order of the difference in

mean processing times ($\frac{1}{\mu_{j1}} - \frac{1}{\mu_{j2}}$, where μ_{ij} is the mean processing time of job j on machine k).

Later, [Cunningham and Dutta 1973] prove the optimality of this sequencing rule, currently referred to as Talwar's Rule. Extensions of the Talwar's rule have been presented by [Kamburowski 2000] to the case of three machines, [Kalczyński and Kamburowski 2004] and [Kalczyński and Kamburowski 2006] for the Gompertz and Weibull distribution, respectively.

Despite this advance, no optimal procedure is available for other probability distributions. Even in the case of three or more machines with exponential processing times, it is not possible to identify the optimal sequence with certainty. The main advantage of assuming the premise of exponential processing times is the possibility to calculate the expected makespan analytically. Therefore, in the remaining cases, we depend on heuristics to generate sequences and simulations to evaluate the quality of the sequences from estimates of expected makespan. By using simulations to estimate the expected makespan, this allows us solving the problem for any probability distribution [Portougal and Trietsch 2006; Baker and Altheimer 2012].

For the two-machines case and general distribution of the processing times, [Portougal and Trietsch 2006] developed and tested the heuristics named PDJ (Pseudo-Deterministic Johnson Rule) and PSH (Pair-Switching Heuristic). The first heuristic applies the Johnson rule to mean processing times, adding a tiebreaker criterion. The second heuristic starts with the sequence found by the PDJ heuristic and applies a neighborhood search based on the API movement (Adjacent Pairwise Interchange). [Baker and Trietsch 2011] compare Johnson's heuristic (Johnson's rule applied to mean processing times), Talwar's heuristic (Talwar's rule applied to mean processing times), and API heuristics. No heuristic dominates the others. Despite this, the authors recommend the use of Johnson's heuristic unless the coefficient of variation is very high, in which case Talwar's heuristic may be superior. The API heuristic is preferable when it is not clear which of the other two heuristics is more suitable.

For the general m machines case, [Baker and Trietsch 2012] suggest three constructive heuristics based on adaptations of the constructive heuristics CDS [Campbell et al. 1970] and NEH [Nawaz et al. 1983]. The authors conclude that the three heuristics have similar performances and mean sub-optimality below 1%. Later, [Framinan and Perez-González 2015] question the results presented by [Baker and Altheimer 2012], showing that, in many cases, significant differences were found in the performance of heuristics.

Additionally, [Framinan and Perez-González 2015] found that the performance of deterministic heuristics applied to mean processing times does not differ much from heuristics specifically designed for the stochastic problem and that these differences decrease as the variability in processing times increases (that is, when the coefficient of variation increases). This finding is aligned with the results presented in [Portougal and Trietsch 2006] and [Juan et al. 2014]. [Portougal and Trietsch 2006] showed that the PDJ heuristic (Johnson's rule applied to mean processing times) is asymptotically optimal under mild conditions, that is, as n grows, the relative difference between the makespan provided by the heuristic and the optimal makespan tends to zero. On the other side, [Juan et al. 2014] developed a simheuristic (an algorithm that combines an Iterated Local Search metaheuristic with Monte Carlo simulation) under the assumption that good sequences for the deterministic problem are probably good sequences for the stochastic problem. [Wang et al. 2015] present a two-stage simulation-based hybrid estimation of distribution algorithm (TSSB-HEDA) to the PFSP with stochastic processing times. The authors do not compare the performance of TSSB-HEDA with the performance of deterministic procedures. More recently, [González-Neira et al. 2019] combine different dispatching rules with Monte Carlo simulation to deal with the PFSP with stochastic processing times. Three probability distributions are considered (uniform, exponential and lognormal). In this study, experimental results were obtained for the expected mean and standard deviation of five indicators: makespan, flowtime, total tardiness, maximum tardiness, and number of tardy jobs.

3. Methodology

3.1 Problem description

The ConWiP flow shop backlog-sequencing problem consists of determining the sequence that n jobs must be processed on m machines in series, to minimize total time to complete all jobs (makespan), that corresponds to the completion time of the last job in the last machine ($C_{\max}$). It is assumed that the sequence of jobs must be the same on all machines and that the release of a job to the line must respect the work-in-process limit w . Other assumptions include those habitual of permutation flow shop scheduling problem such as all jobs and machines continuously available from instant zero, all processing times known with certainty, etc. The complete list of assumptions for the PSFP can be found in [Pinedo 2016] and [Baker and Trietsch 2019].

