University of Groningen

Family-based dispatching with parallel machines

van der Zee, Durk-Jouke

Published in:
International Journal of Production Research

DOI:
[10.1080/00207543.2015.1011289](https://doi.org/10.1080/00207543.2015.1011289)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2015

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):
van der Zee, D-J. (2015). Family-based dispatching with parallel machines. *International Journal of Production Research*, 53(19), 5837-5856. <https://doi.org/10.1080/00207543.2015.1011289>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Family-based dispatching with parallel machines

Durk-Jouke van der Zee*

Faculty of Economics & Business, Department of Operations, University of Groningen, Groningen, The Netherlands

(Received 9 July 2014; accepted 26 November 2014)

Family-based dispatching heuristics seek to lower set-up frequencies by grouping similar types of jobs for joint processing. Hence, job flow times may be improved as less time is spent on set-ups. So far, family-based dispatching with parallel machines received little attention in literature. We address the perceived gap by proposing extensions to existing family-based dispatching heuristics. Main extensions concern improved rules for family priority settings and for coordinating the number of machines in use by a single family. Extended heuristics are tested by an extensive simulation study. Significant performance gains for extended heuristics vs. existing heuristics are found. Performance gains are largest for high set-up to run-time ratios.

Keywords: family-based dispatching; group technology; parallel machines; simulation

1. Introduction

Lead time reduction is a common measure for increasing a manufacturing firm's competitiveness. Its implementation is supposed to benefit both customers – as it improves a firm's abilities in safeguarding a timely response to their demand, and the firm itself – as costs of capital associated with work-in-process and finished goods inventories are reduced. Group technology (GT) principles may be employed to realise lead time reduction. Basically, GT suggests exploiting similarities in product and process design to meet the diversity of customer demand in an economic way. Family-based dispatching builds on GT principles by suggesting to group and jointly dispatch jobs sharing similar requirements with respect to machine set-up. By lowering set-up frequencies family wise, the flow time performance of the machine may be improved.

In this article, we study the application of family-based dispatching heuristics for operational control of parallel machine shops. Our work is motivated by two industrial projects concerning the manufacturing of high tech defence products, and metal plate components and assemblies for cars, power tools and household appliances (Nomden and van der Zee 2008). In deciding on what to produce next on a machine, the operators apply family-based dispatching heuristics. The presence of parallel machines complicates their decision-making, by offering the possibility of job families being assigned to multiple machines at the same time. In principle, allowing for machine pooling may improve flow time performance of the shop, as jobs within the same family are processed earlier. At the same time, machine pooling may imply higher set-up frequencies, suggesting a possible increase of shop flow times. Operators solve the issue by first applying rules for coordinating machine use, i.e. rules restricting the number of machines to be assigned to a family at the same time. In the next step, priorities of remainder families are determined, building on local information on those jobs available in queue. Finally, conventional dispatching rules are applied to determine the job – within the chosen family – to be produced next.

Family-based dispatching heuristics received significant attention in literature, see Pickardt and Branke (2012) for an overview. So far, however, only few authors addressed extension of family-based dispatching heuristics for controlling parallel machine shops. Both Jensen, Malhotra, and Philipoom (1996, 1998) and Kannan (1998) propose to extend family-based dispatching heuristics developed for the single machine case, by adding alternative rules aiming to restrict the number of machines in use by a single family. Respective rules allow families to gain access to at most one machine, unless there are no competing families present in queue. In addition, Jensen, Malhotra, and Philipoom (1998) suggest to adjust, i.e. lower, family priority according to the number of machines the respective family gained access to.

Starting from three main observations, we feel that existing heuristics may be significantly improved. Firstly, we found how rules for machine coordination have been developed in isolation, i.e. they have not been compared for their

*Email: d.j.van.der.zee@rug.nl

performance. Secondly, many promising heuristics – as they have been developed for the single machine case in recent years – have not been considered for their use for controlling parallel machines. Thirdly, in priority setting, existing heuristics do not account for those jobs being withdrawn from the queue if other machines assigned to the family become available during set-up. Clearly, taking due notice of operations being executed on alternative machines may improve shop flow time performance.

Starting from the above observations, we distil (i) a potential for improving current heuristics and (ii) a need for more extensive heuristics testing. We seek to exploit the perceived potential by extending heuristics that have been developed for addressing the single machine case in recent years. More specifically, we address priority setting for those families already having access to one or multiple machines. The need for more extensive heuristics testing is addressed by comparing alternative rules developed for machine coordination. A simulation study is designed to demonstrate the potential of existing and extended heuristics for a large variety of shop configurations in terms of mean flow times per job.

The paper is structured as follows. In Section 2, we review existing heuristics for family-based dispatching. Next, in Section 3, the shop environment is described in detail. Furthermore, a decision framework is used to typify existing and extended heuristics for family-based dispatching. In Section 4, we propose extensions to existing heuristics. Existing and extended heuristics are evaluated by a simulation study (Sections 5 and 6). Finally, main conclusions are summarised in Section 7.

2. Literature review

Family-based dispatching heuristics suggest to group and jointly dispatch jobs sharing similar requirements with respect to machine set-up. By lowering set-up frequencies, family-wise shop performance may be improved both with respect to the timeliness of its operations – as less time is spent on set-ups – and cost – as less resources, i.e. material, labour and capacity are required for readying a machine for producing a product (Groover 2008; Hopp and Spearman 2008; Ciavotta, Meloni, and Pranzo 2013). Note that in many cases, set-up times should not be used as a surrogate for set-up costs in the modelling of scheduling operations, due to the characteristics of the underlying cost function (Ciavotta, Meloni, and Pranzo 2013).

