



**QUEEN'S
UNIVERSITY
BELFAST**

New Scheduling Algorithms and Digital Tool for Dynamic Permutation Flowshop with Newly Arrived Order

Liu, W., Jin, Y., & Price, M. (2017). New Scheduling Algorithms and Digital Tool for Dynamic Permutation Flowshop with Newly Arrived Order. DOI: 10.1080/00207543.2017.1285077

Published in:

International Journal of Production Research

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2017 Informa UK Limited, trading as Taylor & Francis Group.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Production Research on 6/2/2017, available online: <http://www.tandfonline.com/10.1080/00207543.2017.1285077>.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

New Scheduling Algorithms and Digital Tool for Dynamic Permutation Flowshop with Newly Arrived Order

Weibo Liu, Yan Jin*, Mark Price

School of Mechanical and Aerospace Engineering, Queen's University Belfast, Ashby Building,
Belfast, BT9 5AH, UK

[*y.jin@qub.ac.uk](mailto:y.jin@qub.ac.uk) (Yan Jin, Phone: +44 (0)28 9097 4102)

Key words: Heuristics; Dynamic scheduling; New order; Permutation flowshop; Order mixing;

Abstract

The permutation flowshop scheduling problem has been widely studied under static environment by assuming machines and jobs are available at the time of zero. However, in reality, new orders arrive at production systems randomly, which leads to sheer complexity in scheduling due to the dynamic changes given various constraints of resources. Previous studies simply attach new orders directly after the existing schedule. Recent study by Perez-Gonzalez and Framinan (2015) shows mixing jobs of old and new orders could result in better scheduling solutions. But the heuristic algorithms are still lacking to implement the job mixing policy. To address this problem, a novel scheduling strategy is herein proposed by integrating match-up strategy and real-time strategy (MR) in order to make use of the remaining time before the old order due date. Based on the new MR strategy, eleven new heuristics are introduced with ten existing and one new priority rules. Computational results illustrate the effectiveness of the new heuristics. A digital tool is developed for ease of application of these heuristics, and it is validated by case studies.

Key words: Heuristics; Dynamic scheduling; New order; Permutation flowshop; Order mixing;

1. Introduction

Static scheduling has been studied for decades and many approaches have been developed especially for flowshops (see e.g., Liao and Huang 2010; Vasiljevic and Danilovic 2015; Liu, Jin, and Price 2016). However, a static model could not fully reflect production reality. The scheduling problem in the presence of real-time events, termed dynamic scheduling, has begun to draw attention from industry and academia (Perez-Gonzalez and Framinan 2009; Perez-Gonzalez and Framinan 2010; Rahman, Sarker, and Essam 2015). Flowshop as an effective production model is widely used in industry, and lots of real-time events should be responded immediately. The real-time events include new order arrival (see e.g., Perez-Gonzalez and Framinan 2010; Rahman, Sarker, and Essam 2015), machine breakdown (see e.g., Safari and Sadjadi 2011), rush order (see e.g., Nandi and Rogers 2004) and varying processing time (see e.g., Xia, Chen, and Yue 2008; Li et al. 2015) etc. Among these disruptions, new orders arrive frequently in manufacturing industry especially for mass customized production. In reality, new orders arrive randomly associated with disparate arrival times and due dates which increases the complexity of the scheduling problem significantly. Many studies focus on job rescheduling with single-machine (see e.g., Tamer Unal, Uzsoy, and Kiran 1997), two-agent single-machine (see e.g., Feng et al. 2015; Kovalyov, Oulamara, and Soukhal 2015) or multi-agent single-machine (see e.g., Cheng, Ng, and Yuan 2006). Very few studies have been conducted to investigate the scheduling methods for m -machine flowshops which is more practical and common in

industry (Rahman, Sarker, and Essam 2015; Perez-Gonzalez and Framinan 2010). In this paper, the dynamic scheduling problem with new order arrival in permutation flowshops is studied.

The objective is to minimize the maximum completion time of a new order while the due date of the old order has to be maintained. Note that each order may contain multiple jobs. It is a constraint optimization problem which is NP-hard in a strong sense especially when the slack between old order completion time and due date is very large (Perez-Gonzalez and Framinan 2010). Moreover, the problem is practical and meaningful for manufacturers as: 1) it is essential to set a reasonable and tight due date for the new order; 2) it is necessary to determine if the order is acceptable or not when the due date is assigned; 3) it helps to maximize the number of accepted orders so as to achieve the maximum revenue. This study aims to propose new heuristic algorithms for solving this problem.

The existing scheduling approaches to handling newly arrived orders or rush orders can be classified into three categories (Herroelen and Leus 2004; Aytug et al. 2005): 1) reactive approach; 2) proactive approach; 3) predictive-reactive approach. Dispatching rule is a representative of the reactive approach. Different dispatching rules such as shortest processing time (SPT), first come first serve (FCFS), earliest due date (EDD) etc. are evaluated with respect to mean flowtime, maximum flowtime, maximum tardiness etc. when coping with newly arrived orders (Rajendran and Oliver 1999). Four dispatching rules are developed to minimize job tardiness and rejection cost for dynamic flexible flowshops (Kianfar, Fatemi Ghomi, and Karimi 2009). Other work on dispatching rules in dynamic flowshop can be found in (Lodree, Jang, and Klein 2004; Branke and Mattfeld 2005; El-Bouri, Balakrishnan, and Popplewell 2008). Although it can respond to uncertainties immediately, the schedule quality cannot be guaranteed by dispatching rules, because the schedule is constructed based on local information, i.e. job characteristic, rather than from the view of the whole schedule. When multi-job new orders arrive, two scenarios are differentiated (Perez-Gonzalez and Framinan 2010): 1) old order is ‘frozen’ that rescheduling of the old order is not allowed (Perez-Gonzalez and Framinan 2009; Rahman, Sarker, and Essam 2015); 2) old order is ‘active’ that can be mixed and rescheduled with the new order (Perez-Gonzalez and Framinan 2010). When the old order is set as frozen, two strategies are developed including right shifting strategy (RS) and real-time strategy (RT) (Rahman, Sarker, and Essam 2015) to attach the new order after the existing schedule. Right shifting strategy means that new jobs are sequenced first under static environment, and then attached immediately after the old order. Jobs may be shifted to the right of the existing schedule given the machine availability. Real-time strategy is to schedule new jobs accounting the tail profile of the schedule of the old order. Three scheduling policies are proposed for solving three types of problems including classical scheduling problem (CSP), availability scheduling problem (ASP) and multi-agent scheduling problem (MSP) (Perez-Gonzalez and Framinan 2015). The policy for solving CSP is similar to the RS strategy, but the difference is new jobs are immediately processed once the old jobs are finished on the first machine by the RS strategy. For CSP, new jobs start to be processed after all old jobs are completed. The policy for ASP is the same as the RT strategy. For MSP, order mixing and job rescheduling is allowed. Their results show that the policy for MSP provides good makespan for new orders when the slack time (due date - completion time) of the old order is medium or high comparing to CSP and ASP. To the authors’ knowledge, very few heuristic algorithms have been developed for dynamic scheduling problems with mixed orders, except for the meta-heuristic developed by Perez-Gonzalez and Framinan (2010).

The proactive approach is also widely used to deal with uncertainties. By inserting idle times, a proactive schedule is generated to cope with machine breakdowns (see e.g., O’Donovan, Uzsoy, and McKay 1999; Zhu and Zhou 2014). A proactive algorithm considering processing time uncertainties

is developed for hybrid flowshop with the objective of makespan minimization of initial schedule and makespan deviation of initial and realized schedules (Chaari et al. 2011). Three models are developed with processing time variation for single-machine scheduling problems (Wu, Brown, and Beck 2009). Those proactive approaches based on anticipated disturbance are robust to uncertainties. However, only a limited number of disruptions can be absorbed. It is difficult and impractical to deal with new orders with proactive approaches. The information of orders such as order type, job processing times, arriving time, and due date etc. can hardly be predicted and simulated simultaneously.

The predictive-reactive approach is commonly used as a scheduling method under uncertainty. Two steps are included: first, a predictive schedule is generated over the time horizon; then the existing schedule is modified to respond to uncertainties. This method can provide robust schedules and respond to uncertain events quickly. Match-up strategy (Bean and Birge 1986) is deemed as effective to handle rush order (Moratori, Petrovic, and Vázquez 2008), newly arrived order (Moratori, Petrovic, and Vázquez-Rodríguez 2012) or machine breakdown (Akturk and Gorgulu 1999) with an objective of schedule stability. It aims to catch up the schedule at a future point by adding jobs into the existing schedule when handling new jobs. Rush order is accommodated into the existing schedule by a match-up algorithm in a job shop (Moratori, Petrovic, and Vázquez 2008). A match-up algorithm is developed to schedule newly arrived jobs (Moratori, Petrovic, and Vázquez-Rodríguez 2012). However, only one job can be scheduled at one time and the algorithm cannot be applied to the orders consisting of multiple jobs associated with a common due date.