Therefore, defined a sequence of jobs $\pi = [j_1, j_2, \dots, j_n]$, the completion times of each job on each machine, is given by the following equations:

$$C_{11}(\pi) = p_{11} \quad (1)$$

$$C_{1k}(\pi) = C_{1,k-1} + p_{1k} \quad k = 2, \dots, m \quad (2)$$

$$C_{j1}(\pi) = C_{j-1,1} + p_{j1} \quad j = 2, \dots, \min(w, n) \quad (3)$$

$$C_{j1}(\pi) = \max\{C_{j-1,1}, C_{j-w,m}\} + p_{j1} \quad j = w + 1, \dots, n \quad (4)$$

$$C_{jk}(\pi) = \max\{C_{j,k-1}, C_{j-1,k}\} + p_{jk} \quad j = 2, \dots, n, k = 2, \dots, m \quad (5)$$

$$C_{\max}(\pi) = C_{n,m}(\pi) \quad (6)$$

where

- j : job position
- k : machine index
- n : number of jobs
- m : number of machines
- w : work-in-process limit
- p_{ij} : processing time of job in position j on machine k
- C_{jk} : completion time of job j on machine k

Equations (1) and (2) provide the completion times of the first job on the first and the next m machines. Equations (3) and (4) determine the completion times of the upcoming jobs on the first machine. Equation (3) applies to the first w jobs. For the next jobs, Equation (4) applies. This Equation determines that the job release is conditioned to two events: the first machine being free and the job $j - w$ being completed on machine m , to guarantee the restriction of a maximum of w jobs in the line. Equation (5) conditions a job start to its completion on the previous machine and the completion of the previous job on the same machine. This Equation does not apply to the first job and the first machine. Finally, Equation (6) explains that the makespan is equal to the completion time of the last job on the last machine.

In this article, the object of study consists of the stochastic counterpart of this problem, assuming that the processing times are independent and identically distributed random variables with a known probability distribution. Thus, the makespan is also a random variable, and the objective becomes to minimize its expected value ($E[C_{\max}]$).

3.2 Solution methods

For the FSP, when processing times are random, and the objective is to minimize the expected makespan, analytical solution procedures are available only under the assumption of exponential processing times. For the rest of the cases, heuristic methods are used to generate good candidate sequences that can be compared by simulation. In this case, simulation is used to estimate the expected makespan of the generated sequences. This section describes the tested solution methods to address our problem.

To solve the problem defined in Section 3.1, we tested the performance of two constructive heuristics. Both heuristics are based on the NEH heuristic, a constructive heuristic proposed by [Nawaz et al. 1983] for PFSP. The NEH heuristic consists of two phases. In the first phase, jobs are ordered according to the non-increasing values of the sum of their processing times. In the second phase, the first job of the initial order is placed in a partial sequence, initially without any work. Following the initial ordering, the heuristic tests the insertion of the next job in all possible positions of the partial sequence. The position that minimizes the makespan is assigned to the job. This procedure is repeated until a complete sequence is obtained.

The first heuristic (NEH-ConWiP) is an adaptation of the NEH heuristic. Basically, during the construction phase, evaluation of partial sequences considers the limit of WiP, as expressed in Equation (4). This procedure is applied using the mean processing times. Therefore, a sequence is generated as if the problem were deterministic. So, to evaluate the performance of this sequence when applied to the stochastic problem (that is, when the processing times are random), we estimate the expected makespan by using simulations.

The second heuristic (SNEH-ConWiP) is based on the stochastic version of the NEH heuristic [Baker and Altheimer 2012] considering the limit of WiP, as expressed in Equation (4). Unlike NEH-ConWiP, in the construction phase of the sequence, this heuristic to evaluate each insertion based on the expected makespan. For this purpose, it uses simulation to estimate the expected makespan of each partial sequence. The partial sequence that obtains the lowest value of the expected makespan estimate is maintained, and this procedure is repeated until a complete sequence is obtained. The simulations make the computational cost of the heuristic high, which can be prohibitive for large instances.

In both heuristic procedures, we use simulation to estimate the expected makespan. Basically, the expected makespan is calculated through an R number of simulation runs, using the sequence generated by the heuristic method as a solution. For this sequence, a sample of makespan values is obtained at the end of the simulation run ($C_{\max}^I$, $I = 1, \dots, R$) and the expected makespan is estimated by the sample mean ($E[C_{\max}] \approx \bar{C}_{\max} = \frac{1}{R} \sum_{I=1}^R C_{\max}^I$) [Framinan and Perez-González 2015]. Figure 2 illustrates the flowchart of the simulation procedure.