Their explicit focus on set-up efficiencies distinguishes family-based heuristics from common dispatching rules. Similar to common dispatching rules, family-based dispatching heuristics allow for a dynamic approach towards shop control, by relying on limited information, allowing for regular updates of the job list and providing timely answers (McKay, Safayeni, and Buzacott 1988; Stuber 1998). As such, they answer to practical needs – that often cannot be met by analytic approaches, due to problem size and complexity (Allahverdi, Gupta, and Alsdowaisan 1999; Allahverdi et al. 2008).

In a recent survey, Pickardt and Branke (2012) show how heuristics addressing family set-ups may be classified into three categories: purely set-up oriented heuristics, composite heuristics and (purely) family-based heuristics. Our literature review will start from this categorisation. We consider heuristics adopting the minimisation of mean flow time as an objective. Relevance of minimising flow times is related to, among others, improvement of customer responsiveness, maintaining flexibility, improvement of product quality, less need for relying on forecasts and making better forecasts (Hopp and Spearman 2008). See Mosier, Elvers, and Kelly (1984), Mahmoodi and Dooley (1991), Ponnambalam, Aravindan, and Reddy (1999), and Pickardt and Branke (2012) for heuristics addressing due date related criterions. Given the availability of many heuristics, we restrict our discussion to those heuristics that indicated good performance in previous research. See Frazier (1996), Shambu, Suresh, and Pegels (1996), and Pickardt and Branke (2012) for a more complete overview of family-based dispatching heuristics.

The primary objective of *purely set-up oriented heuristics* is to minimise machine time spent on set-ups. A typical heuristic is the short set-up time rule (SST), prioritising the job requiring least set-up time (Gavett 1965). Not surprisingly, as among others, local information on job processing times is not accounted for in job priority setting; Pickardt and Branke (2012) observe that in general, purely set-up oriented heuristics will not suffice in case of a flow time criterion.

Composite heuristics build on a priority index, which also accounts for set-up times. Several heuristics fit in this category. Here, we mention the shortest normalised setup and processing time heuristic (SNSPT), as proposed by Kochhar and Morris (1987) for controlling a flexible flow line. The associated priority index, i.e. $\alpha \frac{s_j}{\bar{s}} + \beta Q \frac{p_{ij}}{\bar{p}}$, sums processing time for job i in family j (p_{ij}) and family setup time (s_j), after being (1) divided by their respective mean values in the queue ($\bar{p}$, $\bar{s}$), and (2) weighted (α , β , Q). Q is -1 if the input buffer of the next workstation is more than $\gamma\%$ full and $+1$ otherwise. For details on tuning α , β , γ see Kochhar and Morris (1987). By means of simulation, Pickardt and Branke

(2012) show how SNSPT outperforms alternative composite heuristics, given a specific setting of weights. Also, it performs well relative to a subset of purely family-based heuristics.

Whereas *purely set-up oriented heuristics* and *composite heuristics* seek to realise set-up efficiencies by considering contributions made by each next job, *purely family-based heuristics* suggest the creation and dispatching of process batches, encompassing multiple jobs belonging to the same family. By allowing for batch formation, purely family-based heuristics seek to *further* increase set-up efficiencies. Purely family-based heuristics are typified by three ordered decisions (Mosier, Elvers, and Kelly 1984):

- (a) *Switching moment*: When to select a new family of jobs for servicing.
- (b) *Family type selection*: Which of the families to process next – assuming the decision in (a) has been made.
- (c) *Job sequencing*: Which job is to be selected from the chosen family.

Heuristics' choice of *switching moments* restricts decision-making to pre-specified moments, meeting specific demands on shop status. Two main classes of heuristics may be distinguished: exhaustive and non-exhaustive heuristics (Mahmoodi and Dooley 1991). Exhaustive heuristics only allow decision-making at the moment the queue of jobs for the current family is exhausted. Non-exhaustive heuristics specify a truncation mechanism for deciding on the switching of families. An early example of a non-exhaustive heuristic is ECON (Mosier, Elvers, and Kelly 1984; Ruben, Mosier, and Mahmoodi 1993). It allows for a switching decision after each job being completed. Other heuristics allow for limited truncation. For example, Russell and Philipoom (1991) suggest switching when either a specified amount of time has elapsed or the family is exhausted. Furthermore, Van der Zee (2010) proposed the MASP_AD and MASP_HY heuristics. These heuristics allow the operator to restrict batch size for a family. No single heuristic – either exhaustive or non-exhaustive – has shown overall best performance. The right choice of heuristic tends to be dependent on shop characteristics (Pickardt and Branke 2012). For example, simulation results indicate how the aforementioned non-exhaustive MASP_AD and MASP_HY heuristics may outperform exhaustive heuristics in case of moderate and high variances of processing and set-up times. In turn, the exhaustive MASP (Russell and Philipoom 1991) and MAS (Nomden, van der Zee, and Slomp 2008) heuristics perform well for low variances of processing and set-up times (Van der Zee 2010).

