

Article

Linking Scheduling Criteria to Shop Floor Performance in Permutation Flowshops

Jose M. Framinan ^{1,*}  and Rainer Leisten ²

¹ Industrial Management, School of Engineering, University of Seville, 41004 Sevilla, Spain

² Industrial Engineering, Faculty of Engineering, University of Duisburg-Essen, 47057 Duisburg, Germany; leisten@uni-due.de

* Correspondence: framinan@us.es

Received: 30 October 2019; Accepted: 4 December 2019; Published: 7 December 2019



Abstract: The goal of manufacturing scheduling is to allocate a set of jobs to the machines in the shop so these jobs are processed according to a given criterion (or set of criteria). Such criteria are based on properties of the jobs to be scheduled (e.g., their completion times, due dates); so it is not clear how these (short-term) criteria impact on (long-term) shop floor performance measures. In this paper, we analyse the connection between the usual scheduling criteria employed as objectives in flowshop scheduling (e.g., makespan or idle time), and customary shop floor performance measures (e.g., work-in-process and throughput). Two of these linkages can be theoretically predicted (i.e., makespan and throughput as well as completion time and average cycle time), and the other such relationships should be discovered on a numerical/empirical basis. In order to do so, we set up an experimental analysis consisting in finding optimal (or good) schedules under several scheduling criteria, and then computing how these schedules perform in terms of the different shop floor performance measures for several instance sizes and for different structures of processing times. Results indicate that makespan only performs well with respect to throughput, and that one formulation of idle times obtains nearly as good results as makespan, while outperforming it in terms of average cycle time and work in process. Similarly, minimisation of completion time seems to be quite balanced in terms of shop floor performance, although it does not aim exactly at work-in-process minimisation, as some literature suggests. Finally, the experiments show that some of the existing scheduling criteria are poorly related to the shop floor performance measures under consideration. These results may help to better understand the impact of scheduling on flowshop performance, so scheduling research may be more geared towards shop floor performance, which is sometimes suggested as a cause for the lack of applicability of some scheduling models in manufacturing.

Keywords: scheduling; shop floor performance; flowshop; manufacturing

1. Introduction

To handle the complexity of manufacturing decisions, these have been traditionally addressed in a hierarchical manner, in which the overall problem is decomposed into a number of sub-problems or decision levels [1]. Given a decision level, pertinent decisions are taken according to specific local criteria. It is clear that, for this scheme to work efficiently, the decisions among levels should be aligned to contribute to the performance of the whole system. Among the different decisions involved in manufacturing, here we focus on scheduling decisions. Scheduling (some authors use the term “detailed scheduling”) is addressed usually after medium-term production planning decisions have been considered, since production planning decision models do not usually make distinction between products within a family, and do not take into account sequence-dependent costs, or detailed machine capacity [2]. A short-term detailed scheduling model usually assumes that there are several jobs—each

one with its own characteristics—that have to be scheduled so one or more *scheduling criteria* are minimised. The schedule is then released to the shop floor, so the events in the shop floor are executed according to the sequence and timing suggested by the schedule [3]. Therefore, there is a clear impact of the chosen scheduling criteria on (medium/long term) shop floor performance, which is eventually reflected on shop floor performance measures such as the throughput of the system (number of jobs dispatched by time unit), cycle time (average time that the jobs spend in the manufacturing system), or work in process. As these performance measures can be linked to key aspects of the competitiveness of the company (e.g., throughput is related to capacity and resource utilisation, while cycle time and work in process are related to lead times and inventory holding costs), the chosen scheduling criterion may have an important impact in the performance of the company, so it is important to assess the impact of different scheduling criteria on shop floor performance measures. However, perhaps for historical reasons, the connection between shop floor performance measures and scheduling criteria has been neglected by the literature since, to the best of our knowledge, there are not contributions addressing this topic. In general, the lack of understanding and quantification of these connections has led to a number of interrelated issues:

- Some widely employed scheduling criteria have been subject of criticism due to their apparent lack of applicability to real-world situations (see, e.g., the early comments in [4] on Johnson’s famous paper, or [5] and [1] on the lack of real-life application of makespan minimisation algorithms), which suggest a poor alignment of these criteria with the companies’ goals.
- Some justifications for using specific scheduling criteria are given without a formal proof. For instance, it is usual in the scheduling literature to mention that minimising the completion time in a flowshop leads to minimising work-in-process, whereas this statement—as we discuss in Section 2.2—is not correct from a theoretical point of view.
- Some scheduling criteria employed in manufacturing have been borrowed from other areas. For instance, the minimisation of the completion time variance is taken from the computer scheduling context; therefore their potential advantages on manufacturing have to be tested.
- There are different formulations for some scheduling criteria intuitively linked to shop floor performance: While machine idle time minimisation can be seen, at least approximately, as related to increasing the utilisation of the system, there are alternative, non-equivalent, manners to formulate idle time. Therefore, it remains an open question to know which formulation is actually better in terms of effectively increasing the utilisation of the system.
- Finally, since it is customary that different, conflicting goals have to be balanced in the shop floor (such as balancing work in process, and throughput), it would be interesting to know the contribution of the different scheduling criteria to shop floor performance in order to properly balance them.

Note that, in two cases, the linkages between scheduling criteria and shop floor performance measures can be theoretically established. More specifically, it can be formally proved that makespan minimisation implies maximising the throughput, and that completion time minimisation implies the minimising the average cycle time. However, for the rest of the cases such relationships cannot be theoretically proved, so they have to be tested via experimentation. To do so, in this paper we carry out an extensive computational study under a different variety of scheduling criteria, shop floor performance measures, and instance parameters.

Since the mathematical expression of the scheduling criteria is layout-dependent, we have to focus on a particular production environment. More specifically, in this paper we assume a flow shop layout where individual jobs are not committed to a specific due date. The main reason for the choice is that flow line environments are probably the most common setting in repetitive manufacturing. Regarding not considering individual due dates for jobs, it should be mentioned that both scheduling criteria and shop floor performance measures differ greatly from due date related settings to non due date related ones, and therefore this aspect must be subject of a separate analysis. Finally, we also assume that all jobs to be scheduled are known in advance.

The results of the experiments carried out in this paper show that

1. There are several scheduling criteria (most notably the completion time variance and one definition of idle time) which are poorly related with any of the indicators considered for shop floor performance.
2. Makespan minimisation is heavily oriented towards increasing throughput, but it yields poor results in terms of average completion time and work-in-process. This confines its suitability to manufacturing scenarios with very high utilisation costs as compared to those associated with cycle time and inventory.
3. Minimisation of one definition of idle times results in sequences with only a marginal worsening in terms of throughput, but a substantial improvement in terms of cycle time and inventory. Therefore, this criterion emerges as an interesting one when the alignment with shop floor performance is sought.
4. Minimisation of completion times also provides quite balanced schedules in terms of shop floor performance measures; note that it does not lead to the minimisation of WIP, as recurrently stated in the literature.

The rest of the paper is organised as follows: In the next section, the scheduling criteria and shop floor performance measures to be employed in the experimentation are discussed, as well as the theoretically provable linkages among them. The methodology adopted in the computational experience is presented in Section 3.2. The results are discussed in Section 4. Finally, Section 5 is devoted to outline the main conclusions and to highlight areas for future research.

2. Background and Related Work

In this section, we first present the usual scheduling criteria employed in the literature, while in Section 2.2 we discuss the usual shop floor performance measures, together with the relationship with the scheduling criteria that can be formally proved. For the sake of brevity, we keep the detailed explanations on both criteria and performance measures at minimum, so the interested reader is referred to the references given for formal definitions.