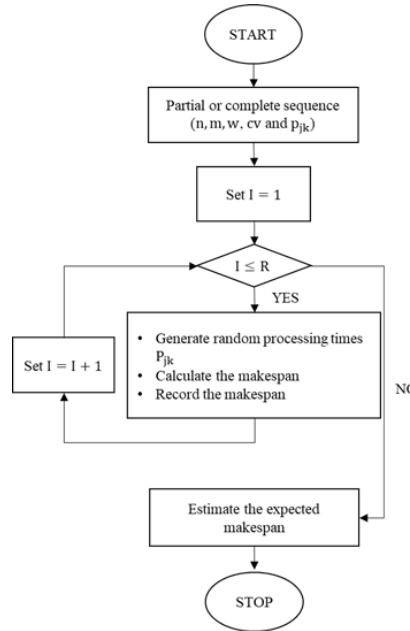


Figure 2: Flowchart of the simulation procedure.

To reduce the computational time required by the SNEH-ConWiP heuristic, we use shorter simulations (r) during the sequence construction phase, to quickly obtain an estimate of the

expected makespan of the partial and complete sequences. In the end, we make an R number of simulation runs ($R > r$) to generate a more accurate estimate of the expected makespan of the best complete sequence found by the heuristic. Other studies also apply shorter simulations in the evaluation of sequences [Gourgand et al. 2003; Juan et al. 2014; Wang et al. 2015].

The literature is not unanimous about the number of simulations runs required to generate accurate estimates of the expected makespan. In general, they use a large number to guarantee the significance of the estimation [Framinan and Perez-González 2015]. For example, [Gourgand et al. 2003] use $R = 200.000$. [Portougal and Trietsch 2006] use $R = 10.000$. [Baker and Trietsch 2011] and [Baker and Altheimer 2012] define $R = 100.000$. More recently, [Framinan and Perez-González 2015] proposed a procedure with a variable number of simulations runs, ensuring that the percentage error in the estimation of the expected makespan is bounded (with high probability) to a small percentage. We use $r = 3000$ and $R = 10,000$ considering the computational time, mainly for the SNEH-ConWiP heuristic. Other studies like [Baker and Altheimer 2012] and [Framinan and Perez-González 2015], who assessed the performance of the SNEH heuristic, are restricted to problems with $n \in \{10\}$ and $m \in \{2,3,6\}$ and $n \in \{5,10,15,20\}$ and $m \in \{2,5,10,20\}$, respectively.

The two heuristics and the simulation procedure were implemented in Python.

3.3 Testbed and design of experiments

The two heuristics described in Section 3.2 are applied to the testbed of [Taillard 1993], which contains 120 flow shop scheduling instances with 20, 50, 100, 200, and 500 jobs and 5, 10, or 20 machines. In total, there are 12 sets of 10 instances, and these are: 20×5 , 20×10 , 20×20 , 50×5 , 50×10 , 50×20 , 100×5 , 100×10 , 100×20 , 200×10 , 200×2 , and 500×20 . Of the 120 instances, only the instances with $n \in \{20,50\}$ and $m \in \{5,10\}$ were selected. We excluded the instances of 20 machines because a single-loop ConWiP is not suitable for long lines. To work well, ConWiP requires that the loop should not be too long. For long lines, a multi-loop ConWiP is more appropriate [Hopp and Spearman 2011]. Additionally, instances with 100 and 200 jobs were not considered in our experimentation due to the amount of computational time required by the SNEH-ConWiP heuristic. Regarding the values for the WiP limits, we use $w \in \{5,10,15\}$. The objective is to create instances of the problem with more restrictive, moderate, and higher WiP limits.

To model processing times, we assume that they follow a log-normal distribution, with mean p_{jk} and standard deviation $cv \times p_{jk}$, where cv is the common coefficient of variation for all processing times. This way, the processing times given by the instances of Taillard are used as the average values of the random processing times. The log-normal distribution has been widely used for several reasons: First, the log-normal variate is nonnegative; second, it is characterized by two parameters: means and standard deviation. This feature allows controlling the variability of processing times, defining different levels of coefficient of variation; and, finally, it has considerable practical value for modeling processing times of real manufacturing environments [Baker and Trietsch 2011; Baker and Altheimer 2012; Juan et al. 2014; Framinan and Perez-González 2015]. To represent different scenarios of uncertainty, we define $cv \in \{0.1, 0.25, 0.5\}$. So, when $cv=0.1$, we represent processing times with low variability, but when $cv=0.50$, the processing times will show high variability. As a result, we have a total of 360 instances (12 problem sizes ($n \times m \times w$) $\times 10$ problems of each size $\times 3$ coefficients of variation).