In this paper, a novel strategy is proposed by integrating match-up strategy and real-time strategy with order mixing. The match-up strategy is used to insert selected jobs from the new order, and the real-time strategy is applied for scheduling the rest jobs of the new order. Based on the new strategy, eleven new heuristics are developed and tested by using ten existing and one new priority rules. A digital tool is developed for implementing these algorithms automatically.

The paper is organized as follows: In Section 2, the problem is defined and assumptions are given. Section 3 describes the new strategy, the new priority rule and the new heuristics. Experimental evaluations of these new heuristics are conducted in Section 4. In Section 5, the newly developed scheduling tool is presented in detail. Finally, conclusions are made in Section 6.

2. Problem statement

The problem studied in this paper is a constraint optimization problem which has been analysed in (Perez-Gonzalez and Framinan 2015). It is assumed that n_o jobs belonging to the old order J_o have been scheduled but not fully completed yet when n_N jobs from the new order J_N arrive at the production system. The jobs from the old order have a common due date d_o which cannot be violated. Mathematically it can be expressed as $T_{max}^{J_o} = 0$ which means the maximum tardiness of jobs from the old order is 0. From the view of dynamic scheduling, it is assumed that many new orders will arrive continuously and randomly, but only one order arrives at one time. So order selection is not considered in this paper. In order to find the earliest due date for J_N , the objective is to minimize the maximum completion time of the new order $C_{max}^{J_N}$ given the constraint of machine availability, a_k , following the convention of (Perez-Gonzalez and Framinan 2009). The problem can be denoted as $Fm|prmu, d_o, a_k|C_{max}^{J_N}/T_{max}^{J_o} = 0$ according to (Graham et al. 1979) where Fm represents a flowshop with m machines, $prmu$ stands for permutation, d_o indicates that all jobs from the old order have a common due date, a_k represents the availability on machine k , and $C_{max}^{J_N}/T_{max}^{J_o} = 0$ is a

constraint objective. Table 1 lists all the notations used in this paper. To define the problem, the assumptions are described as follows.

- 1) All jobs should start as soon as possible;
- 2) Processing time is known and deterministic;
- 3) Setup time is included in processing time;
- 4) Machines are continuously available but cannot process two or more jobs simultaneously;
- 5) Job pre-emption is not permitted;
- 6) Buffers' capacity between machines is infinite;
- 7) Only permutation schedules are allowed;
- 8) Each order arrives randomly;
- 9) Each order may contain one or multiple jobs;
- 10) All jobs in an order should be finished before their due date;
- 11) The job information is known after order arrives;
- 12) Once the new order arrives, job rescheduling is activated;
- 13) If all existing jobs are finished and no new jobs arrive, all machines stay idle and available;
- 14) No more than two orders can be mixed together i.e. only the last old order can be rescheduled with the newly arrived order.

Table 1 Notations

Notation	Description
J_O	Job set of the old order
J_N	Job set of the new order
D_O	Due date of the old order
m	Number of machines
n_O	Number of jobs belonging to the old order
n_N	Number of jobs belonging to the new order
k	Machine index, $k \in [1, m]$
j	Job index for the old order, $j \in [1, n_O]$
i	Job index for the new order, $i \in [1, n_N]$
$t_{i,k}$	Processing time of job i on machine k
F_i	Indicator of weight distribution of processing times on front machines
B_i	Indicator of weight distribution of processing times on back machines
$T_{max}^{J_O}$	The maximum tardiness of the old order
$C_{max}^{J_N}$	The maximum completion time of the new order
AVG_i	The average of processing times of job i
STD_i	The standard deviation of processing times of job i
SKE_i	The skewness of processing times of job i
KUR_i	The kurtosis of processing times of job i

For an 'active' old order, the slack time between its due date and completion time has a direct impact on the schedule performance. If the slack time is very large, more jobs from the new order could be absorbed and scheduled before the due date of the old order. Herein, a relaxed and reasonable due date for the old order is assumed.

To cope with this problem, a new strategy is proposed taking account of computational efficiency and solution quality. The details of the new heuristics are described in subsequent sections.

3. New heuristics based on match-up & real-time strategy

3.1 New strategy: MR strategy

The strength of RS and RT strategies exist in their efficiency at the cost of schedule quality. Ideally, the best schedule could be achieved by mixing new jobs together with uncompleted old jobs given the due date constraint. However, the computation time would be very long if a heuristic focuses on finding jobs from the new order given the large number of new jobs. To improve the schedule quality while maintaining the efficiency, a novel strategy is introduced by integrating match-up strategy (Bean and Birge 1986) and real-time strategy (Rahman, Sarker, and Essam 2015).

In the new strategy, a limited search of mixed sequences is conducted by inserting some jobs from the new order into the old order. Jobs are normally finished before their due date and therefore, there is some space before the old order due date for some jobs from the new order. To fully utilize this slack, some jobs from the new order should be carefully selected and rescheduled with the remaining jobs of the old order. This idea is taken from the match-up strategy (Bean and Birge 1986) when handling rush orders that collect idletime on each machine and accommodate the new ones into the existing schedule. Figure 1 shows the structure of the newly generated schedule, including the completed jobs of the old order, the mixed jobs including the remaining jobs of the old order and the selected jobs from the new order, and the rest jobs of the new order. Herein, the slack between the old order due date and the completion time is taken as the accumulative idletime. The selected jobs from the new order are mixed and scheduled with the remaining jobs of the old order. The rest jobs of the new order are then scheduled by using the real-time strategy considering the profile of the reconstructed schedule. Therefore, the new strategy is called match-up & real-time strategy (MR).

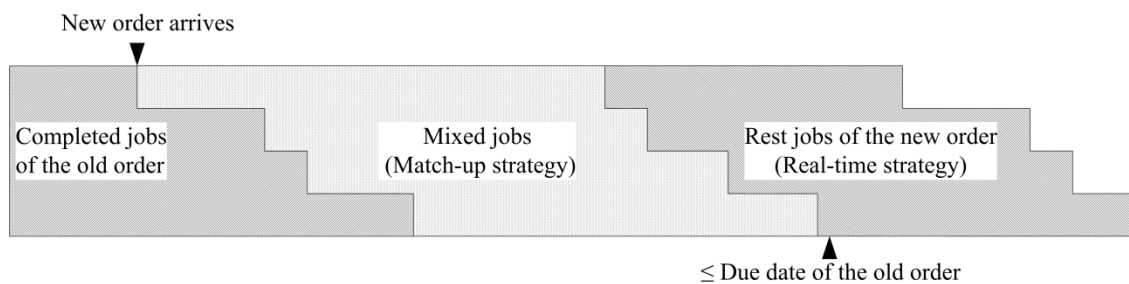


Figure 1 The structure of a generated schedule

The flow chart of the MR strategy is shown in Fig. 2. First, all jobs from the new order are sorted by a priority rule, to determine the priority level for each job to be selected. Then, the number of jobs to be selected from the new order is determined based on the accumulative idletime on the last machine. Next, reschedule the mixed jobs including the remaining jobs of the old order together with the selected jobs from the new order. If the maximum completion time of these mixed jobs violates the due date constraint of the old order, the last selected job will be returned to the rest jobs of the new order, and the updated mixed jobs will be rescheduled again until the due date is satisfied. Note that the maximum completion time of the mixed jobs should be lower or equal to the due date of the old order. And it is used directly to determine if the old order due date is violated or not for heuristic efficiency as it will take extra time to compute the exact completion time of the old order. Finally, schedule the rest jobs of the new order by real-time strategy given the profile of the mixed jobs schedule.