2.1. Scheduling Criteria

Undoubtedly, the most widely employed scheduling criterion is the makespan minimisation (usually denoted as C_{max}) or maximum flow time (see, e.g., [6] for a recent review on research in flowshop sequencing with makespan objective). Another important measure is the (total or average) total completion time or $\sum C_j$. Although less employed in scheduling research than makespan, total completion time has also received a lot of attention, particularly during the last years. Just to mention a few recent papers, we note the contributions in [7,8].

An objective also considered in the literature is the minimisation of machine idle time, which can be defined in (at least) three different ways [9]:

- The idle time, as well as the head and tail, of every single machine, i.e., the time before the first job is started on a machine and the time after the last job is finished on a machine, but the whole schedule has started on the first machine and has not been finished yet on the last machine, can be included into the idle time or not. In a static environment, including all heads and tails means that idle time minimisation is equivalent to minimisation of makespan (see, e.g., in [4]). This case would not have to be considered further.
- Excluding heads and tails would give an idle time within the schedule, implicitly assuming that the machines could be used for other tasks/jobs outside the current problem before and after the current schedule passes/has passed the machine. This definition of idle time is also known as “core idle time” (see, e.g., in [10–12]) and it has been used by [13] and by [14] in the context of a multicriteria problem. We denote this definition of idle time as $\sum IT_j$.

- Including machine heads in the idle time computation whereas the tails are not included means that the machines are reserved for the schedule before the first job of the schedule arrives but are released for other jobs outside the schedule as soon as the last job has left the current machine. In the following, we denote this definition as $\sum ITH_j$. This definition is first encountered in [15] and in [16] and it has been used recently as a secondary criterion for the development of tie-breaking rules for makespan minimisation algorithms (see, e.g., [17,18]).

Figure 1 illustrates these differences in idle time computation for an example of two jobs on three machines. The light grey time-periods (IT and Head) are included in our idle time definition whereas the Tail is not. In the literature, an equivalent expression for heads and tails are Front Delay and Back Delay, respectively, see in [19] or [9].

M1	p(1,1)	p(2,1)		
M2	,Head‘	p(1,2)	IT	p(2,2)
M3		p(1,3)		IT
				,Tail‘
				p(2,3)

Figure 1. Different components of machine idle time.

Finally, the last criterion under consideration is the Completion Time Variance (CTV). CTV was originally introduced by [20] in the computer scheduling context, where it is desirable to organise the data files in on-line computing systems so that the file access times are as uniform as possible. It has been subsequently applied in the manufacturing scheduling context as it is stated to be an appropriate objective for just-in-time production systems, or any other situation where a uniform treatment of the jobs is desirable (see, e.g., in [21–24]). In the flow shop/job shop scheduling context, it has been employed by [25–32].

2.2. Shop Floor Performance Measures

Shop floor performance is usually measured using different indicators. Among classical texts, Goldratt [33] mentions throughput, inventory, and operating expenses as key manufacturing performance measures. Nahmias [34] mentions the following manufacturing objectives: meet due dates, minimise WIP, minimise cycle time, and achieve a high resource utilisation. Wiendahl [35] identifies four main objectives in the production process: short lead times, low schedule deviation, low inventories, and high utilisation. Hopp and Spearman [1] list the following manufacturing objectives: high throughput, low inventory, high utilisation, short cycle times, and high product variety. Li et al. [36] cites utilisation and work-in-process as the two main managerial concerns in manufacturing systems. Finally, throughput and lateness are identified by several authors (e.g., [37,38]) as the main performance indicators in manufacturing.

Although these objectives have remained the same during decades [39], their relative importance has changed across time [40], and also depends on the specific manufacturing sector (for instance, in the semiconductor industry, average cycle time is regarded as the most important objective, see, e.g., [41] or [42]). According to the references reviewed above, we consider three performance measures: Throughput (TH), Work-In-Process (WIP), and Average Cycle Time (ACT) as shop floor performance indicators. With respect to other indicators mentioned in the reviewed references, note that one of them is not relevant in the deterministic environment to which this analysis is constrained (low schedule deviation), while other is not specifically related to shop floor operation (high product variety). Furthermore, as our study does not assume individual due dates for jobs, we exclude due date related measures, although we wish to note that, quite often short cycle times are employed as an

indicator of due date adherence [38,43]. Finally, we prove below that utilisation and throughput are directly related, so utilisation does not need to be considered in addition to throughput.

Regarding the relationship of the shop floor performance measures with the scheduling criteria, it is easy to check that TH the throughput may be defined in terms of $C_{max}(S)$ the makespan of a sequence S of n jobs, i.e.,

$$TH(S) = \frac{n}{C_{max}(S)} \quad (1)$$

As a result, throughput is inversely proportional to makespan. Note that the utilisation $U(S)$ can be defined as (see, e.g., [36]):

$$U(S) = \frac{\sum_i \sum_j p_{ij}}{C_{max}(S)} \quad (2)$$

therefore, it is clear that $U(S) = \frac{\sum_i \sum_j p_{ij}}{n} \cdot TH(S)$, and, as $\frac{\sum_i \sum_j p_{ij}}{n}$ is constant for a given instance, then it can be seen that the two indicators are fully related.

Accordingly, ACT average cycle time can be expressed in terms of the completion time, see, e.g., [44]:

$$ACT(S) = \frac{\sum C_j(S)}{n} \quad (3)$$

It follows that the total completion time is proportional to ACT . Since TH , ACT and WIP are linked through Little's law, the following equation holds.

$$WIP(S) = TH(S) \cdot ACT(S) = \frac{\sum C_j(S)}{C_{max}(S)} \quad (4)$$

From Equation (4), it may be seen that total completion time and WIP minimisation are not exactly equivalent, although it is a common statement in the scheduling flowshop literature: It is easy to show that the two criteria are equivalent for the single-machine case, but this does not necessarily hold for the flowshop case.

As, apart from the two theoretical equivalences above discussed, there are no straightforward relationship between the scheduling criteria and the shop floor performance measures, such relationships should be empirically discovered over a high number of problem instances. This computational experience must take into account that the results might be possibly influenced by the instance sizes and the processing times employed. The methodology to carry out the experimentation is described in the next section.

3. Computational Experience

The following approach is adopted to assess how the minimisation of a certain scheduling criterion impacts on the different shop floor indicators:

1. Build a number of scheduling instances of different sizes and with different mechanisms for generating the processing times. The procedure to build these test-beds is described in Section 3.1.
2. For each one of these instances, find the sequences optimising each one of the scheduling criteria under consideration. For small-sized instances, the optimal solutions can be found, while for the biggest instances, a good solution found by a heuristic approach is employed. The procedure for this step is described in Section 3.2.
3. For each one of these five optimal (or good) sequences, compute their corresponding values of TH , WIP , and ACT . This can be done in a straightforward manner according to Equations (1)–(4).
4. Analyse the so-obtained results. This is carried out in Section 4.

3.1. Testbed Setting

Although, in principle, a possible option to obtain flowshop instances to perform our research may be to extract these data from real-life settings, this option poses a number of difficulties. First, obtaining such data is a representative number is complicated. There are only few references publishing real data in the literature (see [45,46]). It may be thus required to obtain such data from primary sources, which may be a research project itself. Second, processing time data are highly industry-dependent, and it is likely that a sector-by-sector analysis would be required, which in turn makes the analysis even more complicate and increases the need of obtaining additional data. Finally, extracting these data from industry would make processing times to be external (independent) variables in the analysis.