At each switching moment, queue contents is inspected and next priorities are set for all available families to enable *family type selection* (b). Basically, two generations of priority rules may be distinguished. The first generation adopts common dispatching rules like first come first serve (FCFS) or shortest processing time (SPT) for setting family priority. For example, the well-known FCFAM heuristic (Flynn 1987) extends the FCFS rule by assigning highest priority to the family containing the job that arrived earliest. A second generation of priority rules suggests somewhat more informed decision-making by relating family priority to the concept of the weighted shortest processing time rule (WSPT). This well-known dispatching rule sequences jobs by considering the work load for a job relative to its weight, see, for example, Pinedo (1995). The exhaustive MASP (Russell and Philipoom 1991) and MAS heuristics (Nomden, van der Zee, and Slomp 2008) provide important and well-performing examples of such rules. MASP computes work load as the sum of the family set-up time and cumulative processing times of those jobs available in queue, i.e. the process batch, and sets weight equal to the family queue length. MAS is similar to MASP; however, it restricts work load to family set-up time, thereby stressing set-up efficiencies resulting from larger process batches.

A final step in executing a family-based heuristic concerns the decision on *job sequencing* (c). Family-based dispatching heuristics typically rely on common dispatching rules for making this decision. Examples include FCFS, SPT, earliest due date, etc. See Blackstone, Phillips, and Hogg (1982), Jayamohan and Rajendran (2000), Mizrak and Bayhan (2006), and Sarin, Varadarajan, and Wang (2011) for overviews.

Use of family-based dispatching heuristics for controlling parallel machine shops has been studied by few authors. Chern and Liu (2003) suggest the use of family-based dispatching heuristics for controlling parallel wafer steppers in manufacturing integrated circuits. They adopt heuristics that have been developed for the single machine case, without adjusting for shop characteristics. Motivated by, among others, a case example in semiconductor manufacturing Jensen, Malhotra, and Philipoom (1996, 1998) and Kannan (1998) consider control of job shop environments with workstations that consist of parallel machines. In an attempt to balance benefits of machine pooling and its effect on set-up efficiencies, they suggest to extend existing heuristics by adding alternative rules aiming to restrict the number of machines in use by a single family. Essentially, proposed rules allow families to gain and/or keep access to at most one machine, as long as a competing family – having no access to a machine – is available in queue. In addition, Jensen, Malhotra, and Philipoom (1998) suggest to adapt family-based dispatching heuristics by adjusting, i.e. lowering, family priority, according to the number of machines it gained access to. Nomden and van der Zee (2008) employ the heuristics

proposed by Jensen, Malhotra, and Philipoom (1996) in studying operational performance of parallel machine shops given various levels of routing flexibility.

With respect to those heuristics developed for controlling parallel machine shops, we observed how:

- Coordinating the number of machines in use by a single family facilitates balancing benefits of machine pooling and its effects on set-up frequencies. However, rules for machine coordination were not compared for their performance.
- More informed and better performing, second generation rules for family priority setting have not been employed for developing heuristics addressing parallel machine shops.
- Family priority setting for existing heuristics does not account for those jobs in queue that may start processing at alternative machines during machine set-up.

Starting from these observations, in this article, we:

- Propose two extensions to existing family-based dispatching heuristics concerning family priority setting. A first extension concerns replacing first generation priority rules by second generation priority rules. Secondly, more specifically, we address priority setting for those families that employ one or multiple machines at the same time.
- Compare existing rules for machine coordination for their benefits for shop flow time performance.

3. Shop description and decision framework

3.1 Shop description

The shop consists of M identical machines in parallel. Incoming jobs are stored in the buffer until the time they are released to the machine and processed. Alternative choices of job lot sizes are possible. The buffer is assumed to have an unlimited storage capacity. Each job belongs to a certain family $j \in J$. Total number of jobs in queue for each family j equals q_j . Each job family requires a specific, sequence dependent, machine set-up. This so-called major set-up is associated with a set-up time s_{j_0j} . Length of the set-up time is determined by the current set-up – for family j_0 – and the required set-up for family j . Obviously, $s_{j_0j} = 0$ for $j = j_0$. Job-related, so-called minor set-ups are assumed to be included in job processing times ($p_{i,j}$), with i identifying individual jobs within the set of jobs $I_j(t_0)$ available in queue for a family j at a switching moment (t_0), i.e. a moment the operator may decide to switch families. For each family, jobs in queue are assumed to be ordered according to their processing times, i.e. $p_{1,j} < p_{2,j} < \dots < p_{n,j}$, with $n = |I_j(t_0)|$.

3.2 Decision framework

In this section, we define a decision framework. According to the framework, a family-based dispatching heuristic determines the job to be produced next, as a net effect of making three ordered decisions (Mosier, Elvers, and Kelly 1984). Each respective decision is supported by one or multiple rules. Each heuristic is characterised by a unique set of rules. Here, we will discuss main implementation issues, using existing heuristics as vehicles for discussion. As an objective we consider the minimisation of mean flow time per job in the long run. Given N processed jobs, mean flow time per job ($\bar{F}$) is defined as:

$$\bar{F} = \frac{\sum_{j \in J} \sum_{i \in I_j} f_{i,j}}{N} \quad \text{with} \quad (1)$$

$$f_{i,j} = w_{i,j} + p_{i,j}$$

$$N = \sum_{j \in J} \sum_{i \in I_j} 1$$

In computing flow time for a job i belonging to family j ($f_{i,j}$), we distinguish between waiting time ($w_{i,j}$) and job processing time ($p_{i,j}$) for all jobs $i \in I_j$ becoming available over time. Waiting time concerns job delay until processing at the machine. Set-up time (s_{j_0j}) is included in waiting time.