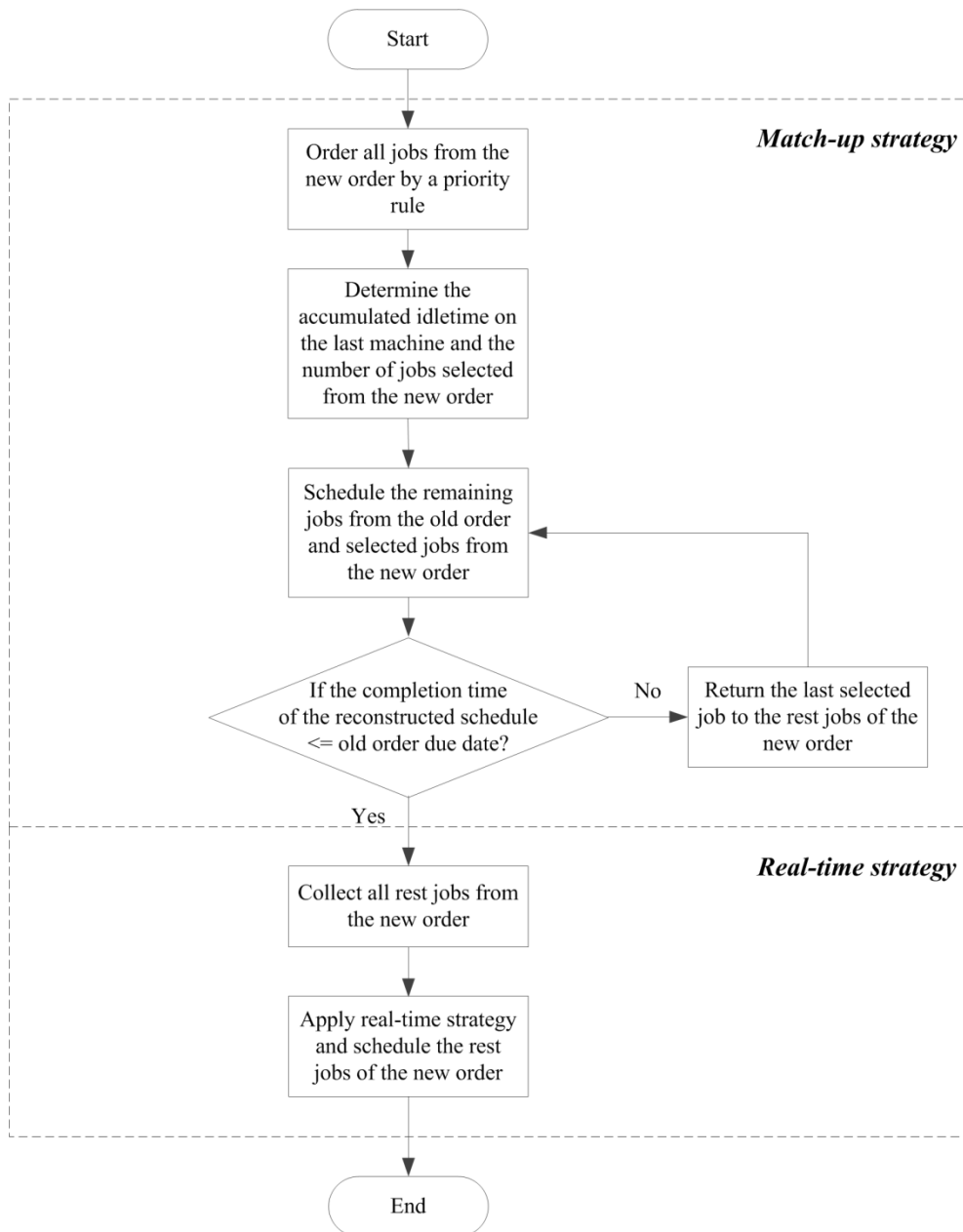


Figure 2 Flow diagram of MR strategy

There are three questions to answer in implementing MR strategy: 1) what kind of jobs should be selected from the new order to be mixed with the remaining old jobs; 2) how many jobs should be selected; 3) the job scheduling algorithm. The details of MR strategy are described below and the three questions are answered.

3.2 Priority rule

What kind of jobs should be selected? The job selection from the new order determines the schedule performance directly. The job priority rule in the MR strategy should be investigated first. Although many priority rules such as Johnson's rule (Johnson 1954), dispatching rules, and priority rules (Dong, Huang, and Chen 2008; Kalczynski and Kamburowski 2008; Kalczynski and Kamburowski 2009) have been developed, there is no dedicated job sorting rule for achieving effective order mixing. In this section, a new rule based on Johnson's rule is introduced exclusively for the MR strategy, and popular existing job priority rules are also used.

3.2.1 A new rule: *Weighted Weight Distribution rule*

According to Johnson's rule (Johnson 1954), the job with small processing times on front machines and large processing times on back machines should be prioritized with an objective of minimizing maximum completion time. In other words, the job with large processing times on front machines and small processing times on back machines should be scheduled last. Based on this idea, a new job sorting rule is proposed in consideration of weight distribution of job processing times on each machine. Jobs are sorted in the descending order of $(B_i - F_i)$ where

$$B_i = \sum_{k=1}^m k * t_{i,k}, \quad (1)$$

$$F_i = \sum_{k=1}^m (m + 1 - k) * t_{i,k}. \quad (2)$$

B_i in Eq. (1) indicates how 'heavy' the job processing times are on back machines by allocating weights to processing times on each machine from 1 to m according to the processing order whereas F_i in Eq. (2) measures how 'heavy' of the job processing times are in the front. Because different weights are assigned to each machine, so the new rule is named as the weighted weight distribution rule (WWD).

3.2.2 *Conventional dispatching rules*

Many dispatching rules have shown effectiveness in scheduling with various criteria, such as total flowtime, makespan and tardiness. In this paper, it is assumed that all jobs from the new order arrive simultaneously with a common due date assigned. So the dispatching rules such as earliest due date, first come first serve, and last come first serve etc. cannot be applied. Only SPT and longest processing time (LPT) based on job processing time data are employed and implemented within the new MR strategy.

3.2.3 *Existing priority rules for NEH heuristic*

The NEH heuristic (Nawaz, Ensore Jr., and Ham 1983) is commonly regarded as the best constructive heuristic for static permutation flowshop scheduling problems with makespan criterion. The priority rule for the NEH heuristic can be applied to the dynamic problem where jobs are selected and scheduled according to their priorities. Until now, many priority rules have been developed and proven effective, such as rules in NEH-D (denoted by STD) (Dong, Huang, and Chen 2008), NEHKK1 (KK1) (Kalczynski and Kamburowski 2008), NEHKK2 (KK2) (Kalczynski and Kamburowski 2009), and rules (SKE, KUR, and LJP) in (Liu, Jin, and Price 2015) etc. The priority rule in the NEH heuristic (AVG) allocates high priorities to jobs with large sum of processing times on all machines. While the STD rule takes standard deviation of processing times into consideration. In SKE, KUR and LJP rules, higher moments of distributions of processing times including skewness and kurtosis are used for job differentiation. KK1 and KK2 are based on Johnson's rule while different weights are allocated to processing times.

To compare the effectiveness of all above priority rules, a random sequence (RND) is also taken as a reference. In total, eleven priority rules implemented with the new MR strategy are introduced, as shown in Table 2. Their performance will be compared in Section 4.

Table 2 Priority rules for the new MR strategy

Heuristics	Reference	Priority rule
MR_WWD	Herein paper	Descending order of $(B_i - F_i)$ where $B_i = \sum_{k=1}^m k * t_{i,k}$ and $F_i = \sum_{k=1}^m (m + 1 - k) * t_{i,k}$
MR_AVG	(Nawaz, Ensore Jr., and Ham 1983)	Descending order of AVG_i where $AVG_i = \frac{1}{m} \sum_{k=1}^m t_{i,k}$
MR_STD	(Dong, Huang, and Chen 2008)	Descending order of $AVG_i + STD_i$, where $STD_i = \sqrt{\frac{1}{m-1} \sum_{k=1}^m (t_{i,k} - AVG_i)^2}$
MR_SKE	(Liu, Jin, and Price 2015)	Descending order of $AVG_i + STD_i + abs(SKE_i)$, where $SKE_i = \frac{\frac{1}{m} \sum_{k=1}^m (t_{i,k} - AVG_i)^3}{\left(\sqrt{\frac{1}{m} \sum_{k=1}^m (t_{i,k} - AVG_i)^2}\right)^3}$
MR_KUR	(Liu, Jin, and Price 2015)	Descending order of $AVG_i + STD_i + 1/KUR_i$ where $KUR_i = \frac{\frac{1}{m} \sum_{k=1}^m (t_{i,k} - AVG_i)^4}{\left(\frac{1}{m} \sum_{k=1}^m (t_{i,k} - AVG_i)^2\right)^2}$
MR_LJP	(Liu, Jin, and Price 2015)	Descending order of $AVG_i + STD_i + abs(SKE_i) + 1/KUR_i$
MR_KK1	(Kalczynski and Kamburowski 2008)	Non-increasing sum of weighted processing times $\min(a1_i, b1_i)$ where $a1_i = \sum_{k=1}^m \left[\frac{(m-1)(m-2)}{2} + m - k \right] t_{i,k}$, $b1_i = \sum_{k=1}^m \left[\frac{(m-1)(m-2)}{2} + k - 1 \right] t_{i,k}$
MR_KK2	(Kalczynski and Kamburowski 2009)	Non-increasing sum of weighted processing times $\min(a2_i, b2_i)$ where $a2_i = \sum_{k=1}^m t_{i,k} + \sum_{h=1}^s \left(\frac{h-3}{s-3} - \varepsilon \right) (t_{s+1-h,i} - t_{t+h,i})$, $b2_i = \sum_{k=1}^m t_{i,k} - \sum_{h=1}^s \left(\frac{h-3}{s-3} - \varepsilon \right) (t_{s+1-h,i} - t_{t+h,i})$ and $s = \lfloor m/2 \rfloor$, $t = \lceil m/2 \rceil$
MR_SPT		Ascending order of operation times on the first machine
MR_LPT		Descending order of operation times on the first machine
MR_RND		Random job sequence

3.3 Number of selected jobs

How many jobs should be selected from the new order for mixing with the remaining jobs of the old order? Obviously, the completion time on the last machine directly determines the length of production time. Therefore, the slack between the due date and completion time of the old order on the last machine can be deemed as the collected idletime and used to accommodate the selected jobs. Generally speaking, the larger the number of selected jobs from the new order to be accommodated in the slack, the better is the resulted schedule. So in the new strategy, the processing times of the selected jobs on the last machine are accumulated until no more jobs from the new order can be accommodated.