Therefore, we generate these data according to test-bed generation methods available in the literature. For the flowshop layout in our research, this means establishing the problem size (number of jobs and machines) and processing times of each job on each machine.

With respect to the values of the number of jobs n and m machines, we have chosen the following: $n \in \{20, 50, 100, 200\}$, and $m \in \{10, 20, 50\}$. For each problem size, 30 instances have been generated. This number has been chosen so that the results have a relatively high statistical significance.

Regarding the generation of the processing times, methods for generating processing times can be classified in random and correlated. In random methods, processing times are assumed to be independent from the jobs and the machines, i.e., they are generated by sampling them from a random interval using a uniform distribution $[a, b]$. The most usual values for this interval are $[1,99]$ (see, e.g., in [47,48]), while in some other cases even wider intervals are employed (e.g., [49] uses $[1,200]$). Random methods intend to produce difficult problem instances, as it is known that, at least with respect to certain scheduling criteria, this generation method yields the most difficult problems [50,51]. As foreseeable, random processing times are not found in practice [52]. Instead of random processing times, in real-life manufacturing environments it is encountered a mixture of job-correlation and machine-correlation for the processing times, as some surveys suggest (e.g., [53]). To model this correlation, several methods have been proposed, such as those of [54–56], or [57]. Among these, the latest method synthesises the others. This method allows obtaining problem instances with mixed correlation between jobs and machines. The amplitude of the interval from which the distribution means of the processing times are uniformly sampled depends on a parameter $\alpha \in [0, 1]$. For low values of α , differences among the processing times in the machines are small, while the opposite occurs for large values of α . For a detailed description of the implementation, the reader is referred to [57].

Finally, it is to note that several works claim the Erlang distribution to better capture the distribution of processing times (e.g., [4,19], or [58]), yet these do not specify whether this has been confirmed in real-life settings. Therefore, we discard this approach.

In [57], the processing times for each job on each machine p_{ij} are generated according to the following steps.

1. Set the upper and lower bounds of processing times, Dur_{LB} and Dur_{UB} , respectively, and a factor α controlling the correlation of the processing times.
2. Obtain the value $Interval_{st}$ by drawing a uniform sample from the interval $[Dur_{LB}, Dur_{UB} + Width_{eff}]$, where $Width_{eff} = rint(\alpha \cdot (Dur_{UB} - Dur_{LB}))$.
3. For each machine j , obtain $D_j = [d_j^{lb}, d_j^{ub}] = [\mu_j - d_j^{hw}, \mu_j + d_j^{hw}]$, where μ_j is sampled from the interval $[Interval_{st}, Interval_{st} + Width_{eff}]$ and d_j^{hw} is uniformly sampled from the interval $[1, 5]$.
4. For each job i , a real value $rank_i$ is uniformly sampled from the interval $[0, 1]$. Then, the processing times p_{ij} are obtained in the following manner: $p_{ij} = rint(rank_i \cdot (d_j^{ub} - d_j^{lb})) + d_j^{lb} + \eta$, where η is a 'noise factor' obtained by uniformly sampling from the interval $[-2, 2]$.
5. p_{ij} are ensured to be within the upper and lower bounds, i.e. if $p_{ij} < Dur_{LB}$, then $p_{ij} = Dur_{LB}$. Analogously, if $p_{ij} > Dur_{UB}$, then $p_{ij} = Dur_{UB}$.

The parameter α controls the degree of correlation, so for the case $\alpha = 0.0$, there is no correlation among jobs and machines. In our research, we consider four different ways to generate processing times:

- LC (Medium Correlation): Processing times are drawn according to the procedure described above and $\alpha = 0.1$.
- MC (Medium Correlation): Processing times are drawn according to the procedure described above and $\alpha = 0.5$.
- HC (High Correlation): Processing times are drawn according to the procedure described above and $\alpha = 0.9$.
- NC (No Correlation): Processing times are drawn from a uniform distribution [1,99]. This represents the “classical” noncorrelated assumption in many scheduling papers.

3.2. Optimisation of Scheduling Criteria

For each one of the problem instances, the sequences minimising each one of the considered scheduling criteria are obtained. For small problem sizes (i.e., $n \in \{5, 10\}$), this has been done by exhaustive search. As for bigger problem sizes, using exhaustive search or any other exact method is not feasible in view of the NP-hardness of these decision problems, we have found the best sequence (with respect to each of the scheduling criteria considered) by using an efficient metaheuristic, which is allowed a long CPU time interval. More specifically, we have built a tabu search algorithm (see, e.g., [59]). The basic outline of the algorithm is as follows.

- The neighbourhood definition includes the sum of the general pairwise interchange and insertion neighbourhoods. Both neighbourhood definitions are widely used in the literature.
- The size of the tabu list L has been set to the maximum value between the number of jobs and the number of machines, i.e., $L = \max n, m$. As the size of the list is used to avoid getting trapped into local optima, the idea is keeping a list size related to the size of the neighbourhood.
- As stopping criterion, the algorithm terminates after a number of iterations without improvement. This number has been set as the minimum of $10 \cdot n$. This ensures a large minimum number of iterations, while increasing this number of iterations with the problem size.

4. Computational Results

4.1. Dominance Relationships among Scheduling Criteria

A first goal of the experiments is to establish which scheduling criterion is more related to the different shop floor performance measures. To check the statistical significance of the results, we test a number of hypotheses using a one-sided test for the differences of means of paired samples (see, e.g., [60]) for every combination of m and n . More specifically, for each pair of scheduling criteria (A, B) and a shop floor performance measure ζ , we would like to know whether the sequence resulting from the minimisation of scheduling criteria A yields a better value for ζ , denoted as $\zeta(A)$, than the sequence resulting from the minimisation of scheduling criteria B . More specifically, we want to establish the significance of the null hypothesis $H_0 : \zeta(A) \text{ better than } \zeta(B)$ to determine whether criterion A is more aligned with SF indicator ζ than criterion B , or vice versa. Note that *better than* may express different ordinal relations depending on the performance measure, i.e., it is better to have a higher TH , but it is better to have lower ACT and WIP , therefore we specifically test the following three hypotheses for every combination of scheduling criteria A and B :

$$H_0 : TH(A) > TH(B)$$

$$H_1 : TH(A) \leq TH(B)$$

with respect to throughput, and

$$H_0 : WIP(A) \leq WIP(B)$$

$$H_1 : WIP(A) > WIP(B)$$

and

$$H_0 : ACT(A) \leq ACT(B)$$

$$H_1 : ACT(A) > ACT(B)$$

with respect to average completion time and work in process, respectively.

The results for each pair of scheduling criteria (A, B) are shown in Tables 1–10 for the different testbeds, where p -values are given as the maximum level of significance to reject H_0 (p represents the limit value to reject hypothesis H_0 resulting from a t -test, i.e., for every level of significance $\alpha \leq p$, H_0 would have to be rejected, whereas for every $\alpha > p$, H_0 would not be rejected. A high p indicates that H_0 can be rejected with high level of significance, and therefore H_1 can be accepted.) To express it in an informal way: a value close to zero in the column corresponding to the performance measure ζ in the table comparing the pair of scheduling criteria (A, B) indicates that minimizing criterion A leads to better values of ζ than minimizing criterion B , whereas a high value indicates the opposite.

To make an example of the interpretation of the procedure adopted, let us take the column TH for any of the testbeds in Table 1 (all zeros). This column shows the p -values obtained by testing the null hypothesis that makespan minimisation produces solutions with higher throughput than those produced by using flowtime minimisation as a scheduling criterion. Since these p -values are zero for all problem sizes, then the null hypothesis cannot be rejected. As a consequence, we can be quite confident (statistically speaking) that makespan minimisation is more aligned with throughput increase than completion time minimisation.