4. Results and discussion

Tables 1 to 3 show the results obtained by the two heuristics described in Section 3.2 for different coefficients of variation (cv). Note that for each problem size ($n \times m \times w$), we have averaged the ten corresponding instances since there are 120 instances for each coefficient of variation. Thus, Tables 1 to 3 present the average values of the expected makespan obtained by each heuristic (labeled as $\overline{E[C_{\max}]}$ in the Tables), as well as the average percentage difference of the expected makespan obtained by the heuristic NEH-ConWiP concerning that one obtained by the heuristic SNEH-ConWiP (labeled as $\text{dif}\%$ in the Tables). For each instance, the percentage difference is calculated from the Equation (7).

$$\text{dif}\% = \left(\frac{E[C_{\max}]_{(\text{NEH-ConWiP})} - E[C_{\max}]_{(\text{SNEH-ConWiP})}}{E[C_{\max}]_{(\text{SNEH-ConWiP})}} \right) \times 100 \quad (7)$$

In these Tables (1 to 3), we do not provide information about computational time demanded by each heuristic. Despite that, the difference is quite significant. For more complicated instances, for example, $n = 50$, $m = 10$, $w = 5$, $cv = 0.5$, the heuristic NEH-ConWiP needs, on average, 26 seconds to solve one instance, while for the heuristic SNEH-ConWiP an average of 5216 seconds is necessary. In total, it was required a computational time equivalent to 12 days to accomplish our experimentation.

Concerning the results obtained, observe that the differences between the heuristic NEH-ConWiP and the heuristic SNEH-ConWiP are small, in some cases being inferior to 1%. These differences seem to decrease when the variation coefficient increases. Additionally, even not being possible to guarantee a pattern, they also seem to fall when the number of machines increases.

For a better analysis of the results, we have performed the Analysis of Variance (ANOVA) to statistically evaluate the influence of the factors of the experiment in the average values of the expected makespan. The main results of the ANOVA are shown in Table 4.

 Table 1: Comparative results of the heuristics NEH-ConWiP and SNEH-ConWiP for $cv = 0.10$.

	n	m	w	cv	NEH-ConWiP	SNEH-ConWiP	dif%
					$E[C_{\max}]$	$E[C_{\max}]$	
1	20	5	5	0.10	1396.654	1384.107	0.916
2	20	5	10	0.10	1281.193	1263.484	1.443
3	20	5	15	0.10	1281.134	1264.514	1.359
4	20	10	5	0.10	2331.945	2316.460	0.702
5	20	10	10	0.10	1664.549	1660.183	0.281
6	20	10	15	0.10	1609.901	1604.117	0.345
7	50	5	5	0.10	3248.772	3223.164	0.796
8	50	5	10	0.10	2806.317	2786.023	0.742
9	50	5	15	0.10	2790.580	2772.631	0.651
10	50	10	5	0.10	5468.169	5431.728	0.666
11	50	10	10	0.10	3624.899	3596.115	0.791
12	50	10	15	0.10	3244.604	3208.195	1.145
Average					2562.393	2542.560	0.820

 Table 2: Comparative results of the heuristics NEH-ConWiP and SNEH-ConWiP for $cv = 0.25$.

	n	m	w	cv	NEH-ConWiP	SNEH-ConWiP	dif%
					$E[C_{\max}]$	$E[C_{\max}]$	
1	20	5	5	0.25	1484.717	1473.573	0.745
2	20	5	10	0.25	1338.840	1318.860	1.525
3	20	5	15	0.25	1338.161	1315.543	1.729
4	20	10	5	0.25	2437.059	2425.161	0.515
5	20	10	10	0.25	1761.940	1744.485	1.045
6	20	10	15	0.25	1701.002	1684.557	0.980
7	50	5	5	0.25	3462.317	3414.980	1.388
8	50	5	10	0.25	2931.764	2889.386	1.478
9	50	5	15	0.25	2884.313	2839.113	1.595
10	50	10	5	0.25	5741.497	5680.927	1.067
11	50	10	10	0.25	3863.657	3822.912	1.069
12	50	10	15	0.25	3442.060	3395.081	1.381
Average					2698.944	2667.048	1.210