3.4 Scheduling algorithm

What algorithm should be used for scheduling? The job sequence of mixed jobs and rest jobs of the new order should be optimized by some algorithm. The problem has been transformed into a static permutation flowshop scheduling problem by the new strategy. Either heuristics or meta-heuristics

can be adopted. In order to save computation time, a heuristic algorithm, NEH heuristic, is chosen and implemented within the new MR strategy. Herein, RS and RT strategies are taken as references and the job scheduling is performed by the NEH heuristic as well, denoted by RS_NEH and RT_NEH.

Therefore, eleven new heuristics are constructed by combining the new MR strategy and different priority rules. The effectiveness of these new heuristics is investigated in the subsequent sections.

4. Experimental evaluation

In order to evaluate the effectiveness of the new strategy, new WWD rule and new heuristics, an intensive analysis is conducted. The new heuristics are validated with references of RS_NEH and RT_NEH heuristics. The Taillard (Taillard 1993) and the VRF (Vallada, Ruiz, and Framinan 2015) benchmarks are used. Taillard test bed is the most widely used for the PFSP including 12 different size problems ranging from small size problem ($n=20$ and $m=5$) to large size problem ($n=500$ and $m=20$). Each problem includes 10 instances. VRF benchmark is the newest test bed with 480 instances, including 240 for small size problems and 240 for large size problems. Therefore, 600 instances in total are used for heuristic evaluation in the herein paper.

To simulate the dynamic flowshop, each instance is divided into two halves in terms of the number of jobs: the first half of the instance is taken as the old order, while the second half is defined as the new order. The old order due date is set as $(1 + \alpha)C_{max}^{Jo}$ where $\alpha \in [0.1, 1]$ is the relaxation factor and C_{max}^{Jo} the maximum completion time of the old order obtained by the NEH heuristic. To evaluate each heuristic, the performance measure of relative percentage deviation (RPD) is computed as follows:

$$RPD = \frac{C_{max}^{JN} - UB}{UB} \times 100\%, \quad (3)$$

where C_{max}^{JN} is the maximum completion time of the new order obtained by each heuristic on every instance, UB the upper bound for each instance. Herein, the UB values provided by Taillard (Taillard 2015) and Vallada et al. (Vallada, Ruiz, and Framinan 2015) are used directly because they can be applied under all scenarios of α . All algorithms are coded in Matlab R2013b and run on a CPU E5520 computer with 6.00G memory.

The relaxation factor α affects the performance of new heuristics directly. First, different values are assigned to α in order to check the impact of the due date on the new heuristic performance. Figures 3 and 4 shows the test results of the new heuristic based on the MR strategy and the WWD rule on Taillard and VRF benchmarks respectively.

As shown in Fig. 3, with α increasing, the average RPD (ARPD) value of MR_WWD decreases gradually, and it is much better than that of RS_NEH and RT_NEH. The performance of MR_WWD improves as the relaxation factor increase. With a large relaxation of α , more new jobs can be accommodated by the old order due date and the probability of obtaining a better schedule improves. The honest significant differences intervals (HSD) between MR_WWD and RS_NEH and RT_NEH heuristics do not intersect when $\alpha \geq 0.2$.

It is essential to verify the effectiveness of the new strategy on the VRF benchmark as it is a new hard benchmark. Figure 4 shows the similar trend that MR_WWD achieves lower APRDs with larger values of α . But the performance of MR_WWD stays steady after $\alpha=0.4$. This is because the slack time before the old order due date is large enough to accommodate the new order when $\alpha \geq 0.4$. To

investigate the performance of all new heuristics, a reasonable and relaxed due date is set that α is defined as 0.4, the same value as defined in (Perez-Gonzalez and Framinan 2010).

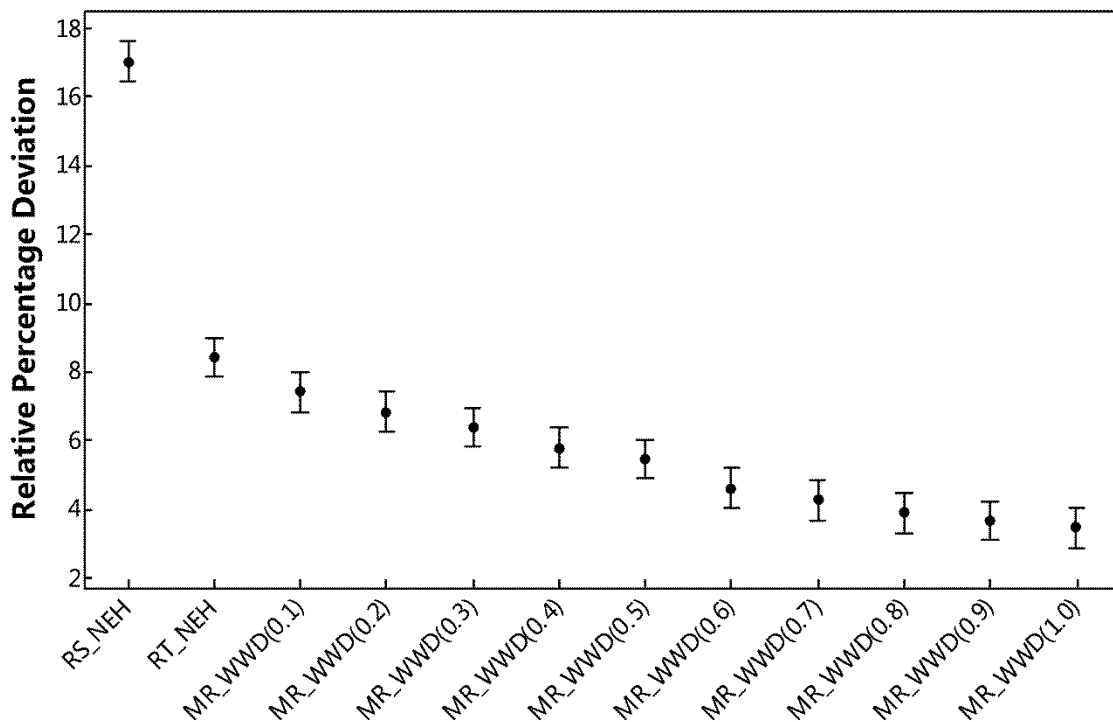


Figure 3 Means and 95% HSD intervals for RS_NEH, RT_NEH and MR_WWD on Taillard benchmark with different α

Table 3 shows the test results of each heuristic on different size problems on Taillard benchmark in terms of ARPD with $\alpha=0.4$. Eleven priority rules are implemented within the MR strategy respectively. The table shows that the new heuristic MR_WWD performs the best in most situations with an ARPD value of 5.74, much better than that of RT_NEH (8.40) and RS_NEH (17.05). On 10/12 problems, MR_WWD generates the best solution and MR_SPT is the best on the rest of problems. Note that all priority rules within the MR strategy have a better performance than RT_NEH except for MR_LPT and MR_RND, proving the effectiveness of the new strategy. Among all MR strategy based heuristics, MR_WWD outperforms the others, followed by MR_SPT, MR_STD, MR_KUR, MR_AVG, MR_LJP, MR_SKE, MR_KK2, MR_KK1, MR_RND and MR_LPT heuristics.

The results of each heuristic on VRF test bed are shown in Tables 4 and 5. The same conclusion can be drawn that MR_WWD has the best performance with an ARPD of 3.59, with 56.95% and 78.12% improvement comparing to RS_NEH and RT_NEH heuristics. On small instances, the ARPD of MR_WWD heuristic is 3.87, and it can achieve the best on 12 out of 24 problems. Other MR based heuristics dominate on the other problems. On large instances, MR_WWD has the lowest ARPD of 3.31, much better than the RS_NEH and RT_NEH heuristics. All MR strategy based heuristics except MR_RND are more effective than RS_NEH and RT_NEH.

To identify statistically significant differences among all examined heuristics, 95% HSD intervals of ARPDs for each heuristic are calculated, as shown in Fig.5 and Fig. 6 on different test beds. It is concluded that on Taillard benchmark, MR_WWD performs significantly better than RS_NEH and RT_NEH heuristics. Significant differences between MR_WWD and MR_AVG, MR_STD,

MR_SKE, MR_KUR, MR_LJP, MR_KK1, MR_KK2, MR_SPT, MR_LPT, and MR_RND can also be observed. On VRF benchmark, all MR strategy based heuristics shows significant differences comparing to RS_NEH and RT_NEH heuristics except for MR_RND.

