Robust cyclic flow shop scheduling problem in an uncertain environment

Abstract

Many production processes are being executed periodically, but due to the nature of the execution environment each of those might be executed in a slightly different way. In order to understand those differences and the environment better, we can collect historical data and use them later in algorithms designing. In the paper we consider cyclic flowshop problem in an uncertain environment with operations processing times, jobs due dates, penalties for jobs' delivery delays and machine setup times. Uncertainty is modelled by random variables with the normal distribution and we investigate variants where processing times and due dates are considered as uncertain parameters. In the problem the goal is to minimize the average cost of many executions for disturbed data. In our research we consider cost based on number of tardy jobs and cost based on total weighted tardiness. We present a set properties of the investigated problem which are applied to the appropriately redesigned tabu search method. The conducted computational experiments show that the proposed methods result in much more robust solutions than the ones obtained in the classic deterministic approach.

Introduction

Discrete optimization problems have been investigated for many years, however significant majority of this research is related to the deterministic models where all the parameters and known and determined in the moment where algorithm's execution is started. For these kind of problems which belong to the *NP-hard* problem class a lot of effective approximation algorithms has been designed and solutions established by those algorithms are very often close to the optimal ones. In practice, however, during the production process execution according to some established earlier schedule, it is quite common situation that the actual parameters are different than the ones assumed earlier. In consequence the execution time might be much worse than the expected one, moreover, the solution might be even no longer feasible. In such a situation we say that solution is not robust, because it is not able to survive in the uncertain environment. Unfortunately there are many scenarios where we are not able to predict the parameters due to the process nature (e.g. depended on the weather conditions), data may come from imprecise measuring tool and there are many more sources of uncertainty.

A response to that challenge is considering uncertainty during the algorithms design, but this approach introduces many problems and therefore those kind problems are studied less often. Recent results have been published in Bożejko et. al. (Bożejko et al. 2011), (Bożejko, Rajba, and Wodecki 2014), (Bożejko, Rajba, and Wodecki 2017) and Rajba i Wodecki (Rajba and Wodecki 2012).

In the paper we consider cyclic flowshop problem with processing times, machine setup times, due dates and penalties in case of delay. Each job needs to be executed on all machines in sequence (technological line) and the sequence is the same for all machines. As we investigate the cyclic version, once all jobs are execute on the machine, a new sequence is being started again and so on. The goal is to minimize the average cycle cost calculated based on some number of cycles (e.g. 100) and we consider two variants where we minimize the sum of weights of tardy jobs and total weighted tardiness.

Scheduling on multiple machines have been investigated for many years. In the beginning problems with number of tardy jobs were considered, i.e. problem with criteria $\sum U_i$. In Lenstra et. al. (Lenstra, Kan, and Brucker 1977) from 1977 it was proved that even for two machines the problem is *NP-hard*. Hariri and Potts (Hariri and Potts 1989) investigated flowshop variant $F||\sum U_i$ and it was one of the first papers on this problem. They presented an optimal algorithm based on the branch and bound method which is able to solve a problem instance of size less than 100 jobs and several machines. Genetic algorithms presented in Bartel and Billau (Bertel and Billaut 2004) oraz Ucar and Tasgetiren (Ucar and Tasgetiren 2006) are quite unique heuristics dedicated only for solving flowshop with criteria $\sum w_i U_i$ as usually this variant is considered as a special case of a more general problem definition. In literature much more results are devoted to problems with criteria $\sum w_i T_i$ and its special case $\sum T_i$, especially worth recommending are results Ta (Ta 2017), Hamdi and Loukil (Hamdi and Loukil 2015), Ta et. al. (Ta, Billaut, and Bouquard 2018) and Sun and Gu (Sun and Gu 2017). In Dhingra et. al. (Dhingra, Kumar, and Dhingra 2017) specialized neural networks have been applied, however, results were not significantly changing the scene. A good overview on methods, algorithms and results on scheduling with minimizing the cost of tardy jobs is in Adamu and Adewumi (Adamu and Adewumi 2016). Parallel consideration for cyclic flowshop problem is presented in Bożejko i in. (Bożejko, Uchroński, and Wodecki 2016).