Table 3: Comparative results of the heuristics NEH-ConWiP and SNEH-ConWiP for $cv = 0.50$.

	n	m	w	cv	NEH-ConWiP	SNEH-ConWiP	dif%
					$E[C_{max}]$	$E[C_{max}]$	
1	20	5	5	0.50	1664.861	1653.172	0.710
2	20	5	10	0.50	1464.233	1433.237	2.174
3	20	5	15	0.50	1461.053	1432.901	1.968
4	20	10	5	0.50	2678.583	2645.436	1.283
5	20	10	10	0.50	1979.595	1962.891	0.872
6	20	10	15	0.50	1900.689	1881.169	1.066
7	50	5	5	0.50	3906.205	3895.059	0.287
8	50	5	10	0.50	3219.913	3186.546	1.046
9	50	5	15	0.50	3095.815	3040.232	1.827
10	50	10	5	0.50	6344.871	6290.070	0.871
11	50	10	10	0.50	4386.093	4377.690	0.196
12	50	10	15	0.50	3883.642	3849.730	0.873
Average					2998.796	2970.678	1.098

Table 4: ANOVA results.

		Sum of squares	df ^a	Mean squares	F value	p-Value
$E[C_{max}]$	n	77719504	1	77719504	3259.9913	0.0000
	m	16053947	1	16053947	673.3924	0.0000
	w	12916822	2	6458411	270.9019	0.0000
	cv	2359455	2	1179728	49.4844	0.0000
	Heuristic	12751	1	12751	0.5349	0.4684
	n x m	2176854	1	2176854	91.3095	0.0000
	n x w	3459228	2	1729614	72.5497	0.0000
	n x cv	384978	2	192489	8.0741	0.0010
	n x Heuristic	1546	1	1546	0.0649	0.8001
	m x w	4400327	2	2200163	92.2872	0.0000
	m x cv	133143	2	66571	2.7924	0.0719
	m x Heuristic	4	1	4	0.0002	0.9894
	w x cv	75440	4	18860	0.7911	0.5371
	w x Heuristic	95	2	47	0.0020	0.9980
	cv x Heuristic	457	2	228	0.0096	0.9905
	Residuals	1072818	45	23840		

^a Degrees of freedom.

These results display that the number of jobs, the number of machines, the WiP limit, and the coefficient of variation influence the average values of the expected makespan. The number of jobs is the factor with greater influence, followed by the number of machines and WiP levels. The influence of the coefficient of variation may have been low because the chosen values for the cv are relatively close. Figures 3 to 6 illustrate the effect of the number of jobs, the number of machines, WiP limits, and the coefficient of variation within the average values of the expected makespan, respectively. The strongest interaction is between the number of machines and the WiP limit. When the number of machines grows, and the WiP limit decreases, this takes us to the worst results of the average values of the expected makespan, as it is illustrated in Figure 7. In summary, the average values of the expected makespan increase to the extent that n, m , and cv increase, and decrease when w increases.

On the one side, the results of ANOVA reinforce that there are no statistically significant differences between the heuristics NEH-ConWiP and SNEH-ConWiP, at least for the configurations of our experimentation. This fact is illustrated in Figure 8. Observe that data behavior for both heuristics is very similar.

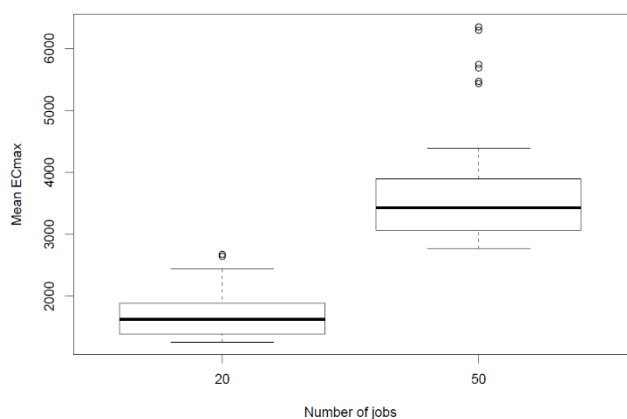


Figure 3: Effect of the number of jobs on the average values of the expected makespan.