In view of the results of the tables, the following comments can be done.

- Regarding Table 1, it is clear that makespan outperforms the total completion time regarding throughput, and that the total completion time outperforms the makespan regarding average cycle time. These results are known from theory and, although they could have been omitted, we include them for symmetry. The table also shows that completion time outperforms makespan with respect to work in process, a result that cannot be theoretically predicted. This result is obtained for all instance sizes and different methods to generate the processing times. As a consequence, if shop floor performance is measured using primarily one indicator, C_{max} would be the most aligned objective with respect to throughput, whereas $\sum C_j$ would be the most aligned with respect to cycle time and work in process.
- From Table 2, it can be seen that makespan outperforms $\sum ITH_j$ with respect to throughput, and, in general, with respect to ACT (with the exception of small problem instances for certain processing times' generation). Finally, regarding work in process, in general, makespan outperforms $\sum ITH_j$ if $n > m$, whereas the opposite occurs if $m \geq n$.
- Tables 3 and 4 show an interesting result: despite the problem size and/or the distribution of the processing times, makespan outperforms both $\sum IT_j$ and CTV for all three shop floor performance measures considered. This result reveals that the minimisation of CTV or $\sum IT_j$ are poorly linked to shop floor performance, as least compared to makespan minimisation.
- Table 5 show that, regardless the generation of processing times and/or the problem size, completion time performs worse than $\sum ITH_j$ for makespan, whereas it outperforms it in terms of average cycle time and work in process.
- Table 6 show that, with few exception cases, the completion time outperforms $\sum IT_j$ for all three SF indicators.
- In Table 7, a peculiar pattern can be observed: while it can be that $\sum C_j$ dominates CTV with respect to the three SF indicators, this is not the case for the random processing times, as in this case the makespan values obtained by CTV are higher than those observed for the total completion time.
- In Tables 8 and 9 it can be seen that $\sum ITH_j$ outperforms both $\sum IT_j$ and CTV for all instance sizes and all generation of the processing times. Regarding considering the heads or not in the idle time

Table 3. Cont.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP									
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4. *p*-values for rejecting the hypotheses H_0 regarding the pair (C_{max}^*, CTV^*) for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP									
5	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 5. *p*-values for rejecting the hypotheses H_0 regarding the pair $(\sum C_j^*, ITH^*)$ for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP									
5	5	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
5	10	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
10	5	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
10	10	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
20	10	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
20	20	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
20	50	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
50	10	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
50	20	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
50	50	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0
		100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	0.0

Table 6. *p*-values for rejecting the hypotheses H_0 regarding the pair $(\sum C_j^*, ITH^*)$ for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP
5	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	5	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
10	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	50	95.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.9	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.6	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
50	50	98.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		29.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49.1	0.0	0.0

Table 7. Maximum level of significance for rejecting the hypotheses H_0 regarding the pair $(\sum C_j^*, CTV^*)$ for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP
5	5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.4	0.0	0.0
5	10	99.9	0.0	0.0	27.5	0.0	0.0	93.6	0.0	0.0	100.0	0.0	0.0
10	5	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.4	0.0	0.0
10	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	50	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
		30.0	0.0	0.0	2.8	0.0	0.0	9.4	0.0	0.0	79.7	0.0	0.0

Table 8. *p*-values for rejecting the hypotheses H_0 regarding the pair (ITH^*, IT^*) for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP
5	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	5	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
10	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	50	95.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.9	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.6	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
50	50	98.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		29.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49.1	0.0	0.0

Table 9. *p*-values for rejecting the hypotheses H_0 regarding the pair (ITH_j^*, CTV^*) for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP
5	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	5	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
10	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
20	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	50	95.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.9	0.0	0.0
50	10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.6	0.0	0.0
50	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
50	50	98.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		29.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49.1	0.0	0.0

Table 10. *p*-values for rejecting the hypotheses H_0 regarding the pair (IT^*, CTV^*) for different testbeds.

<i>n</i>	<i>m</i>	LC			MC			HC			NC		
		TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP	TH	ACT	WIP
5	5	100.0	100.0	100.0	100.00	100.0	67.97	7.29	0.03	0.00	100.0	100.0	100.0
5	10	100.0	100.0	0.0	100.0	100.00	0.00	100.0	100.0	0.0	100.0	100.0	0.0
10	5	0.0	99.96	100.0	6.96	100.0	100.0	0.0	100.0	100.0	7.94	100.0	100.0
10	10	100.0	100.0	100.0	100.0	100.00	98.09	100.00	100.0	100.0	100.0	100.0	100.0
20	10	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	100.0	100.0
20	20	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
20	50	100.0	100.0	64.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0
50	10	100.0	100.0	100.0	100.0	100.0	100.0	8.4	100.0	100.0	0.0	100.0	100.0
50	20	1.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	100.0	100.0
50	50	0.0	1.2	100.0	100.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0
		70.1	89.0	86.4	100.0	100.0	100.0	63.6	100.0	88.9	66.7	100.0	80.0

Table 11. Efficient criteria for each pair of SF indicators.

SF Indicators	Efficient Scheduling Criteria
(WIP, ACT)	$\sum C_j$
(WIP, TH)	$\sum C_j, \sum ITH_j, C_{max}$
(ACT, TH)	$\sum C_j, \sum ITH_j, C_{max}$

4.2. Ranking of Scheduling Criteria

In this section, we further try to explore the trade-off among the different criteria by answering the following question: Once we choose certain scheduling criterion according to the aforementioned ranking, how are the gains (or losses) that we can expect in the different shop floor performance measures when we switch from one scheduling criterion to another. More formally, we intend to quantify the difference between picking one scheduling criterion or another for a given shop floor performance measure. To address this issue, we define the RD_{PM} or Relative Deviation with respect to a given PM (performance measure) in the following manner.

$$RD(A)_{PM} = \frac{PM(S_A) - PM(S_{A^+})}{PM(S_{A^+})} \cdot 100 \tag{5}$$

where $PM(S_A)$ is the value of PM obtained for the sequence S_A which minimises scheduling criterion A . Analogously, S_{A^+} is the sequence obtained by minimising scheduling criterion A^+ , being A^+ the scheduling criterion ranking immediately behind A for the performance measurement PM . When A is the scheduling criterion ranking last for PM , then RD is set to zero.

Note that this definition of RD allows us to obtain more information than the mere rank of scheduling criteria. For instance, let us consider the scheduling criteria A , B , and C , which rank (ascending order) with respect to the performance measure PM in the following manner: B , C , A . This information (already obtained in Section 4.1) simply states that B (C) is more aligned that C (A) with respect to performance measure PM , but does not convey information on whether there are substantial differences between the three criteria for PM , or not. This information can be obtained by measuring the corresponding RD : If $RD(B)$ is zero or close to zero, it implies that B and C yield similar values for PM , and therefore there is not so much difference (with respect to PM) between minimizing B , or C . In contrast, a high value of $RD(C)$ indicates a great benefit (with respect to PM) when switching from minimizing A to minimizing C .

Since RD is defined for a specific instance, we use the Average Relative Deviation (ARD) for comparison across the testbed, consisting in averaging the RD s. The results of the experiments for the different testbeds with respect to ARD are shown in Tables 12–15, together with the rank of each criterion for each problem size. In addition, the cumulative ARD of the scheduling criteria for each shop floor performance measures are shown in Figure 2 for the different testbed. In view of the results, we give the following comments.

Table 12. Average Relative Deviation (ARD) and ranks (in parentheses) of the scheduling criteria for the random test-bed.