3.2.1 Switching moment: when to select a new family of jobs for servicing

Typically, family-based dispatching heuristics restrict choice of switching moments, i.e. moments an operator may decide to select a new family of jobs for servicing. For example, exhaustive heuristics only allow for a new dispatching decision if the queue for the current family is exhausted. By maximising batch size in this way, respective heuristics

seek to increase set-up efficiencies. Other, so-called, non-exhaustive heuristics do force alternative requirements on decision-making. For example, switching moments for non-exhaustive heuristics may be related to static parameters like time fences for processing a specific family (Russell and Philipoom 1991), process batch sizes (Van der Zee 2010) and each job being completed (Mosier, Elvers, and Kelly 1984). Note that machine characteristics may refine choice of switching moments.

3.2.2 Family type selection: which of the families to process next

In order to select the family to process next, each family is assigned a priority. We distinguish between two generations of family priority rules. The first generation of rules relates family priority setting to common dispatching rules, like FCFS or SPT. For example, the so-called FCFAM heuristic (Flynn 1987) prioritises families by considering the *earliest arrival moment* ($t_{i,j}$) among the jobs i available at the switching moment (t_0) for a family $j \in J$:

$$J_{FCFAM}^* = \arg \min_{i \in I_j(t_0), j \in J} t_{i,j} \quad (2)$$

A second generation of rules relates family priority setting to the well-known concept of the WSPT, see, for example, Pinedo (1995). A good illustration of such a rule for the single machine case is provided by the MASP heuristic (Russell and Philipoom 1991). MASP relates system set-up to the choice of family j^* for which a minimum weighted work load is foreseen. Hereby, work load is estimated by the sum of family set-up time ($s_{j_0,j}$) and cumulative processing times ($\sum_{i \in I_j(t_0)} p_{i,j}$) for those jobs within family j that are available at the switching moment. Weights are related to family queue length (q_j):

$$J_{MASP}^* = \arg \min_{j \in J: q_j > 0} \frac{s_{j_0,j} + \sum_{i \in I_j(t_0)} p_{i,j}}{q_j} \quad (3)$$

Alternative priority rules following the WSPT scheme differ from MASP by (1) relating their definition of work load to either set-up time or cumulative processing time, and (2) their choice of weight. Alternative choices of weight considered are 1 (equal weight), compare SWORK (Nomden, van der Zee, and Slomp 2008) and process batch size $k = 1 \dots q_j$, compare MASP_AD and MASP_HY (Van der Zee 2010).

3.2.3 Job sequencing: which job is to be selected from the chosen family

Job sequencing is realised by employing *conventional dispatching rules*, such as, for example, FCFS and SPT.

4. Extending heuristics – family-based dispatching with parallel machines

In this section, we discuss existing and new heuristics for family dispatching with parallel machines according to their decision structure, see Section 3. An overview of the heuristics introduced in this section can be found in the Appendix 1.

4.1 Existing heuristics

With the exception of the M-SPT heuristic (Jensen, Malhotra, and Philipoom 1998), all existing heuristics focus on flow time minimisation, see Table 1. The M-SPT heuristic has been included, as it allows for adjusting priority setting for those families having access to multiple machines. Note that we named the heuristic proposed by Kannan (1998) RepetitiveLots-FCFS.

4.1.1 Switching moment: when to select a new family of jobs for servicing

Except for RepetitiveLots-FCFS, all heuristics are exhaustive; i.e. they only allow for switching families if the queue for the current family is empty. Alternatively, RepetitiveLots-FCFS seeks to coordinate machine use, by running a check on each machine after completing a job: in case the current family has access to multiple machines, and one or multiple families in queue do not have such access, a new choice of family is made among the latter families; otherwise, a new job is selected from the current family.

4.1.2 Family type selection: which of the families to process next

All existing heuristics rely on first generation rules for family priority setting, compare Section 3.2. Two categories of rules may be distinguished. The first category of rules, as implemented for FCFS-F and SRPT-F, does not adjust family priorities for shop characteristics, i.e. presence of multiple machines. Essentially, FCFS-F boils down to the FCFAM heuristic, see Section 3.2. SRPT-F is quite similar to FCFS-F, except for the fact that the SPT rule is applied in selecting a next family. The second category of priority rules, as implemented for FCFS-L, APT-L, M-SPT, RepetitiveLots-FCFS and FCFAM-C, does tailor family priorities to shop specifics, in two ways by:

- Coordinating machine use by pre-selecting families $j \in J$ that have no access to a machine yet ($m(j) = 0$). It does so by adjusting, i.e. lowering, priorities for families having access to one or multiple machines ($m(j) \geq 1$) by adding a big number (M), compare FCFS-L, APT-L, RepetitiveLots-FCFS, FCFAM-C.
- Adjusting priorities for families already having access to multiple machines by relating use of the idle machine under consideration to a ‘less favourable’ subset of jobs available for respective families at the switching moment (t_0), compare M-SPT (Jensen, Malhotra, and Philipoom 1998). If a family already has access to one machine, the respective subset of jobs concerns the second half of the queue. Likewise, in case two machines are in use by the family, the last third of the jobs within the family are considered in priority setting, and so on.

Table 1. Existing family-based dispatching heuristics – decision structure.