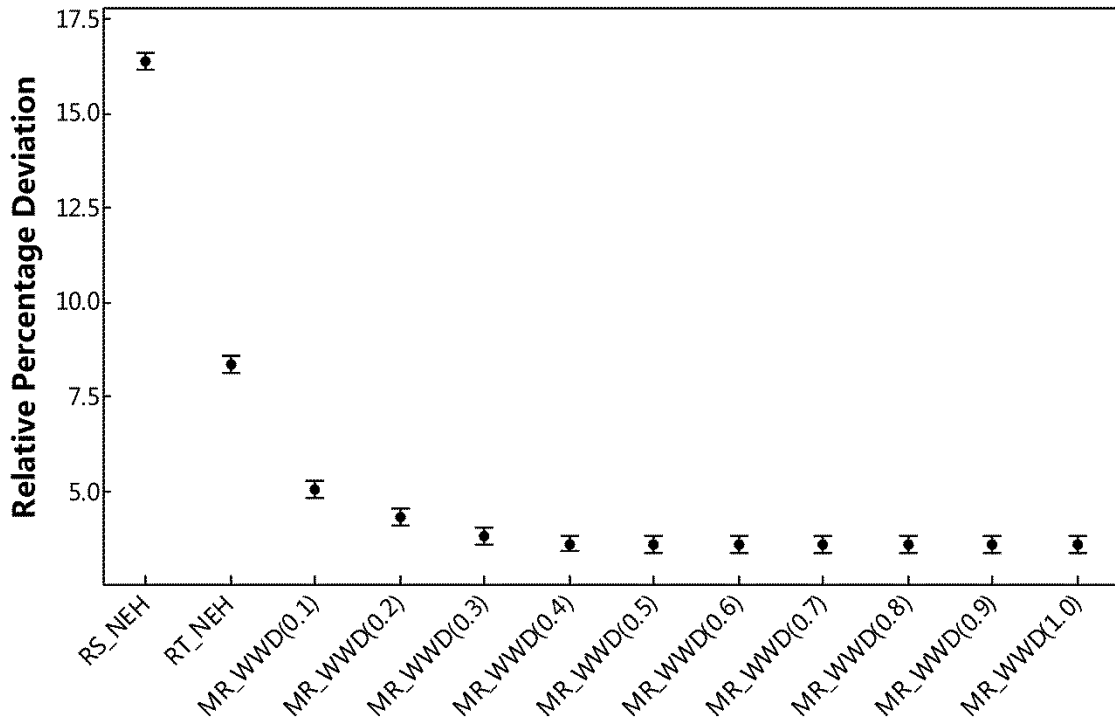


Figure 4 Means and 95% HSD intervals for RS_NEH, RT_NEH and MR_WWD on VRF benchmark with different α

To further confirm the statistically significant differences between heuristics, the paired-samples t test is conducted with a confidence level of 95%. Key results are shown in Tables 6 and 7. These results are consistent with above conclusions that significant differences exist between MR_WWD and other heuristics on Taillard. On VRF benchmark, the differences are also significant except for MR_AVG, MR_STD, MR_SKE, MR_KUR, and MR_LJP heuristics. This is because MR strategy is used in all these heuristics.

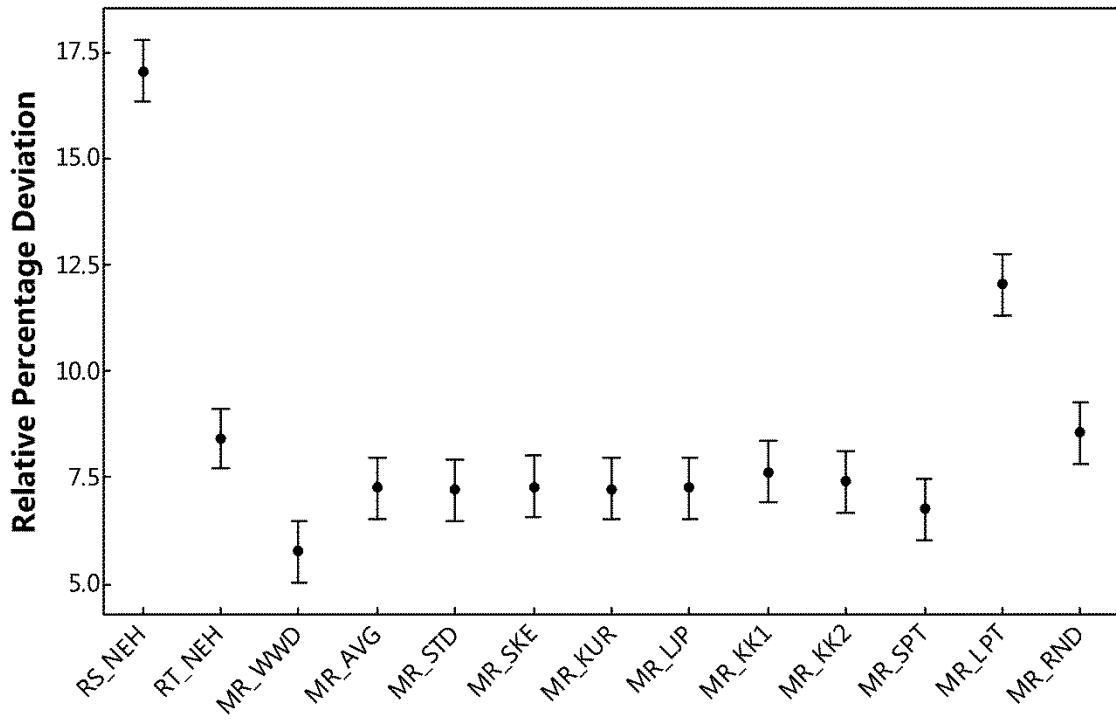


Figure 5 Means plot and 95% HSD intervals of each heuristic on Taillard benchmark

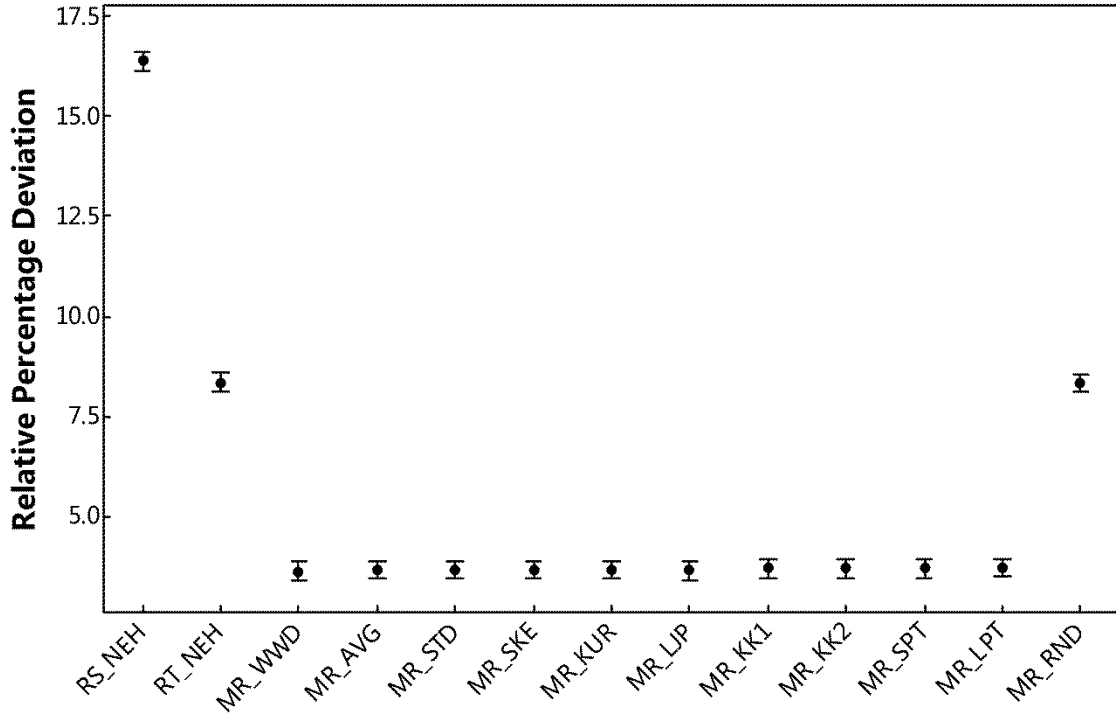


Figure 6 Means plot and 95% HSD intervals of each heuristic on VRF benchmark

Table 3 ARPD values of each heuristic on Taillard test bed with $\alpha=0.4$

Problem	RS_NEH	RT_NEH	MR strategy										
			WWD	AVG	STD	SKE	KUR	LJP	KK1	KK2	SPT	LPT	RND
20×5	19.95	10.35	6.51	10.11	11.39	11.71	11.39	11.71	10.23	9.76	7.36	14.75	9.95
20×10	19.36	11.14	9.46	10.60	9.59	9.51	9.59	9.51	12.16	10.78	11.04	13.49	11.54
20×20	18.09	10.86	3.73	3.73	3.73	3.73	3.73	3.73	3.81	3.81	3.73	3.81	11.05
50×5	11.83	5.28	2.78	4.52	4.92	4.94	5.02	4.92	3.65	3.88	4.11	9.64	5.81
50×10	22.90	11.09	8.24	9.70	9.04	9.08	9.04	9.08	11.43	10.94	8.86	15.55	10.79
50×20	23.00	12.10	9.19	11.43	11.30	11.44	11.30	11.44	12.56	12.62	12.62	13.65	13.44
100×5	10.45	3.77	1.71	2.99	2.68	2.74	2.72	2.67	2.93	3.46	2.14	9.29	3.85
100×10	15.30	7.25	4.61	6.70	6.82	6.75	6.82	6.79	6.68	6.23	5.56	12.46	7.20
100×20	20.16	10.98	8.01	9.93	9.82	9.88	9.82	9.88	10.66	10.31	10.03	13.19	10.75
200×10	12.43	4.11	3.05	4.19	4.25	4.23	4.35	4.22	3.96	3.72	3.03	11.51	4.13
200×20	18.24	8.94	7.03	8.05	8.27	8.49	8.33	8.30	8.31	8.34	7.81	14.3	8.91
500×20	12.83	4.94	4.60	4.94	4.55	4.62	4.57	4.64	4.97	4.83	4.54	12.75	4.93
AVG	17.05	8.40	5.74	7.24	7.20	7.26	7.22	7.24	7.61	7.39	6.73	12.03	8.53