<i>n</i>	<i>m</i>	<i>TH</i>					<i>ACT</i>					<i>WIP</i>				
		<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>
5	5	2.10	3.68	6.03	0.00	0.42	0.55	13.44	4.39	0.00	0.96	2.52	6.34	9.24	0.00	4.74
		(1)	(4)	(2)	(5)	(3)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
5	10	2.79	1.89	1.48	0.00	1.50	2.33	9.88	1.24	0.64	0.00	1.51	5.85	5.38	0.00	2.66
		(1)	(4)	(2)	(5)	(3)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
10	5	1.82	0.00	8.51	0.13	0.37	4.98	18.48	0.54	0.00	1.43	1.24	10.03	12.90	0.00	1.00
		(1)	(5)	(2)	(3)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
10	10	5.13	1.30	2.66	0.00	2.36	2.17	13.59	2.71	0.00	0.88	2.26	7.20	6.87	0.00	4.37
		(1)	(4)	(2)	(5)	(3)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	10	3.77	0.00	5.21	0.45	2.17	2.51	19.88	0.97	0.00	3.47	2.68	11.21	8.72	0.00	2.21
		(1)	(5)	(2)	(3)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	20	5.71	0.67	1.93	0.00	1.92	1.53	14.62	1.01	0.00	1.36	4.47	7.21	4.14	0.00	3.83
		(1)	(4)	(2)	(5)	(3)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	50	4.79	0.00	3.27	0.13	0.02	0.28	10.66	1.83	0.30	0.00	2.84	5.20	2.36	0.00	2.90
		(1)	(5)	(3)	(4)	(2)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
50	10	1.70	1.87	5.61	0.11	0.00	1.95	23.84	2.92	0.00	4.02	3.75	18.67	9.11	0.00	2.03
		(1)	(4)	(2)	(3)	(5)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	20	5.11	0.45	3.49	0.66	0.00	0.53	17.96	3.41	0.00	0.18	5.56	10.31	4.26	0.00	2.37
		(1)	(4)	(2)	(3)	(5)	(2)	(1)	(4)	(5)	(3)	(2)	(1)	(3)	(5)	(4)
50	50	6.04	0.57	0.53	0.00	1.27	1.12	11.93	1.21	0.00	0.03	4.50	6.12	1.75	0.00	2.96
		(1)	(4)	(2)	(5)	(3)	(4)	(1)	(2)	(5)	(3)	(2)	(1)	(3)	(5)	(4)
		3.90	1.04	3.87	0.15	1.00	1.80	15.43	2.02	0.09	1.23	3.13	8.81	6.47	0.00	2.91

Table 13. Average Relative Deviation (ARD) and ranks (in parentheses) of the scheduling criteria for the LC test-bed.

<i>n</i>	<i>m</i>	<i>TH</i>					<i>ACT</i>					<i>WIP</i>				
		<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>
5	5	0.08	0.06	0.33	0.00	0.15	0.05	0.83	0.32	0.00	0.05	0.14	0.42	0.71	0.00	0.20
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
5	10	0.08	0.09	0.13	0.00	0.03	0.07	0.45	0.00	0.02	0.00	0.08	0.22	0.22	0.00	0.09
		(1)	(4)	(2)	(5)	(3)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
10	5	0.03	0.00	0.31	0.05	0.04	0.03	1.08	0.39	0.00	0.10	0.06	0.66	0.74	0.00	0.05
		(1)	(5)	(2)	(4)	(3)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
10	10	0.15	0.15	0.32	0.00	0.16	0.31	1.00	0.00	0.00	0.27	0.15	0.55	0.77	0.00	0.42
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	10	0.09	0.14	0.29	0.00	0.07	0.19	1.33	0.44	0.00	0.21	0.27	0.96	0.87	0.00	0.28
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	20	0.34	0.26	0.83	0.00	0.14	0.25	2.54	0.05	0.00	0.63	0.29	1.46	1.39	0.00	0.74
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	50	0.29	0.00	0.39	0.02	0.02	0.05	0.89	0.19	0.00	0.00	0.10	0.38	0.63	0.00	0.03
		(1)	(5)	(2)	(4)	(3)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	10	0.24	0.30	0.88	0.00	0.24	0.26	10.83	0.98	0.00	1.26	0.49	9.72	2.06	0.00	1.64
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	20	0.07	0.08	0.13	0.01	0.00	0.07	0.94	0.20	0.00	0.14	0.13	0.74	0.42	0.00	0.14
		(1)	(3)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	50	0.52	0.42	1.20	0.06	0.00	0.33	3.37	0.31	0.00	0.42	0.21	1.97	2.24	0.05	0.00
		(1)	(4)	(2)	(3)	(5)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(4)	(5)
		0.19	0.15	0.48	0.01	0.09	0.16	2.33	0.29	0.00	0.31	0.19	1.71	1.01	0.01	0.36

Table 14. Average Relative Deviation (ARD) and ranks (in parentheses) of the scheduling criteria for the MC test-bed.

<i>n</i>	<i>m</i>	<i>TH</i>					<i>ACT</i>					<i>WIP</i>				
		<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>
5	5	0.06	0.07	0.19	0.00	0.11	0.20	0.52	0.01	0.00	0.00	0.05	0.28	0.48	0.00	0.10
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
5	10	0.16	0.01	0.39	0.00	0.24	0.08	0.79	0.10	0.11	0.00	0.05	0.35	0.74	0.00	0.12
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
10	5	0.03	0.05	0.21	0.01	0.00	0.02	0.86	0.17	0.00	0.19	0.04	0.60	0.44	0.00	0.18
		(1)	(3)	(2)	(4)	(5)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
10	10	0.17	0.15	0.27	0.00	0.17	0.39	0.80	0.09	0.00	0.02	0.08	0.46	0.91	0.00	0.18
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	10	0.07	0.07	0.16	0.00	0.06	0.23	0.99	0.04	0.00	0.14	0.02	0.81	0.51	0.00	0.20
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	20	0.09	0.08	0.22	0.00	0.07	0.08	0.77	0.01	0.00	0.20	0.08	0.47	0.39	0.00	0.26
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	50	0.18	0.06	0.44	0.00	0.06	0.07	0.87	0.12	0.00	0.04	0.05	0.38	0.69	0.00	0.09
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	10	0.05	0.10	0.13	0.00	0.04	0.01	2.05	0.31	0.00	0.30	0.06	1.88	0.53	0.00	0.36
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	20	0.09	0.11	0.10	0.00	0.05	0.02	1.12	0.21	0.00	0.25	0.11	0.94	0.43	0.00	0.29
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	50	0.09	0.07	0.16	0.00	0.04	0.09	0.58	0.05	0.00	0.09	0.04	0.39	0.35	0.00	0.13
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
		0.10	0.08	0.23	0.00	0.08	0.12	0.94	0.11	0.01	0.12	0.06	0.66	0.55	0.00	0.19

Table 15. Average Relative Deviation (ARD) and ranks (in parentheses) of the scheduling criteria for the HC test-bed.