Heuristic	Switching strategy		Priority index	Job sequencing
	Exhaustive	Batch size		
FCFS-L (Jensen, Malhotra, and Philipoom 1996)	Yes	$k = q_j$	$t_{i,j} + x$ with $i \in I_j(t_0)$ $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	FCFS
APT-L (Jensen, Malhotra, and Philipoom 1996)	Yes	$k = q_j$	$\frac{\sum_{i \in I_j(t_0)} p_{i,j}}{q_j} + x$ with $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	FCFS
M-SPT (Jensen, Malhotra, and Philipoom 1998)	Yes	$k = q_j$	$p_{i,j}$ with $i = \begin{cases} 1 & \text{if } m(j) = 0 \\ 1 + \left\lfloor q_j * \frac{m(j)}{m(j) + 1} \right\rfloor & \text{else} \end{cases}$	Critical ratio
RepetitiveLots-FCFS (Kannan 1998)	Hybrid	$k = q_j$	$q_j + x$ with $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	FCFS
FCFS-F (Chern and Liu 2003)	Yes	$k = q_j$	$t_{i,j}$ with $i \in I_j(t_0)$	FCFS
SRPT-F (Chern and Liu 2003)	Yes	$k = q_j$	$p_{i,j}$ with $i \in I_j(t_0)$	SPT
FCFAM-C (Nomden, van der Zee, and Slomp 2008)	Yes	$k = q_j$	$t_{i,j} + x$ with $i \in I_j(t_0)$ $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	SPT

Table 2. Proposed family-based dispatching heuristics – decision structure.

Heuristic	Switching strategy		Family priority index	Job sequencing
	Exhaustive	Batch size		
FCFAM_PAR_A	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$t_{i,j} + x$ with $i \in \tilde{I}_j(t_0 + s_{j_0,j})$ $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	SPT
FCSPT_PAR_A	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$p_{i,j} + x$ with $i \in \tilde{I}_j(t_0 + s_{j_0,j})$ $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	SPT
MJ_PAR_A	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$\tilde{q}_j(t_0 + s_{j_0,j}) + x$ with $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	SPT
MASP_PAR	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$s_{j_0,j} + \sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}$ $\frac{\sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}}{\tilde{q}_j(t_0 + s_{j_0,j})}$	SPT
MAP_PAR_A	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$\frac{\sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}}{\tilde{q}_j(t_0 + s_{j_0,j})} + x$ with $x = \begin{cases} M & \text{if } m(j) \geq 1 \\ 0 & \text{else} \end{cases}$	SPT
MAS_PAR	Yes	$k = \tilde{q}_j(t_0 + s_{j_0,j})$	$\frac{s_{j_0,j}}{\tilde{q}_j(t_0 + s_{j_0,j})}$	SPT
MASP_AD_PAR	No	$k^{\min} \leq k \leq \tilde{q}_j(t_0 + s_{j_0,j})$	$s_{j_0,j} + \sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}$ $\frac{\sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}}{k}$ with $k^{\min} = \text{round}(s_{j_0,j}/(1/\lambda_T - \bar{p}))$	SPT
MASP_HY_PAR	Hybrid	$k^{\min} \leq k^* \leq \tilde{q}_j(t_0 + s_{j_0,j})$	$s_{j_0,j} + \sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}$ $\frac{\sum_{i \in \tilde{I}_j(t_0 + s_{j_0,j})} p_{i,j}}{k^*}$ with $k^{\min} = \text{round}(s_{j_0,j}/(1/\lambda_T - \bar{p}))$	SPT
SNSPT_PAR	No	$k = 1$	$\alpha \frac{s_{j_0,j}}{s} + \beta Q \frac{p_{i,j}}{p} + x$ with $i \in \tilde{I}_j(t_0 + s_{j_0,j})$ $\alpha = \beta = Q = 1$	SPT

4.1.3 Job sequencing: which job is to be selected from the chosen family

Heuristics suggest to sequence jobs according to FCFS or SPT.

4.2 New heuristics

In this section, we propose new heuristics for family-based dispatching with parallel machines, see Table 2. All new heuristics result from extensions of existing heuristics. Two categories of existing heuristics have been included in our study, i.e. heuristics addressing the parallel machine case, see Section 4.1, and heuristics that showed good performance for the single machine case (Nomden, van der Zee, and Slomp 2008; Van der Zee 2010; Pickardt and Branke 2012). By studying nine heuristics in total, we strive for a good coverage of a wide range of shop parameters, knowing that, so far, there is no single heuristic showing overall best performance (Frazier 1996; Van der Zee 2010; Pickardt and Branke 2012).

Four heuristics, i.e. FCFAM_PAR_A, FCSPT_PAR_A, MJ_PAR_A and MAP_PAR_A, have been developed starting from existing heuristics addressing the parallel machine case, i.e. FCFAM-C, SRPT-F, RepetitiveLots-FCFS and APT-L, see Tables 1 and 2. Note that our choice of names for the new heuristics reflects (i) the name of the heuristic as it has been developed for the single machine case (e.g. MAP), (ii) the fact that family priorities are adjusted for the PAR and (iii) the rule applied for machine coordination (A). See below for details on the way family priorities and rules for machine coordination are implemented. FCFS-L and FCFS-F (compare Table 1) have not been included in our study because of their similarity to selected heuristics, i.e. FCFAM-C and FCFAM. Likewise, M-SPT is not considered for extension towards the parallel machine case due to its lack of focus on flow time minimisation. Its choice of family priority rule is, however, evaluated in our simulation study, see Sections 4.1, 5 and 6.