Deterministic cyclic flowshop problem

Let $\mathcal{J} = \{1, \ldots, n\}$ be a set of jobs to be executed cyclically (repeatedly) on m machines from the set $\mathcal{M} = \{1, \ldots, m\}$. At any given moment a specific machine can execute at most one job and all jobs needs to be executed without the preemption. Each job $j \in \mathcal{J}$ needs to be executed in sequence $1, 2, \ldots, m$ on every machine to follow technological line and if a job is being executed on the machine k then it means that it has been executed on machine the $k-1$ ($k = 2, 3, \ldots, m$). Jobs are executed in a given order determined by a permutation that is applied on all machines.

A set of jobs within a single cycle is named MPS (minimal part set) and MPSs are executed periodically one after another exactly the same way (i.e. jobs permutation) in each cycle. Having that, we can easily determine the beginning of jobs of any cycle: having the insight into the processing times of the first cycle, we can just multiply the cycle time by the number of considered MPS. In the paper we consider cyclic flowshop problem with machine setups and due dates defined as $(p_{i,j}, s_{k,i,j}, w_i, d_i)$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$, $k = 1, \ldots n$) where $p_{i,j}$ are processing times of operations, $s_{k,i,j}$ are machine setup times, w_i are weights for all jobs and $\tilde{d}_i$ are due dates for all jobs.

In order to simplify the further considerations we assume w.l.o.g. that later in the paper the considered solution is the natural permutation, i.e. $\pi = (1, 2, \ldots, n)$. Moreover, we define Π as a set of all permutations of the set $\mathcal{J}$ and if X is a random variable, then F_X denotes its cumulative distribution function.

Problem in the uncertain environment

In the paper we consider the described problem in the environment with uncertain parameters. We investigate the following variants of uncertainty:

- random processing times ($p_{i,j}$),
- random due dates (d_i).

For each of those variants we apply the following ways of calculating cost function:

- number of weights for tardy jobs,
- the total weighted tardiness.

where the actual cost is calculated on the last machine.

Random processing times

In this section we consider uncertain processing times which are represented by random variables with the normal distribution $\tilde{p}_{i,j} \sim N(p_{i,j}, c \cdot p_{i,j}))$, $i \in \mathcal{J}$, $j \in \mathcal{M}$. Other parameters, i.e. setup times $s_{k,i,j}$, due dates d_i and cost weights w_i are deterministic ($i \in \mathcal{J}$, $j, k \in \mathcal{M}$). Next we define

$$\widetilde{C}_{i,j} = \begin{cases} \sum_{k=1}^{i} p_{i,j}, & \text{for } j = 1, \\ \widetilde{C}_{i,j-1} + p_{i,j}, & \text{for } i = 1, j > 1, \\ \max\{\widetilde{C}_{i,j-1}, \widetilde{C}_{i-1,j}\} + \tilde{p}_{i,j}, & \text{for } i > 1, j > 1, \end{cases}$$

as the completion time of execution job i on machine j.

Now let's investigate the two considered ways of calculating the cost of jobs execution.

Number of weights for tardy jobs Having random variable $\widetilde{C}$ and other deterministic parameters, we define the cost as

$$\sum_{i=1}^{n} w_i \tilde{U}_i$$

where

$$\tilde{U}_i = \begin{cases} 0 & \text{for} \quad \widetilde{C}_{i,m} \leqslant d_i, \\ 1 & \text{for} \quad \widetilde{C}_{i,m} > d_i. \end{cases}$$

We consider the optimization problem where the goal is to find a permutation $\pi^* \in \Pi$ which minimizes cost of execution of all operations:

$$\widetilde{\mathcal{W}}(\pi^*) = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n} w_{\pi(i)} \widetilde{U}_{\pi(i)} \right).$$

In order to compare the costs of permutations from the set Π we introduce the following *comparison function* to calculate the value:

$$\mathcal{W} = w_i E(\tilde{U}_i) \tag{1}$$

where $E(\tilde{U}_i)$ is the expected value of the random variable $\tilde{U}_i$.