<i>n</i>	<i>m</i>	<i>TH</i>					<i>ACT</i>					<i>WIP</i>				
		<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>	<i>C_{max}</i>	ΣC_j	ΣITH_j	ΣIT_j	<i>CTV</i>
5	5	0.05	0.08	0.26	0.01	0.00	0.07	0.59	0.16	0.05	0.00	0.11	0.29	0.48	0.04	0.00
		(1)	(3)	(2)	(4)	(5)	(2)	(1)	(3)	(4)	(5)	(2)	(1)	(3)	(4)	(5)
5	10	0.16	0.24	0.20	0.00	0.02	0.12	0.52	0.02	0.03	0.00	0.12	0.18	0.38	0.00	0.16
		(1)	(4)	(2)	(5)	(3)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
10	5	0.01	0.06	0.35	0.05	0.00	0.03	0.99	0.33	0.00	0.14	0.03	0.67	0.66	0.00	0.12
		(1)	(3)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
10	10	0.08	0.14	0.20	0.00	0.09	0.01	0.61	0.22	0.00	0.13	0.09	0.31	0.56	0.00	0.22
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	10	0.07	0.08	0.17	0.00	0.08	0.02	0.97	0.26	0.00	0.24	0.07	0.72	0.50	0.00	0.28
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	20	0.11	0.07	0.18	0.00	0.12	0.00	0.70	0.11	0.00	0.19	0.11	0.42	0.35	0.00	0.30
		(1)	(3)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
20	50	0.11	0.11	0.20	0.00	0.05	0.01	0.43	0.07	0.00	0.07	0.03	0.20	0.37	0.00	0.12
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)
50	10	0.02	0.03	0.12	0.00	0.00	0.03	1.27	0.21	0.00	0.13	0.06	1.14	0.37	0.00	0.14
		(1)	(3)	(2)	(4)	(5)	(2)	(1)	(3)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	20	0.05	0.06	0.11	0.00	0.04	0.14	0.98	0.01	0.00	0.16	0.04	0.83	0.32	0.00	0.19
		(1)	(3)	(2)	(5)	(4)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
50	50	0.07	0.06	0.19	0.03	0.00	0.05	0.67	0.04	0.00	0.13	0.02	0.45	0.37	0.00	0.09
		(1)	(3)	(2)	(4)	(5)	(3)	(1)	(2)	(5)	(4)	(2)	(1)	(3)	(5)	(4)
		0.07	0.09	0.20	0.01	0.04	0.05	0.77	0.14	0.01	0.12	0.07	0.52	0.44	0.00	0.16

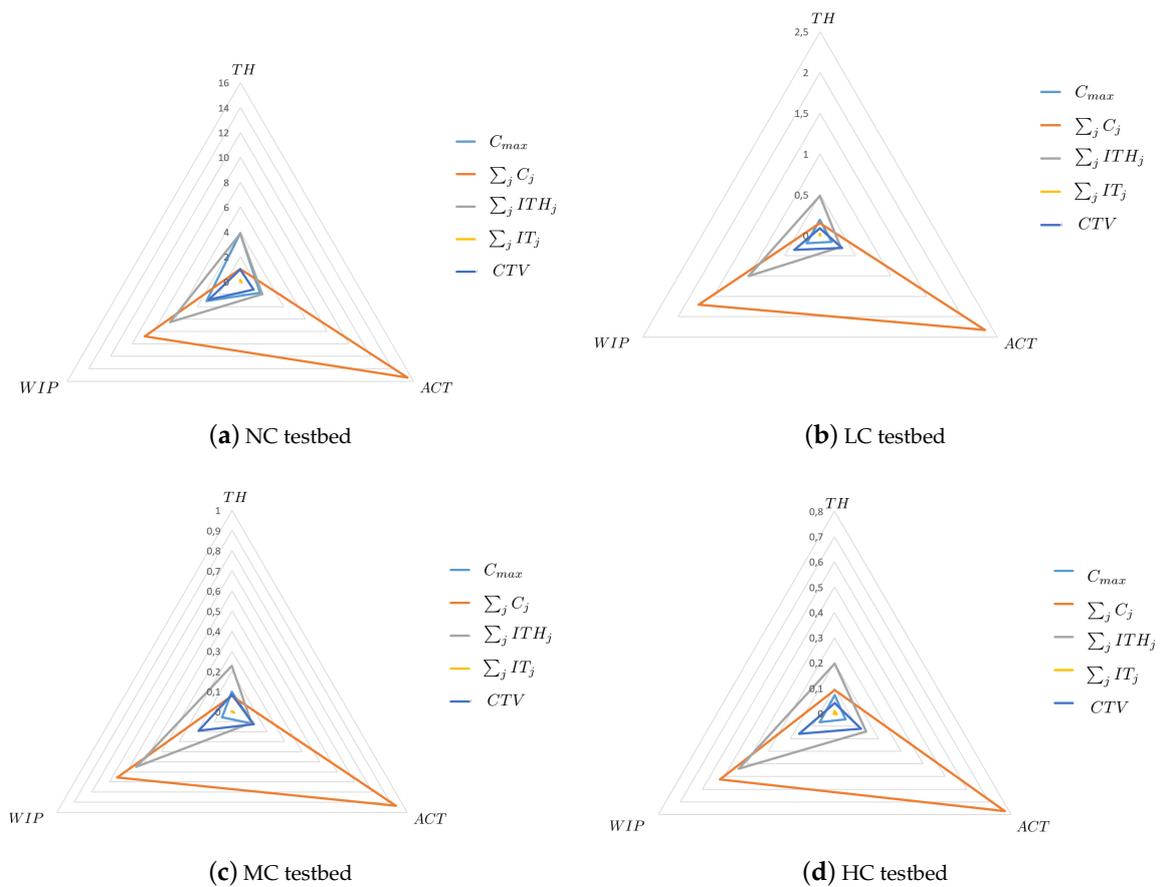


Figure 2. Relative performance of the criteria for the different testbeds.

- $\sum ITH_j$ emerges as an interesting criterion as its performance is only marginally worse than C_{max} with respect to TH —particularly in the NC testbed, see Figure 2a, but it obtains better values regarding ACT and WIP . Similarly, although it performs worse than $\sum C_j$ for ACT and WIP , it performs better in terms of throughput.
- The differences in ARD for throughput are, in general, smaller than those for ACT and WIP . For the correlated test-beds (LC to HC), the differences never reach 1%. This speaks for the little difference between minimising any of the scheduling measures if throughput maximisation is sought. The highest differences are encountered for the random test-bed (~6%).
- The differences in all measures for structured instances are smaller than for random test-bed. For instance, whereas makespan ranks first for TH (theoretically predictable), the maximum ARD for a given problem size in the random test-bed is 6.04%, whereas this is reduced to 0.52% for LC, and to 0.16% for HC. Analogously, the maximum differences between the completion time (ranking first for ACT) and the next criterion raise up to 23.84% for the random test-bed while dropping to 1.27% for HC. This means that the structured problems are easier than random problems because the distribution of the processing times flattens the objective functions, at least with respect to the considered shop floor performance measures.

5. Conclusions and Further Research

An extensive computational study has been carried out in order to analyse the links between several scheduling criteria in a flowshop and well-known shop floor performance measures. These results give some insights into the nature of these links, which can be summarised as follows.

- Roughly speaking, we could divide the considered scheduling criteria into two big categories: those tightly related to any (some) shop floor performance measure, and those poorly related to

SF performance. Among the later, we may classify CTV and $\sum IT_j$. Nevertheless, this is not meant to say that these criteria are not useful. However, from a shop floor performance perspective, it may be interesting to investigate whether these scheduling criteria relate to other performance measures. Perhaps extending the analysis to a due date scenario might yield some positive answer.