Family-based dispatching heuristics showing good performance for the single machine case are SNSPT (Kochhar and Morris 1987), MASP (Russell and Philipoom (1991), MAS (Nomden, van der Zee, and Slomp 2008), MASP_AD (Van der Zee 2010) and MASP_HY (Van der Zee 2010). New heuristics that resulted from extensions to respective heuristics are MASP_PAR, MAS_PAR, MASP_AD_PAR, MASP_HY_PAR and SNSPT_PAR.

4.2.1 Switching moment: when to select a new family of jobs for servicing

As far as the switching strategy is concerned, three categories of heuristics are considered, i.e. exhaustive heuristics (FCFAM_PAR_A, FCSPT_PAR_A, MJ_PAR_A, MASP_PAR, MAP_PAR, MAS_PAR), non-exhaustive heuristics (MASP_AD_PAR, SNSPT_PAR) and hybrid heuristics (MASP_HY_PAR). Non-exhaustive heuristics studied allow switching families at the moment a process batch of pre-specified size has been completed. The respective batch size is pre-determined as 1 (SNSPT_PAR) or the outcome of family type selection (MASP_AD_PAR), see below. MASP_HY_PAR allows for a hybrid switching strategy. In principle, it follows MASP_AD_PAR logic. However, in case the pre-specified batch size equals family queue length at the switching moment, it allows for an exhaustive policy in processing the respective family.

4.2.2 Family type selection: which of the families to process next

As far as rules for family priority setting are concerned, we propose two extensions:

- For all heuristics studied, priorities for families having access to one or multiple machines are adjusted, accounting for those jobs that start processing at other machines during the new machine set-up. Only those jobs available after completing the set-up ($i \in \tilde{I}_j(t_0 + s_{j_0,j})$) are considered in family priority setting. Queue contents for a family after machine set-up is estimated by building a schedule for the period up to the moment the machine set-up has been completed ($t_0 + s_{j_0,j}$). Scheduling involves those jobs available at the switching moment (t_0) and the machines already assigned to the family. Note that adjustment of family priorities involves both work load estimates and their weighting. For example, for MASP_PAR work load is adjusted by considering processing times for ($p_{i,j}$) for those jobs available after set-up ($i \in \tilde{I}_j(t_0 + s_{j_0,j})$). Likewise, weight for MASP_PAR is adapted by considering family queue length after set-up ($\tilde{q}_j(t_0 + s_{j_0,j})$).
- For a subset of heuristics, the rule for machine coordination as suggested by Jensen, Malhotra, and Philipoom (1996, 1998) and Kannan (1998) is adopted, see Section 4.1. According to this rule, families in queue having no access to a machine are pre-selected. The rule is implemented by lowering priority for families $j \in J$ that already have access to machines ($m(j) \geq 1$) by adding M, i.e. a big number. This rule is implemented for FCFAM_PAR_A, FCSPT_PAR_A, MJ_PAR_A and MAP_PAR_A.

Table 3. (Continued).

Experimental factors

Each family gains access to at most one machine, unless there are no competing families in queue.	<i>FCFAM-C</i>	FCFAM_PAR_A
	<i>FCFS-L</i>	FCSPT_PAR_A
	<i>RepetitiveLots-</i>	MJ_PAR_A
	<i>FCFS</i>	MASP_PAR_A
	<i>APTL</i>	MAP_PAR_A
		MAS_PAR_A
		SNSPT_PAR_A
		MASP_AD_PAR_A
		MASP_HY_PAR_A
		FCFAM_PAR_AU
Each family gains access to at most one machine, unless there are no competing families in queue; Machine use is reconsidered after each job being completed		FCSPT_PAR_AU
		MJ_PAR_AU
		MASP_PAR_AU
		MAP_PAR_AU
		MAS_PAR_AU
		SNSPT_PAR_AU
		MASP_AD_
		PAR_AU
		MASP_HY_
		PAR_AU

^aMean set-up time divided by mean processing time.^bMean job arrival intervals have been determined for alternative settings of the number of families, set-up to runtime ratio and work load, assuming FCFAM is chosen as the family based dispatching rule.^cSRPT-F and FCSPT are identical.

Table 4. Mean job flow times for existing and proposed heuristics.