Table 4 ARPD values of each heuristic on small problems of VRF test bed with $\alpha=0.4$

Problem(S)	RS_NEH	RT_NEH	MR strategy										
			WWD	AVG	STD	SKE	KUR	LJP	KK1	KK2	SPT	LPT	RND
10×5	14.25	8.09	2.06	2.98	2.98	2.98	2.98	2.98	3.16	3.16	3.16	3.42	9.32
10×10	10.93	5.54	1.63	1.63	1.63	1.63	1.63	1.63	1.63	1.63	1.63	1.63	5.97
10×15	12.94	7.43	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	7.72
10×20	13.44	6.71	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	6.17
20×5	18.43	7.65	1.53	1.51	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.51	8.37
20×10	22.85	12.95	4.82	4.82	4.82	4.82	4.82	4.82	4.82	4.82	4.82	4.82	13.32
20×15	19.32	11.26	4.29	4.33	4.33	4.33	4.33	4.33	4.33	4.29	4.29	4.33	12.03
20×20	18.16	10.49	4.12	4.12	4.12	4.12	4.12	4.12	4.12	4.12	4.12	4.12	10.52
30×5	14.82	5.16	1.64	1.43	1.41	1.41	1.41	1.41	1.98	1.98	1.86	1.43	4.85
30×10	22.82	12.33	5.26	5.26	5.19	5.19	5.19	5.19	5.18	5.18	5.26	5.18	11.81
30×15	25.97	14.67	5.73	5.83	5.83	5.83	5.83	5.83	5.73	5.73	5.74	5.87	14.01
30×20	19.73	11.67	5.41	5.41	5.41	5.41	5.41	5.41	5.41	5.41	5.41	5.41	11.51
40×5	13.49	5.42	1.20	1.50	1.34	1.34	1.34	1.34	1.74	1.74	1.20	1.70	5.05
40×10	22.31	12.41	4.97	4.97	4.94	4.94	4.94	4.94	4.94	4.97	4.94	4.97	11.97
40×15	22.64	12.13	6.08	6.05	6.02	6.02	6.02	6.02	5.95	5.95	6.11	5.90	12.50
40×20	21.88	13.13	5.16	5.14	5.28	5.28	5.28	5.28	5.28	5.28	5.30	5.14	11.66
50×5	12.14	4.38	1.05	1.06	0.85	0.85	0.85	0.85	1.39	1.39	1.60	1.62	3.61
50×10	20.09	10.40	4.45	4.58	4.47	4.47	4.50	4.47	4.57	4.55	4.56	4.57	10.71
50×15	22.77	12.52	6.38	6.52	6.65	6.65	6.65	6.65	6.52	6.52	6.36	6.62	12.41
50×20	21.56	11.97	5.91	5.96	5.79	5.79	5.79	5.79	5.84	5.84	5.79	5.96	11.26
60×5	12.35	3.59	1.51	1.25	1.18	1.08	1.18	1.08	1.36	1.31	1.89	2.38	3.69
60×10	18.74	10.25	3.81	3.96	4.00	4.00	4.00	4.00	4.24	4.24	4.25	4.02	10.67
60×15	22.73	12.12	5.97	5.79	6.12	6.12	6.12	6.12	5.90	6.04	6.04	6.11	11.86
60×20	20.52	12.06	6.44	6.45	6.43	6.43	6.43	6.43	6.51	6.56	6.53	6.51	13.01
AVG(S)	18.54	9.76	3.87	3.92	3.91	3.91	3.91	3.91	3.99	3.99	4.00	4.03	9.75

Table 5 ARPD values of each heuristic on large problems of VRF test bed with $\alpha=0.4$

Problem(L)	RS_NEH	RT_NEH	MR strategy										
			WWD	AVG	STD	SKE	KUR	LJP	KK1	KK2	SPT	LPT	RND
100×20	20.82	10.75	5.63	5.71	5.74	5.74	5.74	5.74	5.78	5.75	5.62	5.82	10.88
100×40	18.07	9.97	5.48	5.47	5.51	5.51	5.51	5.51	5.53	5.53	5.53	5.53	10.15
100×60	17.00	9.52	4.76	4.95	4.84	4.84	4.84	4.84	4.94	4.98	4.91	4.68	9.03
200×20	17.29	8.37	4.17	4.23	4.36	4.36	4.36	4.34	4.27	4.08	4.22	4.26	8.66
200×40	17.61	9.40	4.74	4.71	4.68	4.68	4.68	4.68	4.90	4.88	4.68	4.74	9.60
200×60	16.05	8.74	4.40	4.55	4.43	4.43	4.43	4.43	4.36	4.30	4.46	4.39	8.72
300×20	14.52	6.79	3.03	3.00	3.19	3.14	3.08	3.14	3.07	2.95	2.96	3.06	6.67
300×40	16.20	8.07	4.02	4.08	4.13	4.12	4.13	4.10	4.09	4.01	4.03	4.12	8.05
300×60	15.30	8.01	3.98	3.93	3.96	4.00	3.96	4.00	4.00	4.02	4.01	4.11	7.84
400×20	13.34	5.77	2.48	2.58	2.46	2.49	2.46	2.48	2.41	2.48	2.43	2.51	5.55
400×40	14.76	7.20	3.54	3.66	3.62	3.68	3.62	3.67	3.66	3.71	3.66	3.55	7.30
400×60	14.51	7.12	3.62	3.56	3.52	3.54	3.57	3.51	3.60	3.63	3.59	3.57	7.20
500×20	12.34	4.86	2.05	2.27	2.06	2.00	2.02	1.95	1.98	2.06	2.01	2.03	4.93
500×40	13.93	6.66	3.09	3.20	3.19	3.11	3.20	3.17	3.25	3.21	3.13	3.11	6.65
500×60	14.03	6.88	3.28	3.12	3.26	3.21	3.23	3.18	3.36	3.16	3.27	3.26	6.90
600×20	10.97	4.27	1.72	1.57	1.74	1.66	1.69	1.67	1.75	1.70	1.69	1.68	4.13
600×40	13.51	6.05	3.07	3.13	3.00	3.02	2.95	3.00	2.95	2.86	3.09	2.95	6.31
600×60	12.99	6.43	2.88	2.93	2.97	2.98	2.96	2.97	2.92	3.02	2.91	2.95	6.52
700×20	10.39	4.09	1.38	1.40	1.46	1.44	1.46	1.44	1.41	1.51	1.41	1.43	4.06
700×40	12.81	5.96	2.71	2.77	2.79	2.69	2.77	2.70	2.82	2.69	2.80	2.68	5.88
700×60	12.78	6.19	2.71	2.75	2.78	2.75	2.78	2.73	2.80	2.78	2.81	2.76	6.08
800×20	9.49	3.79	1.29	1.23	1.30	1.28	1.31	1.31	1.30	1.29	1.30	1.28	3.67
800×40	11.93	5.32	2.60	2.46	2.47	2.53	2.50	2.55	2.45	2.64	2.54	2.56	5.21
800×60	12.13	5.76	2.71	2.71	2.74	2.76	2.79	2.75	2.65	2.70	2.73	2.84	5.84
AVG(L)	14.28	6.92	3.31	3.33	3.34	3.33	3.34	3.33	3.34	3.33	3.32	3.33	6.91
AVG	16.41	8.34	3.59	3.63	3.63	3.62	3.62	3.62	3.66	3.66	3.66	3.68	8.38

Table 6 Paired-samples *t* test on Taillard benchmark

Pairs	Mean	SEM	Lower	Upper	t	Sig.
RS_NEH-RT_NEH	8.64500	.27213	8.10616	9.18383	31.768	.000
RS_NEH-MR_WWD	11.30092	.36875	10.57075	12.03109	30.646	.000
RT_NEH-MR_WWD	2.65592	.26131	2.13850	3.17334	10.164	.000
MR_WWD-MR_AVG	-1.49726	.21128	-1.91561	-1.07891	-7.087	.000
MR_WWD-MR_STD	-1.45275	.21767	-1.88376	-1.02173	-6.674	.000
MR_WWD-MR_SKE	-1.51702	.21634	-1.94539	-1.08865	-7.012	.000
MR_WWD-MR_KUR	-1.47950	.21820	-1.91156	-1.04744	-6.780	.000
MR_WWD-MR_LJP	-1.49760	.21630	-1.92591	-1.06930	-6.924	.000
MR_WWD-MR_KK1	-1.86925	.19663	-2.25859	-1.47990	-9.506	.000
MR_WWD-MR_KK2	-1.64609	.19875	-2.03964	-1.25254	-8.282	.000
MR_WWD-MR_SPT	-.99050	.17683	-1.34065	-.64036	-5.601	.000
MR_WWD-MR_LPT	-6.28837	.35899	-6.99919	-5.57754	-17.517	.000
MR_WWD-MR_RND	-2.78358	.26992	-3.31804	-2.24911	-10.313	.000