- Makespan matches (as theoretical predicted) throughput maximisation better than any other considered criteria. However, it turns out that differences between its minimisation and the minimisation of other criteria with respect to throughput are very small. Additionally, given the relatively poor performance of makespan with respect to ACT , one might ask whether makespan minimisation pays off for many manufacturing scenarios in terms of shop floor performance as compared, e.g., to completion time or $\sum ITH_j$ minimisation. A positive answer seems to be confined to these scenarios where costs associated to cycle time are almost irrelevant as compared to costs related to machine utilisation. The fact that this situation is not common in many manufacturing scenarios may lead to the lack of practical descriptions on the application of this criteria already discussed by [4].
- Completion time minimisation matches extremely well both work in process and average cycle time minimisation (the latter being theoretical predictable), better than any other criteria. In addition, the rest of the scheduling criteria perform much worse. Therefore, completion time minimisation emerges as a major criterion when it comes to increase shop floor performance. This empirical reasoning indicates the interest of the research on completion time minimisation rather than on other criteria, at least within the flowshop scheduling context.
- The minimisation of idle time (including the heads) performs better than completion time with respect to throughput. However, its performance is substantially worse than completion time regarding ACT and WIP . Hence, it seems an interesting criterion when throughput maximisation is the most important performance measure but work-in-process costs are not completely irrelevant.
- With respect to the influence of the test-bed design on the results, there are noticeable differences between the overall results obtained in the correlated test-beds (LC-HC), and those obtained from the random test-bed. In general, the introduction of structured processing times seems to reduce the differences between the scheduling criteria. At a first glance, this means that random processing times make it difficult to achieve a good shop floor performance by the application of a specific scheduling criterion. It is widely known that random problems produce difficult instances in the sense that there were high differences between bad and good schedules (with respect to a given scheduling criterion), at least for the makespan criterion. In view of the results of the experiments, we can also assert that these also translate into shop floor performance measures.

From these results, some aspects warrant future research:

- $\sum ITH_j$ emerges as an interesting scheduling criterion, with virtues in between makespan and completion time. For most of the problem settings, it compares to makespan in terms of cycle time, and it outperforms total completion time in terms of throughput. In view of these results, perhaps it is interesting devoting more efforts to flowshop minimisation with this criterion, which so far has been used only as a secondary tie-breaking rule. Interestingly, the results in this paper might suggest that its excellent performance in terms of tie-breaking rule is motivated by its alignment with shop floor performance.
- While it is possible to perfectly match the shop floor objectives of throughput and average cycle time with scheduling criteria (makespan and completion time, respectively), WIP cannot be linked to a scheduling criterion in a straightforward manner. Although the minimisation of completion time achieves the best results for WIP minimisation among the tested criteria, “true” work-in-process optimization is not the same as completion time minimisation. Here, the quotient between total completion time and makespan emerges as a “combined” scheduling criteria which may be worth of research as it matches an important shop floor performance measure such as work-in-process minimisation.

- The results of the present study are limited by the shop layout (i.e., the permutation flowshop) and the scheduling criteria (i.e., not due date-related criteria) considered. Therefore, an obvious extension of this study is to analyse other environments and scheduling measures. Particularly, the inclusion of due date related criteria could provide some additional insights on the linkage between these and the shop floor performance measures, as well as between the due date and non-due date scheduling criteria.

Author Contributions: Methodology, J.M.F. and R.L.; Writing—original draft, J.M.F.; Writing —review & editing, J.M.F. and R.L.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hopp, W.; Spearman, M. *Factory Physics. Foundations of Manufacturing Management*, 3rd ed.; Irwin: New York, NY, USA, 2008.
2. Framinan, J.; Leisten, R.; Ruiz, R. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1–400.
3. Aytug, H.; Lawley, M.A.; McKay, K.; Mohan, S.; Uzsoy, R. Executing production schedules in the face of uncertainties: A review and some future directions. *Eur. J. Oper. Res.* **2005**, *161*, 86–110. [[CrossRef](#)]
4. Conway, R.; Maxwell, W.L.; Miller, L.W. *Theory of Scheduling*; Dover: Mineola, NY, USA, 1967.
5. Dudek, R.A.; Panwalkar, S.S.; Smith, M.L. The Lessons of Flowshop Scheduling Research. *Oper. Res.* **1992**, *40*, 7–13, [[CrossRef](#)]
6. Fernandez-Viagas, V.; Ruiz, R.; Framinan, J. A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation. *Eur. J. Oper. Res.* **2017**, *257*, 707–721. [[CrossRef](#)]
7. Fernandez-Viagas, V.; Framinan, J. A beam-search-based constructive heuristic for the PFSP to minimise total flowtime. *Comput. Oper. Res.* **2017**, *81*, 167–177. [[CrossRef](#)]
8. Fernandez-Viagas, V.; Framinan, J. A new set of high-performing heuristics to minimise flowtime in permutation flowshops. *Comput. Oper. Res.* **2015**, *53*, 68–80. [[CrossRef](#)]
9. Framinan, J.; Leisten, R.; Rajendran, C. Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. *Int. J. Prod. Res.* **2003**, *41*, 121–148. [[CrossRef](#)]
10. Benkel, K.; Jørnsten, K.; Leisten, R. Variability aspects in flowshop scheduling systems. In Proceedings of the 2015 International Conference on Industrial Engineering and Systems Management (IESM), Seville, Spain, 21–23 October 2015; pp. 118–127.
11. Maassen, K.; Perez-Gonzalez, P.; Framinan, J.M. Relationship between common objective functions, idle time and waiting time in permutation flowshop scheduling. In Proceedings of the 29th European Conference on Operational Research (EURO 2018), Valencia, Spain, 8–11 July 2018.
12. Maassen, K.; Perez-Gonzalez, P. Diversity of processing times in permutation flow shop scheduling problems. In Proceedings of the 66th Operations Research Conference, Dresden, Germany, 3–6 September 2019.
13. Liao, C.J.; Tseng, C.T.; Luarn, P. A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput. Oper. Res.* **2007**, *34*, 3099–3111. [[CrossRef](#)]
14. Liu, W.; Jin, Y.; Price, M. A new Nawaz-Enscore-Ham-based heuristic for permutation flow-shop problems with bicriteria of makespan and machine idle time. *Eng. Optim.* **2016**, *48*, 1808–1822. [[CrossRef](#)]
15. Sridhar, J.; Rajendran, C. Scheduling in flowshop and cellular manufacturing systems with multiple objectives—a genetic algorithmic approach. *Prod. Plan. Control* **1996**, *7*, 374–382. [[CrossRef](#)]
16. Ho, J.; Chang, Y.L. A new heuristic for the n-job, M-machine flow-shop problem. *Eur. J. Oper. Res.* **1991**, *52*, 194–202. [[CrossRef](#)]
17. Fernandez-Viagas, V.; Framinan, J. On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. *Comput. Oper. Res.* **2014**, *45*, 60–67. [[CrossRef](#)]
18. Fernandez-Viagas, V.; Framinan, J. A best-of-breed iterated greedy for the permutation flowshop scheduling problem with makespan objective. *Comput. Oper. Res.* **2019**, *112*, 104767. [[CrossRef](#)]