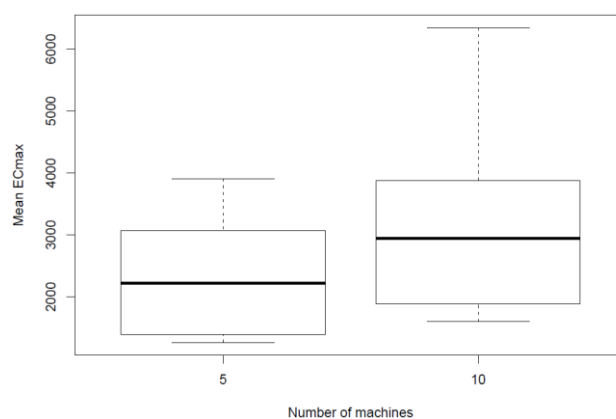


Figure 4: Effect of the number of machines on the average values of the expected makespan.

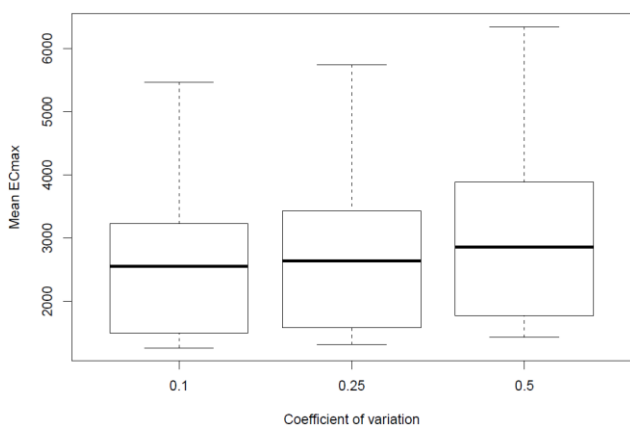


Figure 5: Effect of the coefficient of variation on the average values of the expected makespan.

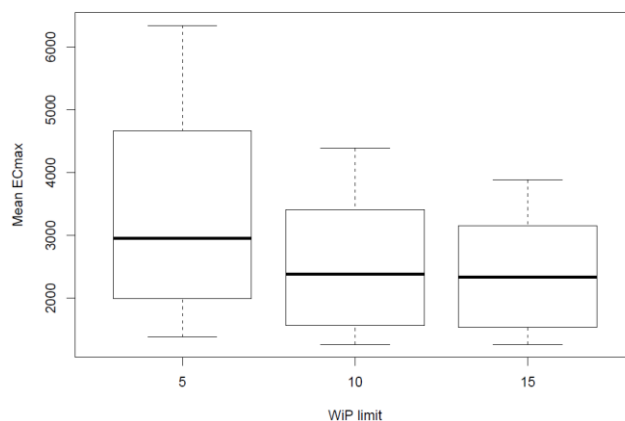


Figure 6: Effect of the WiP limit on the average values of the expected makespan.

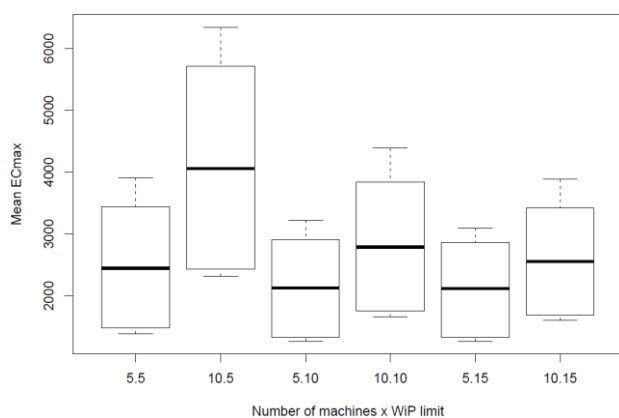


Figure 7: Effect of the interaction between the number of machines and WiP limit on the average values of the expected makespan.

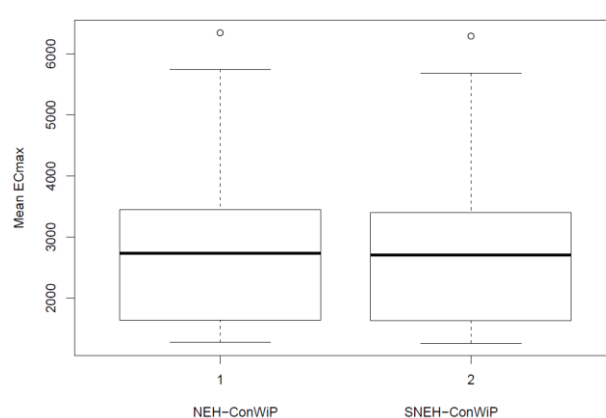


Figure 8: Effect of the heuristics on the average values of the expected makespan.