Table 7 Paired-samples *t* test on VRF benchmark

Pairs	Mean	SEM	Lower	Upper	t	Sig.
RS_NEH-RT_NEH	8.07032	.12962	7.81563	8.32501	62.263	.000
RS_NEH-MR_WWD	12.82062	.19122	12.44488	13.19636	67.046	.000
RT_NEH-MR_WWD	4.75030	.12072	4.51310	4.98750	39.351	.000
MR_WWD-MR_AVG	-.03690	.02675	-.08945	.01565	-1.380	.168
MR_WWD-MR_STD	-.03684	.02622	-.08836	.01468	-1.405	.161
MR_WWD-MR_SKE	-.03010	.02601	-.08121	.02102	-1.157	.248
MR_WWD-MR_KUR	-.03403	.02629	-.08569	.01764	-1.294	.196
MR_WWD-MR_LJP	-.02807	.02609	-.07934	.02319	-1.076	.283
MR_WWD-MR_KK1	-.07517	.03319	-.14039	-.00995	-2.265	.024
MR_WWD-MR_KK2	-.07178	.03335	-.13731	-.00624	-2.152	.032
MR_WWD-MR_SPT	-.07145	.02987	-.13015	-.01275	-2.392	.017
MR_WWD-MR_LPT	-.09016	.04042	-.16957	-.01075	-2.231	.026
MR_WWD-MR_RND	-4.74069	.11762	-4.97180	-4.50959	-40.307	.000

5. Digital tool for dynamic scheduling

In order to ease of application for these algorithms and automatically obtain the scheduling results, a digital tool has been developed in Matlab R2013b. The interface of the digital tool is shown in Fig. 7. The goal is to set a tight common due date for the new order or to determine if the new order is acceptable or not.

One old order and one new order containing job information should be loaded first respectively. The file format of each order is shown in Fig. 8. In the file, each row is associated with one job. In each row, every pair of data represents the machine index and its processing time for the job. For the case shown in Fig. 8, there are 20 jobs and 5 machines. For the first job, the processing times on the five machines are 54, 79, 16, 66 and 58 respectively. The due date of the old order is necessary and it has to be input. If the due date of the new order is not given, the software tool will automatically set the objective as to find a tight due date. Otherwise, the objective is to determine if the new order should be accepted or not. After inputting all essential variables, one can select the target algorithm for scheduling from provided candidate heuristics such as RS_NEH, RT_NEH, MR_WDD or MR_AVG heuristics etc. as shown in Fig. 9.

Finally, press the button 'Run' and the objective value can be automatically generated as well as the job sequence and acceptance status. If the makespan of the new order is larger than its due date, it will be shown rejected in the box of acceptance status. If no due date is assigned, the acceptance status will be shown 'N/A'. The Gantt chart of the schedule can be generated after pressing 'Output Gantt Chart'. An example is shown in Fig. 10. The best schedule can be chosen after running all candidate heuristics by users.



Figure 7 Interface of the scheduling digital tool

Job ID	Due Date	Operation Time	Seq 1	Seq 2	Seq 3
1 54	2 79	3 16	4 66	5 58	
1 83	2 3	3 89	4 58	5 56	
1 15	2 11	3 49	4 31	5 20	
1 71	2 99	3 15	4 68	5 85	
1 77	2 56	3 89	4 78	5 53	
1 36	2 70	3 45	4 91	5 35	
1 53	2 99	3 60	4 13	5 53	
1 38	2 60	3 23	4 59	5 41	
1 27	2 5	3 57	4 49	5 69	
1 87	2 56	3 64	4 85	5 13	
1 76	2 3	3 7	4 85	5 86	
1 91	2 61	3 1	4 9	5 72	
1 14	2 73	3 63	4 39	5 8	
1 29	2 75	3 41	4 41	5 49	
1 12	2 47	3 63	4 56	5 47	
1 77	2 14	3 47	4 40	5 87	
1 32	2 21	3 26	4 54	5 58	
1 87	2 86	3 75	4 77	5 18	
1 68	2 5	3 77	4 51	5 68	
1 94	2 77	3 40	4 31	5 28	

Figure 8 Order file format



Figure 9 Drop-down menu of algorithms

A case study is conducted to validate the digital tool. The instance TA001 of Taillard benchmark is used where the old order is taken from the first half of the instance while the new order is the second half. The due date of the old order is set as 1087.8, as explained in Section 4. Once loading all job information and inputting necessary variables, i.e. due date of the old order, MR_WWD is selected and run. The completion time, acceptance status, and job sequence can be automatically obtained by the digital tool as well as the Gantt chart of the schedule if it is in need. The results obtained by the digital tool are shown in Fig. 11, which are exactly as expected.

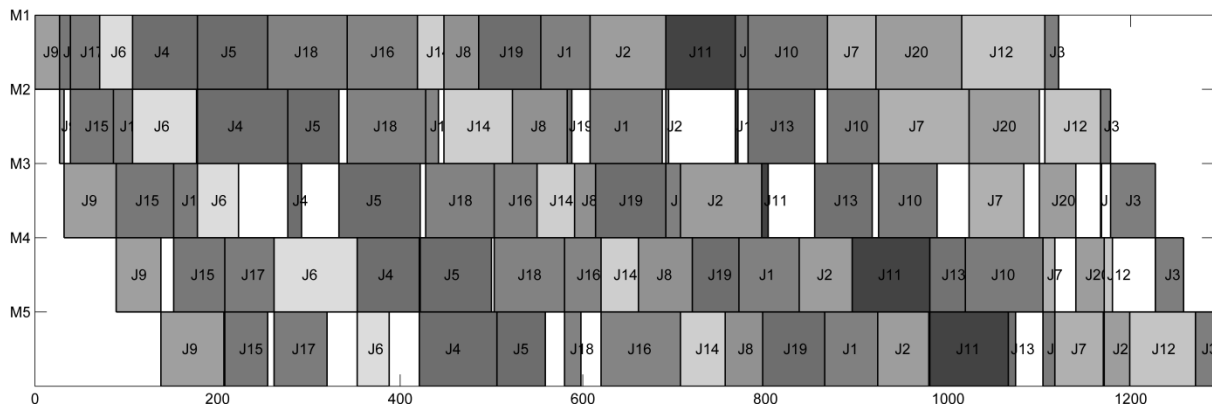


Figure 10 A Gantt chart example

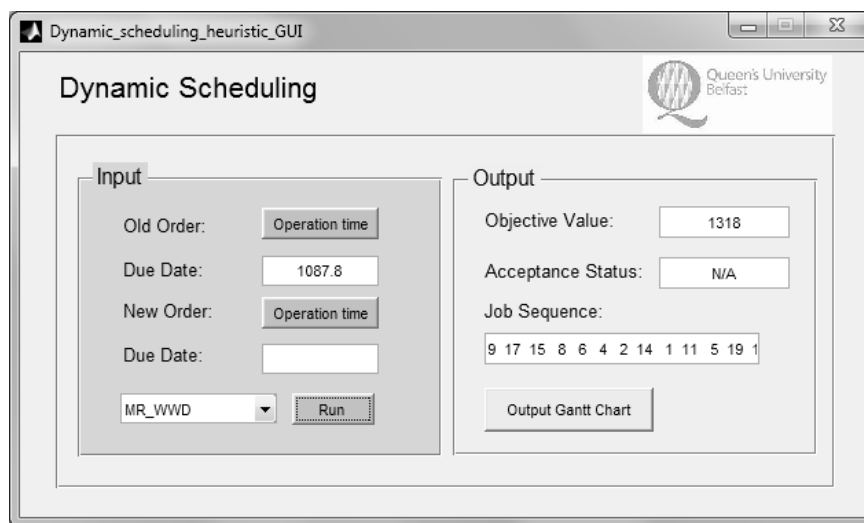


Figure 11 Output of the case study

6. Conclusions

A new strategy named MR strategy is proposed for the permutation flowshop scheduling problem with new order arrivals. By using the new MR strategy, order mixing is achieved and a better schedule is obtained. One new priority rule named as weighted weight distribution (WWD) is proposed. Eleven different priority rules including WWD rule are tested and eleven new heuristics based on the new MR strategy are developed.