It is easy to show that

$$E(\tilde{U}_i) = P(\widetilde{C}_{i,m} > d_i) = 1 - F_{\tilde{C}_{i,m}}(d_i)$$

what finally gives us the following criteria to be applied the the modified tabu search implementation:

$$\mathcal{W} = \sum_{i=1}^{N} w_i (1 - F_{\tilde{C}_{i,m}}(d_i)) \tag{2}$$

The total weighted tardiness In this variant we define the cost as

$$\sum_{i=1}^{n} w_i \tilde{T}_i$$

where

$$\tilde{T}_i = \begin{cases} 0 & \text{for} \quad \widetilde{C}_{i,m} \leqslant d_i, \\ \widetilde{C}_{i,m} - \tilde{d}_i & \text{for} \quad \widetilde{C}_{i,m} > d_i. \end{cases}$$

We consider the optimization problem where the goal is to find a permutation $\pi^* \in \Pi$ which minimizes cost of execution of all operations:

$$\widetilde{W}(\pi^*) = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n} w_{\pi(i)} \widetilde{T}_{\pi(i)} \right).$$

In order to compare the costs of permutations from the set Π we introduce the following *comparison function* to calculate the value:

$$\mathcal{W} = w_i E(\tilde{T}_i) \tag{3}$$

where $E(\tilde{T}_i)$ is the expected value of the random variable $\tilde{T}_i$. Let's know recall the theorem from (Bożejko, Rajba, and Wodecki 2017):

Theorem 1 *Let assume that processing times are represented by independent random variables with the normal distribution* $\tilde{p}_{i,j} \sim N(p_{i,j}, c \cdot p_{i,j})$ $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, m)$. *Then for* $i \in \mathcal{J}$ *we have:*

$$E(\tilde{T}_i) = (1 - F_{\tilde{C}_{i,m}}(d_i)) \left(\frac{\sigma_i}{\sqrt{2\pi}} e^{\frac{-(d_i-\mu_i)^2}{2\sigma_i^2}} + (\mu_i - d_i) \left(1 - F_{N(0,1)}\left(\frac{d_i - \mu_i}{\sigma_i}\right) \right) \right).$$

where μ_i *and* σ_i *represent the mean and the standard deviation of* $\tilde{C}_{i,m}$.

The proof, supporting Lemmas and all other details can be found in (Bożejko, Rajba, and Wodecki 2017).

The formula obtained in Theorem 1 can be inserted into (3) and the comparison criteria can be applied to the modified tabu search implementation.

Random due dates

In this section we consider uncertain due dates which are represented by random variables with the normal distribution $\tilde{d}_i \sim N(d_i, c \cdot d_i))$, $i \in \mathcal{J}$. Other parameters, i.e. processing times $p_{i,j}$, setup times $s_{m,i,j}$, and cost weights w_i are deterministic. Next we define

$$C_{i,j} = \begin{cases} \sum_{k=1}^{i} p_{i,j}, & \text{for } j = 1, \\ C_{i,j-1} + p_{i,j}, & \text{for } i = 1, j > 1, \\ \max\{C_{i,j-1}, C_{i-1,j}\} + p_{i,j}, & \text{for } i > 1, j > 1, \end{cases}$$

as the completion time of execution job i on machine j.

Now let's investigate the two considered ways of calculating the cost of jobs execution.

Number of weights for tardy jobs

In this variant we define the cost similarly for the the variant with random processing times:

$$\sum_{i=1}^{n} w_i \tilde{U}_i$$

where

$$\tilde{U}_i = \begin{cases} 0 & \text{for} \quad C_{i,m} \leqslant \tilde{d}_i, \\ 1 & \text{for} \quad C_{i,m} > \tilde{d}_i. \end{cases}$$

We consider the optimization problem where the goal is to find a permutation $\pi^* \in \Pi$ which minimizes cost of execution of all operations:

$$\widetilde{W}(\pi^*) = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n} w_{\pi(i)} \widetilde{U}_{\pi(i)} \right).$$

In order to compare the costs of permutations from the set Π we introduce the following *comparison function* to calculate the value:

$$\mathcal{W}(\pi) = w_i E(\tilde{U}_i) \tag{4}$$

where $E(\tilde{U}_i)$ is the expected value of the random variable $\tilde{U}_i$.