19. King, J.; Spachis, A. Heuristics for flow-shop scheduling. *Int. J. Prod. Res.* **1980**, *18*, 345–357. [[CrossRef](#)]
20. Merten, A.; Muller, M. Variance minimization in single machine sequencing problems. *Manag. Sci.* **1972**, *18*, 518–528. [[CrossRef](#)]
21. Kanet, J.J. Minimizing variation of flow time in single machine systems. *Manag. Sci.* **1981**, *27*, 1453–1464. [[CrossRef](#)]
22. Baker, K.R.; Scudder, G.D. Sequencing with earliness and tardiness penalties. A review. *Oper. Res.* **1990**, *38*, 22–36. [[CrossRef](#)]
23. Gupta, M.; Gupta, Y.; Bector, C. Minimizing the flow-time variance in single-machine systems. *J. Oper. Res. Soc.* **1990**, *41*, 767–779. [[CrossRef](#)]
24. Cai, X.; Cheng, T. Multi-machine scheduling with variance minimization. *Discret. Appl. Math.* **1998**, *84*, 55–70. [[CrossRef](#)]
25. Cai, X. V-shape property for job sequences that minimize the expected completion time variance. *Eur. J. Oper. Res.* **1996**, *91*, 118–123. [[CrossRef](#)]
26. Marangos, C.; Govande, V.; Srinivasan, G.; Zimmers, E., Jr. Algorithms to minimize completion time variance in a two machine flowshop. *Comput. Ind. Eng.* **1998**, *35*, 101–104. [[CrossRef](#)]
27. Gowrishankar, K.; Rajendran, C.; Srinivasan, G. Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *Eur. J. Oper. Res.* **2001**, *132*, 643–665. [[CrossRef](#)]
28. Leisten, R.; Rajendran, C. Variability of completion time differences in permutation flow shop scheduling. *Comput. Oper. Res.* **2015**, *54*, 155–167. [[CrossRef](#)]
29. Ganesan, V.; Sivakumar, A.; Srinivasan, G. Hierarchical minimization of completion time variance and makespan in jobshops. *Comput. Oper. Res.* **2006**, *33*, 1345–1367. [[CrossRef](#)]
30. Gajpal, Y.; Rajendran, C. An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshops. *Int. J. Prod. Econ.* **2006**, *101*, 259–272. [[CrossRef](#)]
31. Krishnaraj, J.; Pugazhendhi, S.; Rajendran, C.; Thiagarajan, S. A modified ant-colony optimisation algorithm to minimise the completion time variance of jobs in flowshops. *Int. J. Prod. Res.* **2012**, *50*, 5698–5706. [[CrossRef](#)]
32. Krishnaraj, J.; Pugazhendhi, S.; Rajendran, C.; Thiagarajan, S. Simulated annealing algorithms to minimise the completion time variance of jobs in permutation flowshops. *Int. J. Ind. Syst. Eng.* **2019**, *31*, 425–451. [[CrossRef](#)]
33. Goldratt, E. *The Haystack Syndrome: Shifting Information out of the Data Ocean*; North River Press: Croton-on-Hudson, NY, USA, 1996.
34. Nahmias, S. *Production and Operations Analysis*; Irwin: Homewood, IL, USA, 1993.
35. Wiendahl, H.P. *Load-Oriented Manufacturing Control*; Springer: Berlin/Heidelberg, Germany, 1995.
36. Li, W.; Dai, H.; Zhang, D. The Relationship between Maximum Completion Time and Total Completion Time in Flowshop Production. *Procedia Manuf.* **2015**, *1*, 146–156. [[CrossRef](#)]
37. Land, M. Parameters and sensitivity in workload control. *Int. J. Prod. Econ.* **2006**, *104*, 625–638. [[CrossRef](#)]
38. Thürer, M.; Stevenson, M.; Land, M.; Fredendall, L. On the combined effect of due date setting, order release, and output control: An assessment by simulation. *Int. J. Prod. Res.* **2019**, *57*, 1741–1755. [[CrossRef](#)]
39. Land, M. Workload in Job Shop, Grasping the Tap. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 2004.
40. Wiendahl, H.P.; Glassner, J.; Petermann, D. Application of load-oriented manufacturing control in industry. *Prod. Plan. Control* **1992**, *3*, 118–129. [[CrossRef](#)]
41. Grewal, N.S.; Bruska, A.C.; Wulf, T.M.; Robinson, J.K. Integrating targeted cycle-time reduction into the capital planning process. In Proceedings of the 1998 Winter Simulation Conference, Washington, DC, USA, 13–16 December 1998; Volume 2, pp. 1005–1010.
42. Leachman, R.; Kang, J.; Lin, V. SLIM: Short cycle time and low inventory in manufacturing at Samsung electronics. *Interfaces* **2002**, *32*, 61–77. [[CrossRef](#)]
43. Sandell, R.; Srinivasan, K. Evaluation of lot release policies for semiconductor manufacturing systems. In Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, USA, 8–11 December 1996; pp. 1014–1022.
44. Abedini, A.; Li, W.; Badurdeen, F.; Jawahir, I. Sustainable production through balancing trade-offs among three metrics in flow shop scheduling. *Procedia CIRP* **2019**, *80*, 209–214. [[CrossRef](#)]

45. Bestwick, P.F.; Hastings, N. New bound for machine scheduling. *Oper. Res. Q.* **1976**, *27*, 479–487. [[CrossRef](#)]
46. Lahiri, S.; Rajendran, C.; Narendran, T. Evaluation of heuristics for scheduling in a flowshop: A case study. *Prod. Plan. Control* **1993**, *4*, 153–158. [[CrossRef](#)]
47. Taillard, E. Benchmarks for Basic Scheduling Problems. *Eur. J. Oper. Res.* **1993**, *64*, 278–285. [[CrossRef](#)]
48. Vallada, E.; Ruiz, R.; Framinan, J. New hard benchmark for flowshop scheduling problems minimising makespan. *Eur. J. Oper. Res.* **2015**, *240*, 666–677. [[CrossRef](#)]
49. Demirkol, E.; Mehta, S.; Uzsoy, R. Benchmarks for shop scheduling problems. *Eur. J. Oper. Res.* **1998**, *109*, 137–141. [[CrossRef](#)]
50. Campbell, H.G.; Dudek, R.A.; Smith, M.L. A Heuristic Algorithm for the n Job, m Machine Sequencing Problem. *Manag. Sci.* **1970**, *16*, B-630–B-637. [[CrossRef](#)]
51. Dannenbring, D.G. An evaluation of flowshop sequencing heuristics. *Manag. Sci.* **1977**, *23*, 1174–1182. [[CrossRef](#)]
52. Amar, A.D.; Gupta, J. Simulated versus real life data in testing the efficiency of scheduling algorithms. *IIE Trans.* **1986**, *18*, 16–25. [[CrossRef](#)]
53. Panwalkar, S.S.; Dudek, R.; Smith, M.L. Sequencing research and the industrial scheduling problem. In *Symposium on the Theory of Scheduling and Its Applications*; Springer: Berlin/Heidelberg, Germany, 1973; pp. 29–38.
54. Rinnooy Kan, A. *Machine Scheduling Problems*; Martinus Nijhoff: The Hague, The Netherlands, 1976.
55. Lageweg, B.; Lenstra, J.; Rinnooy Kan, A. A general bounding scheme for the permutation flow-shop problem. *Oper. Res.* **1978**, *26*, 53–67. [[CrossRef](#)]
56. Reeves, C. A genetic algorithm for flowshop sequencing. *Comput. Oper. Res.* **1995**, *22*, 5–13. [[CrossRef](#)]
57. Watson, J.P.; Barbulescu, L.; Whitley, L.; Howe, A. Contrasting structured and random permutation flow-shop scheduling problems: Search-space topology and algorithm performance. *INFORMS J. Comput.* **2002**, *14*, 98–123. [[CrossRef](#)]
58. Park, Y.; Pegden, C.; Ensore, E. A survey and evaluation of static flowshop scheduling heuristics. *Int. J. Prod. Res.* **1984**, *22*, 127–141. [[CrossRef](#)]
59. Hoos, H.H.; Stützle, T. *Stochastic Local Search: Foundations and Applications*; Elsevier: Amsterdam, The Netherlands, 2005.
60. Montgomery, D.C. *Design and Analysis of Experiments*; John Wiley & Sons: Hoboken, NJ, USA, 2006.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).