By using the new heuristics, the due date setting problem is solved. Tests are conducted on both Taillard and VRF benchmarks including 600 instances in total. Results of evaluations show that the new heuristics, especially MR_WWD are effective in solving the dynamic scheduling problem. The performance is improved by 66.33% and 31.67% comparing to existing RS_NEH and RT_NEH on Taillard benchmark. 78.12% and 56.95% improvements are obtained on VRF benchmark. A digital scheduling tool for dynamic permutation flowshops is developed and validated by implementing these algorithms automatically.

Based on the newly proposed strategy and heuristic algorithms, meta-heuristic methods will be investigated to tackle the dynamic scheduling problem in the near future.

Acknowledgements

The Ph.D. scholarship supported by the Queen's University Belfast UK and China Scholarship Council is acknowledged.

Reference

- Akturk, M. S., and E. Gorgulu. 1999. "Match-up Scheduling under a Machine Breakdown." *European Journal of Operational Research* 112 (1): 81–97.
- Aytug, H., M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. 2005. "Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions." *European Journal of Operational Research* 161 (1): 86–110.
- Bean, J. C., and J. R. Birge. 1986. "Match-up Real-Time Scheduling." In *Proceedings of the Symposium on Real-Time Optimization in Automated Manufacturing Facilities, National Bureau of Standards, Special Publication 724*, 197–212.
- Branke, J., and D. C. Mattfeld. 2005. "Anticipation and Flexibility in Dynamic Scheduling." *International Journal of Production Research* 43 (15): 3103–3129.
- Chaari, T., S. Chaabane, T. Loukil, and D. Trentesaux. 2011. "A Genetic Algorithm for Robust Hybrid Flow Shop Scheduling." *International Journal of Computer Integrated Manufacturing* 24 (9): 821–833.
- Cheng, T. C. E., C. T. Ng, and J. J. Yuan. 2006. "Multi-Agent Scheduling on a Single Machine to Minimize Total Weighted Number of Tardy Jobs." *Theoretical Computer Science* 362 (1–3). Elsevier: 273–281.
- Dong, X., H. Huang, and P. Chen. 2008. "An Improved NEH-Based Heuristic for the Permutation Flowshop Problem." *Computers & Operations Research* 35 (12): 3962–3968.
- El-Bouri, A., S. Balakrishnan, and N. Popplewell. 2008. "Cooperative Dispatching for Minimizing Mean Flowtime in a Dynamic Flowshop." *International Journal of Production Economics* 113 (2): 819–833.
- Feng, Q., B. Fan, S. Li, and W. Shang. 2015. "Two-Agent Scheduling with Rejection on a Single Machine." *Applied Mathematical Modelling* 39 (3–4): 1183–1193.
- Graham, R.L., E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. 1979. "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey." *Annals of Discrete Mathematics* 5: 287–326.
- Herroelen, W., and R. Leus. 2004. "Robust and Reactive Project Scheduling: A Review and Classification of Procedures." *International Journal of Production Research* 42 (8): 1599–1620.
- Johnson, S. M. 1954. "Optimal Two and Three-Stage Production Schedules with Setup Times." *Naval Research Logistics Quarterly* 1: 61–68.
- Kalczynski, P. J., and J. Kamburowski. 2008. "An Improved NEH Heuristic to Minimize Makespan in Permutation Flow Shops." *Computers & Operations Research* 35 (9): 3001–3008.
- Kalczynski, P. J., and J. Kamburowski. 2009. "An Empirical Analysis of the Optimality Rate of Flow Shop Heuristics." *European Journal of Operational Research* 198 (1): 93–101.

- Kianfar, K., S. M. T. Fatemi Ghomi, and B. Karimi. 2009. "New Dispatching Rules to Minimize Rejection and Tardiness Costs in a Dynamic Flexible Flow Shop." *The International Journal of Advanced Manufacturing Technology* 45 (7–8): 759–771.
- Kovalyov, M. Y., A. Oulamara, and A. Soukhal. 2015. "Two-Agent Scheduling with Agent Specific Batches on an Unbounded Serial Batching Machine." *Journal of Scheduling* 18 (4): 423–434.
- Li, M., L. Yao, J. Yang, and Z. Wang. 2015. "Due Date Assignment and Dynamic Scheduling of One-of-a-Kind Assembly Production with Uncertain Processing Time." *International Journal of Computer Integrated Manufacturing* 28 (6): 616–627.
- Liao, L., and C. Huang. 2010. "Tabu Search for Non-Permutation Flowshop Scheduling Problem with Minimizing Total Tardiness." *Applied Mathematics and Computation* 217 (2): 557–567.
- Liu, W., Y. Jin, and M. Price. 2015. "New Priority Rule for NEH Algorithm to Minimize Makespan in Flowshop Scheduling." In *32nd International Manufacturing Conference*. Belfast, UK.
- Liu, W., Y. Jin, and M. Price. 2016. "A New Nawaz–Enscore–Ham-Based Heuristic for Permutation Flow-Shop Problems with Bicriteria of Makespan and Machine Idle Time." *Engineering Optimization* 48 (10): 1808–1822.
- Lodree, E., W. Jang, and C. M. Klein. 2004. "A New Rule for Minimizing the Number of Tardy Jobs in Dynamic Flow Shops." *European Journal of Operational Research* 159 (1): 258–263.
- Moratori, P., S. Petrovic, and J. A. Vázquez-Rodríguez. 2012. "Match-up Approaches to a Dynamic Rescheduling Problem." *International Journal of Production Research* 50 (1): 261–276.
- Moratori, P., S. Petrovic, and A. Vázquez. 2008. "Match-up Strategies for Job Shop Rescheduling." In *New Frontiers in Applied Artificial Intelligence, Lecture Notes in Computer Science*, 5027:119–128. Berlin: Heidelberg: Springer-Verlag.
- Nandi, A., and P. Rogers. 2004. "Using Simulation to Make Order Acceptance/rejection Decisions." *Simulation* 80 (3): 131–142.
- Nawaz, M., E. E. Enscore Jr., and I. Ham. 1983. "A Heuristic Algorithm for the M-Machine, N-Job Flow-Shop Sequencing Problem." *Omega* 11 (1): 91–95.
- O'Donovan, R., R. Uzsoy, and K. N. McKay. 1999. "Predictable Scheduling of a Single Machine with Breakdowns and Sensitive Jobs." *International Journal of Production Research* 37 (18): 4217–4233.
- Perez-Gonzalez, P., and J. M. Framinan. 2009. "Scheduling Permutation Flowshops with Initial Availability Constraint: Analysis of Solutions and Constructive Heuristics." *Computers and Operations Research* 36 (10): 2866–2876.
- Perez-Gonzalez, P., and J. M. Framinan. 2010. "Setting a Common Due Date in a Constrained Flowshop: A Variable Neighbourhood Search Approach." *Computers & Operations Research* 37 (10): 1740–1748.
- Perez-Gonzalez, P., and J. M. Framinan. 2015. "Assessing Scheduling Policies in a Permutation Flowshop with Common Due Dates." *International Journal of Production Research* 53 (19): 5742–5754.
- Rahman, H. F., R. Sarker, and D. Essam. 2015. "A Real-Time Order Acceptance and Scheduling Approach for Permutation Flow Shop Problems." *European Journal of Operational Research*

247 (2): 488–503.

- Rajendran, C., and H. Oliver. 1999. “A Comparative Study of Dispatching Rules in Dynamic Flowshops and Jobshops.” *European Journal of Operational Research* 116: 156–170.
- Safari, E., and S. J. Sadjadi. 2011. “A Hybrid Method for Flowshops Scheduling with Condition-Based Maintenance Constraint and Machines Breakdown.” *Expert Systems with Applications* 38 (3): 2020–2029.
- Taillard, E. 2015. “Summary of Best Known Lower and Upper Bounds of Taillard’s Instances.” Accessed May 22. http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/flowshop.dir/best_lb_up.txt.
- Taillard, E. 1993. “Benchmarks for Basic Scheduling Problems.” *European Journal of Operational Research* 64: 278–285.
- Tamer Unal, A., R. Uzsoy, and A. S. Kiran. 1997. “Rescheduling on a Single Machine with Part-Type Dependent Setup Times and Deadlines.” *Annals of Operations Research* 70 (0): 93–113.
- Vallada, E., R. Ruiz, and J. M. Framinan. 2015. “New Hard Benchmark for Flowshop Scheduling Problems Minimising Makespan.” *European Journal of Operational Research* 240 (3): 666–677.
- Vasiljevic, D., and M. Danilovic. 2015. “Handling Ties in Heuristics for the Permutation Flow Shop Scheduling Problem.” *Journal of Manufacturing Systems* 35: 1–9.
- Wu, C. W., K. N. Brown, and J. C. Beck. 2009. “Scheduling with Uncertain Durations: Modeling β -Robust Scheduling with Constraints.” *Computers & Operations Research* 36 (8): 2348–2356.
- Xia, Y., B. Chen, and J. Yue. 2008. “Job Sequencing and Due Date Assignment in a Single Machine Shop with Uncertain Processing Times.” *European Journal of Operational Research* 184 (1): 63–75.
- Zhu, H., and H. Zhou. 2014. “Single Machine Predictive Scheduling Using Inserted Idle Times.” *Journal of Applied Mathematics* 2014.