It is easy to show that

$$E(\tilde{U}_i) = P(C_{i,m} > \tilde{d}_i) = F_{\tilde{d}_i}(C_{i,m})$$

what finally gives us the following criteria to be applied the the modified tabu search implementation:

$$\mathcal{W} = \sum_{i=1}^{N} w_i F_{\tilde{d}_i}(C_{i,m}) \tag{5}$$

The total weighted tardiness

In this variant we define the cost as

$$\sum_{i=1}^{n} w_i \tilde{T}_i$$

where

$$\tilde{T}_i = \begin{cases} 0 & \text{for} \quad C_{i,m} \leqslant \tilde{d}_i, \\ C_{i,m} - \tilde{d}_i & \text{for} \quad C_{i,m} > \tilde{d}_i. \end{cases}$$

We consider the optimization problem where the goal is to find a permutation $\pi^* \in \Pi$ which minimizes cost of execution of all operations:

$$\widetilde{W}(\pi^*) = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n} w_{\pi(i)} \widetilde{T}_{\pi(i)} \right).$$

In order to compare the costs of permutations from the set Π we introduce the following *comparison function* to calculate the value:

$$\mathcal{W} = w_i E(\tilde{T}_i) \tag{6}$$

where $E(\tilde{T}_i)$ is the expected value of the random variable $\tilde{T}_i$. Let's know recall the theorem from (Bożejko, Rajba, and Wodecki 2017):

Theorem 2 *Let assume that due dates are represented by independent random variables with the normal distribution* $\tilde{d}_i \sim N(d_i, c \cdot d_i)$ $(i = 1, 2, \ldots, n)$. *Then for* $i \in \mathcal{J}$ *we have:*

$$E(\tilde{T}_i) = F_{N(0,1)}\left(\frac{C_i - \mu_i}{\sigma_i}\right) \left(C_i F_{N(0,1)}\left(\frac{C_i - \mu_i}{\sigma_i}\right) + \frac{\sigma_i}{\sqrt{2\pi}} e^{-\frac{(C_i-\mu_i)^2}{2\sigma_i^2}} - \mu_i F_{N(0,1)}\left(\frac{C_i - \mu_i}{\sigma_i}\right) \right).$$

where μ_i *and* σ_i *represent the mean and the standard deviation of* $\tilde{d}_i$.

The proof, supporting Lemmas and all other details can be found in (Bożejko, Rajba, and Wodecki 2017).

The formula obtained in Theorem 1 can be inserted into (6) and the comparison criteria can be applied to the modified tabu search implementation.

Computational experiments

In this section we present the results of the tabu search execution for the considered problem variants, i.e.

- problem variant $\sum w_i U_i$ for solutions obtained in models: deterministic and random,

- problem variant $\sum w_i T_i$ for solutions obtained in models: deterministic and random.

All tests are performed with the usage of a tailored version of the tabu search method described in (Bożejko, Grabowski, and Wodecki 2006).

The base data come from (Ruiz and Stützle 2008) where we can four instance data sets. Each set is based on the original 120 instances described in (Taillard 1993) where one can find 12 groups with 10 instances each. A groups is defined by the number of jobs n and the number of machines m and all combinations $n \times m$ are: $20, 50, 100 \times 5, 10, 20$, $200 \times 10, 20$ and 500×20. Each of the four data sets is based on the different ranges from which setup times are uniformly generated: $[1, 9]$, $[1, 49]$, $[1, 99]$ and $[1, 124]$ (the maximum value in ranges is a percentage of the maximum processing time: 10%, 50%, 100% or 125% from 99, respectively).

For our experiments we selected the data set with with the smallest range for setup times as in most of the practical cases this one is the most realistic one and we focus on uncertainty of processing times. Then for all examples we have generated 100 instances of disturbed data (in total 12000 instances) and the description of the method for disturbed data generation can be found in (Bożejko, Rajba, and Wodecki 2018).

All the results are calculated as relative coefficient according to the following formula: $\delta = \frac{W - W^*}{W^*} 100\%$ and it expresses in how many percents the investigated solution W is worse than the reference (best known) solution W^*. Details of calculating robustness of the investigated methods can be also found in (Bożejko, Rajba, and Wodecki 2018).

The algorithm based on deterministic model we denote by AD, the one based on random model by AP.

Results

We have two sets of results: for random processing times and for random due dates. All diagrams present the robustness coefficient (the smaller, the better) for algorithms deterministic (noted as AD) and based on probabilistic model (noted as AP) expressed as percentage. We calculated results for different levels of data disturbance from 0.1 to 0.5 and we can also observe how the level of disturbance influences the solutions' robustness. We also discarded from the dataset the outstanding values which are significantly bigger than the majority, so this very small subset doesn't affect the actual result.