Shop configuration				Existing heuristics						Proposed heuristics								
# Machines	# Job Families	S/R	Work Load (%)	FCFS-F	RepetitiveLoss-FCFS	APT-L	FCFS-L	FCFS-C	FCFAM-FCFAM	MJ_PAR_A	MAP_PAR_A	FCFSPT_PAR_A	MASP_PAR	MAS_PAR	MASP_HY_PAR	MASP_AD_PAR	SNSPT_PAR	
4	4	0.25	75	1.6025	1.6086	1.5844	1.5807	1.5017	1.4924	1.4637	1.4656	1.4608	1.4597	1.4565	1.4426	1.4363	1.4298	1.4264
4	4	0.25	90	2.5431	2.5893	2.4898	2.4537	2.0729	2.0551	2.0000	1.9966	1.9921	1.9885	2.0015	1.9825	1.9431	1.9197	1.8679
4	4	0.50	75	1.7315	1.7447	1.7096	1.7032	1.6406	1.6197	1.5218	1.5176	1.5192	1.5165	1.5035	1.4931	1.4922	1.4938	1.4895
4	4	0.50	90	2.4956	2.5109	2.4265	2.3811	2.1191	2.0523	1.9362	1.9248	1.9412	1.9315	1.9198	1.8860	1.8863	1.8888	1.8711
4	4	1.00	75	2.0451	2.0524	2.0014	1.9813	1.9522	1.9033	1.6060	1.6049	1.6067	1.6047	1.5857	1.5780	1.5812	1.5897	1.5907
4	4	1.00	90	2.7073	2.7141	2.5981	2.5283	2.3956	2.2813	1.9440	1.9427	1.9479	1.9461	1.9104	1.8747	1.8886	1.9091	1.9169
4	8	0.25	75	1.6539	1.6244	1.5994	1.6214	1.5525	1.5622	1.5442	1.5362	1.5187	1.5194	1.4926	1.4807	1.4556	1.4497	1.4436
4	8	0.25	90	2.6097	2.4403	2.4238	2.4437	2.2562	2.1933	2.1674	2.1530	2.1178	2.1140	2.0609	1.9771	1.8966	1.8799	1.8466
4	8	0.50	75	1.8450	1.7837	1.7652	1.7770	1.7528	1.7242	1.6535	1.6473	1.6412	1.6391	1.5845	1.5551	1.5511	1.5516	1.5545
4	8	0.50	90	2.6547	2.4218	2.4374	2.4218	2.3736	2.2336	2.1507	2.1265	2.1395	2.1242	2.0077	1.8942	1.8833	1.8828	1.8763
4	8	1.00	75	2.2658	2.1507	2.1419	2.1308	2.1834	2.0849	1.8787	1.8735	1.8776	1.8739	1.7675	1.7366	1.7448	1.7498	1.7636
4	8	1.00	90	3.0621	2.7190	2.7419	2.7006	2.8155	2.5506	2.3284	2.3085	2.3347	2.3224	2.0826	1.9962	2.0113	2.0258	2.0574
4	16	0.25	75	1.7065	1.6683	1.6031	1.6747	1.5779	1.6370	1.6295	1.6191	1.5572	1.5548	1.5089	1.4941	1.4599	1.4552	1.4536
4	16	0.25	90	2.7417	2.5211	2.4147	2.5590	2.3497	2.4458	2.4182	2.3601	2.2007	2.2237	2.0424	1.9409	1.8388	1.8271	1.7956
4	16	0.50	75	1.9442	1.8586	1.8170	1.8649	1.8282	1.8302	1.7944	1.7831	1.7452	1.7411	1.6337	1.5825	1.5828	1.5803	1.5857
4	16	0.50	90	2.9137	2.5811	2.5779	2.6463	2.6066	2.5229	2.4738	2.4174	2.3838	2.3540	2.0126	1.8700	1.8607	1.8594	1.8589
4	16	1.00	75	2.4564	2.2844	2.2756	2.3072	2.3401	2.2650	2.1470	2.1368	2.1135	2.1048	1.8730	1.8209	1.8334	1.8634	1.8634
4	16	1.00	90	3.5061	2.9788	3.0645	3.0421	3.2274	2.9504	2.8298	2.7649	2.7913	2.7437	2.1554	2.0388	2.0633	2.0662	2.1168
8	4	0.25	75	1.2802	1.2764	1.2784	1.2765	1.2415	1.2437	1.2009	1.1999	1.1984	1.1965	1.1977	1.1915	1.1932	1.1890	1.1888
8	4	0.25	90	1.7522	1.7408	1.7841	1.7324	1.5388	1.5445	1.4721	1.4640	1.4674	1.4583	1.4593	1.4430	1.4475	1.4402	1.4273
8	4	0.50	75	1.3851	1.3826	1.3926	1.3808	1.3533	1.3486	1.2216	1.2208	1.2211	1.2204	1.2192	1.2153	1.2159	1.2166	1.2161
8	4	0.50	90	1.7899	1.7755	1.8389	1.7692	1.6272	1.6224	1.4289	1.4203	1.4315	1.4225	1.4196	1.4048	1.4124	1.4120	1.4116
8	4	1.00	75	1.6124	1.6072	1.6235	1.5974	1.5713	1.5627	1.3746	1.3714	1.3740	1.3732	1.3665	1.3495	1.3609	1.3709	1.3709
8	4	1.00	90	2.0050	1.9811	2.0663	1.9696	1.8593	1.8424	1.3465	1.3314	1.3740	1.3732	1.3665	1.3645	1.3667	1.3705	1.3766
8	8	0.25	75	1.3215	1.3150	1.3040	1.3054	1.2850	1.2822	1.2450	1.2438	1.2418	1.2411	1.2296	1.2161	1.2155	1.2152	1.2130
8	8	0.25	90	1.8153	1.8057	1.7965	1.7435	1.6414	1.6081	1.5465	1.5322	1.5398	1.5366	1.5194	1.4590	1.4593	1.4516	1.4331
8	8	0.50	75	1.4616	1.4549	1.4460	1.4334	1.4303	1.4114	1.3014	1.2977	1.2976	1.2970	1.2810	1.2639	1.2678	1.2691	1.2680
8	8	0.50	90	1.9338	1.8935	1.8970	1.8204	1.7947	1.7131	1.5448	1.5354	1.5398	1.5402	1.4977	1.4412	1.4593	1.4599	1.4571
8	8	1.00	75	1.7718	1.7433	1.7237	1.7045	1.7352	1.6762	1.3676	1.3668	1.3669	1.3680	1.3444	1.3318	1.3371	1.3392	1.3453
8	8	1.00	90	2.2790	2.1989	2.1936	2.1026	2.1495	2.0088	1.5728	1.5682	1.5785	1.5737	1.5106	1.4800	1.4944	1.4999	1.5133
8	16	0.25	75	1.3528	1.3319	1.3163	1.3305	1.3070	1.3087	1.2869	1.2885	1.2778	1.2771	1.2529	1.2323	1.2303	1.2269	1.2269
8	16	0.25	90	1.9111	1.7854	1.7721	1.7788	1.7301	1.6737	1.4444	1.4324	1.6138	1.6077	1.4540	1.4501	1.4447	1.4447	1.4208
8	16	0.50	75	1.5365	1.4882	1.4812	1.4818	1.4939	1.4607	1.3892	1.3842	1.3821	1.3820	1.3340	1.3030	1.3123	1.3111	1.3113
8	16	0.50	90	2.0940	1.8974	1.9012	1.8842	1.9659	1.8011	1.6988	1.6907	1.6891	1.6850	1.5522	1.4570	1.4834	1.4847	1.4796
8	16	1.00	75	1.9120	1.7987	1.7900	1.7810	1.8668	1.7548	1.5382	1.5350	1.5371	1.5364	1.4469	1.4239	1.4341	1.4358	1.4463
8	16	1.00	90	2.5896	2.2499	2.2647	2.2136	2.4722	2.1450	1.8536	1.8369	1.8520	1.8442	1.6262	1.5615	1.5909	1.5905	1.6170
Overall				2.0636	1.9716	1.9599	1.9479	1.9595	1.8448	1.7057	1.6943	1.6872	1.6820	1.6008	1.5588	1.5557	1.5551	1.5545