In general, heuristic SNEH-ConWiP is better than heuristic NEH-ConWiP. However, the difference between both is not sufficient to justify the quantity of computational time demanded by heuristic SNEH-ConWiP. Thus, our results suggest that a more practical way of solving ConWiP flow shop backlog sequencing problem, when the processing times are assumed to be random, is to apply heuristic NEH-ConWiP to the mean processing times. For it provides good solutions to our stochastic problem and in a reasonable time, at least within the limits of our research. Our results are aligned with the findings of Portougal e Trietsch (2006), Juan et al. (2014) e Framinan and Perez-González (2015) for PFSP with random processing times.

5. Conclusions

This article presents some exploratory results for the ConWiP flow shop backlog sequencing problem when the processing times are random, and the objective is to minimize the expected makespan. For so, first, we adapt the constructive heuristics NEH and SNEH, which offer suitable solutions for PFSP with random processing times, to our problem. It's done by inserting the work-in-process constraint in the makespan calculation. Whereas the first heuristic is deterministic (applied to the mean processing times), the second is a heuristic specifically designed for the stochastic problem. Then, we compare their performances. To do this, use a simulation procedure to estimate the expected makespan. Both heuristics are tested on different problem sizes and coefficients of variation.

The results revealed, despite the preliminary nature of this research, that there are no statistically significant differences between the heuristics. There is also no statistical evidence of the number of jobs, the number of machines, the WiP limits, and coefficients of variation affect heuristics performance. However, the difference between the heuristics apparently decreases when the number of machines and the coefficient of variation increases. Therefore, even if the heuristic SNEH-ConWiP performs better than the NEH-ConWiP, this difference does not justify the SNEH-ConWiP computational cost, at least within the limits of our experiments.

Regarding future work, we plan to consider the higher coefficients of variation for the log-normal distribution, to verify whether the performance between the two heuristics remains similar, as well as whether the differences between them decrease. Also, it may be interesting to introduce smaller instances and different probability distributions, for example, the exponential distribution, where, at least for the FSP, analytical solution procedures are available. Finally, more time can be spent on analyzing the performance of other deterministic methods, known for providing robust solutions for PFSP, when applied to our problem.

Referências

- Aglan, C. and M. B. Durmusoglu. (2015). Lot-splitting approach of a hybrid manufacturing system under CONWIP production control: a mathematical model. *International Journal of Production Research*, 53(5):1561–1583.
- Ajorlou S. and I. Shams. (2013). Artificial bee colony algorithm for CONWIP production control system in a multi-product multi-machine manufacturing environment. *Journal of Intelligent Manufacturing*, 24(6):1145-1156.
- Azouz, N., & Pierreval, H. (2019). Adaptive smart card-based pull control systems in context-aware manufacturing systems: training a neural network through multi-objective simulation optimization. *Applied Soft Computing*, 75:46-57.
- Baker, K. and D. Altheimer. (2012). Heuristic solution methods for the stochastic flowshop problem. *European Journal of Operational Research*, 216(1):172–177.
- Baker, K. R. and D. Trietsch. (2011). Three heuristic procedures for the stochastic, two-machine flow shop problem. *Journal of Scheduling*, 14:445-454.
- Baker, K. R., & Trietsch, D. (2019). Principles of sequencing and scheduling. 2nd ed. John Wiley & Sons, Hoboken, NJ.
- Bertolini, M., Braglia, M., Frosolini, M., & Marrazzini, L. (2020). Work in next queue CONWIP. *Computers & Industrial Engineering*, 143:106437.
- Campbell, H. G. R. A. Dudek, and M. L. Smith. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management science*, 16(10):B630-B637.
- Cao, D. and M. Chen. (2005). A mixed integer programming model for a two line CONWIP-based production and assembly system. *International Journal of Production Economics*, 95(3):317–326.