Let's now review the results for random processing times with the criteria $\sum w_i U_i$ which are presented on Fig. 1.

First, we can observe that applying probabilistic model significantly improves the robustness coefficient. We can also see that with increasing disturbance factor robustness is getting worse for the AD algorithm, however, what is a little surprising, but positive, it is on the same level for the AP one.

Then, looking at the result for random processing times with the criteria $\sum w_i T_i$ (Fig. 2) we can observe a slightly different situation. Again, AP algorithm performs better than AD one, but first, the differences are much smaller, and second, now with increasing disturbance factor for both algorithms robustness coefficient is increasing.

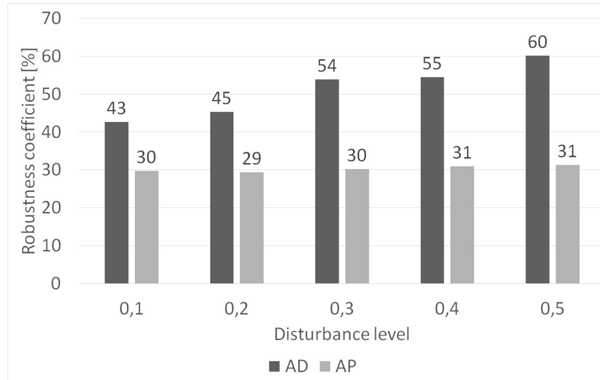

Figure 1: Comparison of the robustness level for random processing times with criteria $\sum w_i U_i$

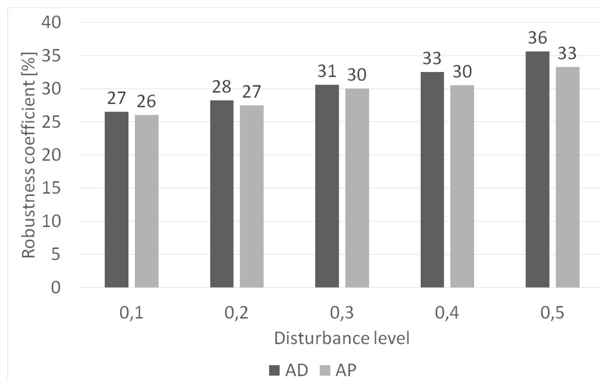

Figure 2: Comparison of the robustness level for random processing times with criteria $\sum w_i T_i$

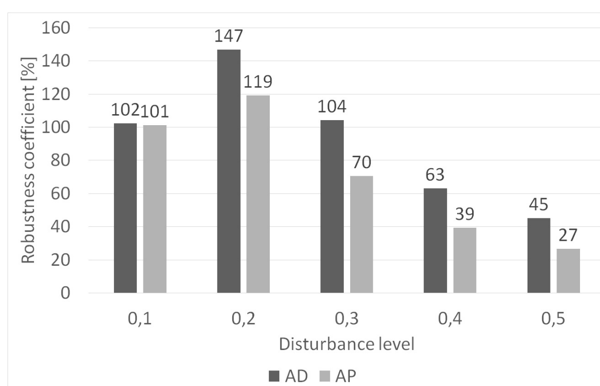

Figure 3: Comparison of the robustness level for random due dates times with criteria $\sum w_i U_i$

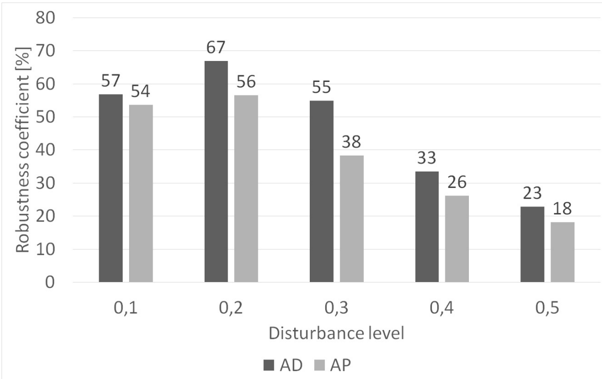

Figure 4: Comparison of the robustness level for random due dates times with criteria $\sum w_i T_i$

Analyzing results for random due dates both with the criteria $\sum w_i U_i$ (Fig. 3) and $\sum w_i T_i$ (Fig. 4) we can observe something surprising. Although between results for disturbance factor 0.1 and 0.2 the trend is as expected, the values for further disturbance levels are decreasing what is counter-intuitive. However, still AP is performing better than AD.

Conclusions

In the paper we investigated a cyclic flowshop problem with due dates and machine setup times in the uncertain environment modeled by random variables with the normal distribution. We proposed tabu search method modifications in a way which improves the robustness of calculated solutions. Computational experiments conducted on disturbed data confirmed that the proposed method offers more robust solutions than the ones obtained by the traditional deterministic approach.

References

Adamu, M. O.; and Adewumi, A. O. 2016. Minimizing the weighted number of tardy jobs on multiple machines: A review. *Journal of Industrial & Management Optimization* 12(4): 1465.

Bertel, S.; and Billaut, J.-C. 2004. A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research* 159(3): 651–662.

Bożejko, W.; Hejducki, Z.; Rajba, P.; and Wodecki, M. 2011. Project management in building process with uncertain tasks Times. *Management and Production Engineering Review* 2(1): 3–9.

Bożejko, W.; Rajba, P.; and Wodecki, M. 2014. Scheduling problem with uncertain parameters in Just in Time system. In *International Conference on Artificial Intelligence and Soft Computing*, 456–467. Springer.

Bożejko, W.; Uchroński, M.; and Wodecki, M. 2016. Parallel metaheuristics for the cyclic flow shop scheduling problem. *Computers & Industrial Engineering* 95: 156–163.

Bożejko, W.; Grabowski, J.; and Wodecki, M. 2006. Block approach-tabu search algorithm for single machine total weighted tardiness problem. *Computers & Industrial Engineering* 50(1/2): 1–14.

Bożejko, W.; Rajba, P.; and Wodecki, M. 2017. Stable scheduling of single machine with probabilistic parameters. *Bulletin of the Polish Academy of Sciences: Technical Sciences* 65(2): 219–231.

Bożejko, W.; Rajba, P.; and Wodecki, M. 2018. Robustness of the Uncertain Single Machine Total Weighted Tardiness Problem with Elimination Criteria Applied. In *International Conference on Dependability and Complex Systems*, 94–103. Cham.: Springer.

Dhingra, A. K.; Kumar, N.; and Dhingra, S. 2017. Minimization of Makespan and Total Tardiness in a Flow Shop Scheduling Using Artificial Neural Network. *IUP Journal of Operations Management* 16(1): 7.

Hamdi, I.; and Loukil, T. 2015. Minimizing total tardiness in the permutation flowshop scheduling problem with minimal and maximal time lags. *Operational Research* 15(1): 95–114.

Hariri, A.; and Potts, C. 1989. A branch and bound algorithm to minimize the number of late jobs in a permutation flow-shop. *European Journal of Operational Research* 38(2): 228–237.

Lenstra, J. K.; Kan, A. R.; and Brucker, P. 1977. Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, 343–362. Elsevier.

Rajba, P.; and Wodecki, M. 2012. Stability of scheduling with random processing times on one machine. *Applicationes Mathematicae* 2(39): 169–183.

Ruiz, R.; and Stützle, T. 2008. An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research* 187(3): 1143–1159.

Sun, Z.; and Gu, X. 2017. Hybrid algorithm based on an estimation of distribution algorithm and cuckoo search for the no idle permutation flow shop scheduling problem with the total tardiness criterion minimization. *Sustainability* 9(6): 953.

Ta, Q. C. 2017. *Matheuristic algorithms to minimize total tardiness in flow shop scheduling*. Editions universitaires europeennes.

Ta, Q. C.; Billaut, J.-C.; and Bouquard, J.-L. 2018. Matheuristic algorithms for minimizing total tardiness in the m-machine flow-shop scheduling problem. *Journal of Intelligent Manufacturing* 29(3): 617–628.

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

Ucar, H.; and Tasgetiren, M. F. 2006. A particle swarm optimization algorithm for permutation flow shop sequencing problem with the number of tardy jobs criterion. In *Proceedings of 5 th International Symposium of Intelligent Manufacturing Systems*, 1110–1120.