- Cunningham, A. A. and S. K. Dutta. (1973). Scheduling jobs, with exponentially distributed processing times, on two machines of a flow shop. *Naval Research Logistics Quarterly*, 20(1):69-81.
- Diaz, R., & Ardalan, A. (2023). Innovating in data-driven production environments: simulation analysis of Net-CONWIP priority rule. *Industrial Management & Data Systems*, 123(5):1569-1598.
- Framinan, J. M. and P. Perez-Gonzalez. (2015). On heuristic solutions for the stochastic flowshop scheduling problem. *European Journal of Operational Research*, 246(2):413–420.
- Framinan, J. M., P. L. Gonzalez, and R. Ruiz-Usano. (2003). The CONWIP production control system: review and research issues. *Production Planning & Control*, 14(3):255–265.
- Framinan, J. M., P. L. Gonzalez, and R. Ruiz-Usano. (2006). Dynamic card controlling in a CONWIP system. *International Journal of Production Economics*, 99(1-2):102–116.
- Framinan, J. M., R. Ruiz-Usano, and R. Leisten. (2001). Sequencing CONWIP flow-shops: analysis and heuristics. *International Journal of Production Research*, 39(12):2735–2749.
- Framinan, J. M., Ruiz-Usano, R., & Leisten, R. (2000). Input control and dispatching rules in a dynamic CONWIP flow-shop. *International Journal of Production Research*, 38(18):4589-4598.
- Golany, B., E. M. Dar-El, and N. Zeev. (1999). Controlling shop floor operations in a multi-family, multi-cell manufacturing environment through Constant Work-In-Process. *IIE Transactions*, 31(8):771–781.
- Gourgand, M. N. Grangeon, and S. Norre. (2003). A contribution to the stochastic flow shop scheduling problem. *European Journal of Operational Research*, 151:415–433.
- Herer, Y. T. and M. Chen. (1997). Mathematical programming formulation of CONWIP based production lines; and relationships to MRP. *International Journal of Production Research*, 35(4):1067–1076.
- Hopp, W. J. and M. L. Roof. (1998). Setting WIP levels with Statistical Throughput Control (STC) in CONWIP production lines. *International Journal of Production Research*, 36(4):867–882.
- Hopp, W. J. and M. L. Spearman. (2011). *Factory Physics*. 3rd ed. Waveland Press Inc, Long Grove, IL.
- Juan, A. A., B.B Barrios, E. Vallada, D. Riera, and J. Jorba. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46:101-117.
- Kalczynski, P. J. and J. Kamburowski. (2004). Generalization of Johnson's and Talwar's scheduling rules in two-machine stochastic flow shops. *Journal of the Operational Research Society*, 55(12):1358-1362.
- Kalczynski, P. J. and J. Kamburowski. (2006). A heuristic for minimizing the expected makespan in two-machine flow shops with consistent coefficients of variation. *European Journal of Operational Research*, 169(3):742-750.
- Kamburowski, J. (2000). On three-machine flow shops with random job processing times. *European Journal of Operational Research*, 125(2):440-448.
- Leu, B-Y. (2000). Generating a backlog list for a CONWIP production line: a simulation study. *Production Planning & Control*, 11(4):409–418.
- Luh, P. B., X. Zhou, and R. N. Tomastik. (2000). An effective method to reduce inventory in job shops. *IEEE Transactions on Robotics and Automation*, 16(4):420–424.
- Nawaz, M; E. E. Ensore Jr, and I. Ham. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91-95.
- Pinedo, M. L. (2016). *Scheduling: theory, algorithms, and systems*. 5th ed. Springer, New York, NY.
- Portougal, V. and D. Trietsch. (2006). Johnson's problem with stochastic processing times and optimal service level. *European Journal of Operational Research*, 169:751–760.
- Spearman, M. L., D. L. Woodruff, and W. J. Hoop. (1990). CONWIP: a pull alternative to Kanban. *International Journal of Production Research*, 28(5):879–894.
- Spearman, M. L., Woodruff, D. L., & Hopp, W. J. (2022). CONWIP Redux: reflections on 30 years of development and implementation. *International Journal of Production Research*, 60(1), 381-387.
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278-285.
- Talwar, P. (1967). A note on sequencing problems with uncertain job times. *Journal of the Operations Research Society of Japan*, (9):93–97.
- Tardif, V. and Maaseidvaag, L. (2001). An adaptive approach to controlling Kanban systems. *European Journal of Operational Research*, 132(2):411–424.
- Thürer, M., N. O. Fernandes, M. Stevenson, and T. Qu. (2017). On the backlog-sequencing decision for extending the applicability of ConWIP to high-variety contexts: an assessment by simulation. *International Journal of Production Research*, 55(16):4695–4711.
- Wang, K. S. H. Choi, and H. Lu. (2015). A hybrid estimation of distribution algorithm for simulation-based scheduling in a stochastic permutation flowshop. *Computers & Industrial Engineering*, 90:186-196.
- Zhang, W. and M. Chen. (2001). A mathematical programming model for production planning using CONWIP. *International Journal of Production Research*, 39(12):2723-2734.