Table 5. ANOVA results for mean job flow time.

Source	<i>F</i>	<i>p</i> -value	Source	<i>F</i>	<i>p</i> -value
H	139547.379	<.001	H * M * SR	315.262	<.001
M	3629735.494	<.001	H * M * WL	1257.813	<.001
F	240200.521	<.001	H * F * SR	213.049	<.001
SR	422569.505	<.001	H * F * WL	1284.525	<.001
WL	2976477.688	<.001	H * SR * WL	238.883	<.001
H * M	4925.563	<.001	M * F * SR	1714.781	<.001
H * F	4350.776	<.001	M * F * WL	824.215	<.001
H * SR	9284.924	<.001	M * SR * WL	4308.935	<.001
H * WL	21270.117	<.001	F * SR * WL	1297.334	<.001
M * F	8370.355	<.001	H * M * F * SR	46.879	<.001
M * SR	18570.794	<.001	H * M * F * WL	283.772	<.001
M * WL	256385.852	<.001	H * M * SR * WL	73.774	<.001
F * SR	37301.641	<.001	H * F * SR * WL	44.849	<.001
F * WL	8966.063	<.001	M * F * SR * WL	66.957	<.001
SR * WL	17055.272	<.001	H * M * F * SR * WL	10.435	<.001
H * M * F	974.490	<.001			

Note that MASP_AD_PAR and MASP_HY_PAR set a lower bound for batch size ($k_j^{\min}$). It relates set-up time ($s_{j_0,j}$) to the mean job arrival interval ($1/\lambda_T$) and mean processing time per job at the machine over all families ($\bar{p}$). In case of a tie, i.e. all families do not meet their lower bound, $k_j^{\min} = 0$. For more details, see Van der Zee (2010) and Van der Zee, Gaalman, and Nomden (2011).

4.2.3 Job sequencing: which job is to be selected from the chosen family

In line with findings from previous research (Mahmoodi, Dooley, and Starr 1990; Wemmerlov 1992; Pickardt and Branke 2012) all heuristics adopt the SPT rule for job sequencing. In principle, jobs are selected from those jobs that are available at the switching moment ($i \in I_j(t_0)$). However, in case a machine is set-up for a family that already has access to one or multiple machines, job selection is restricted to jobs being available after machine set-up ($i \in \tilde{I}_j(t_0 + s_{j_0,j})$), compare (b).

5. Design of the simulation study

5.1 Research issues

By our simulation study, we aimed to gain insights in the potential of new and existing heuristics for family-based dispatching with parallel machines. Heuristics' potential is assessed in two series of experiments. Firstly, flow time performance of existing heuristics and proposed heuristics is compared for various shop configurations. Secondly, we seek to underpin our choices made with respect to the construction of proposed heuristics. More specifically, we study relevance of specific rules for (1) adjusting priorities for those families having access to multiple machines at the same time, and (2) machine coordination, for improving shop performance, also see Section 4.2.

5.2 Experimental design – shop configurations

The experimental design for the simulation study is shown in Table 3. Choices with respect to fixed factors, experimental factors and their ranges are in line with literature on family-based dispatching (Wemmerlov and Vakharia 1991; Wemmerlov 1992; Frazier 1996; Jensen, Malhotra, and Philipoom 1996; Shambu, Suresh, and Pegels 1996; Jensen, Malhotra, and Philipoom 1998; Kannan 1998; Marsh, Shafer, and Meredith 1999; Chern and Liu 2003; Nomden and van der Zee 2008; Pickardt and Branke 2012).

All configurations studied concern a parallel machine shop. Job families are assumed to have an equal share in product mix. Job inter-arrival times are modelled by a negative exponential distribution. Job processing and machine set-up times are drawn from a negative exponential distribution, with mean of 1.

The key experimental factor is the family-based dispatching heuristic. All heuristics are classified for the way they implement rules for machine coordination, and the way priorities for those families having access to multiple machines

may be adjusted relative to those families not having such access. Furthermore, three categories of heuristics are distinguished, i.e. existing heuristics (marked in *italics*), proposed heuristics (marked in **bold**) and a set of ‘supplementary’ heuristics (no markings). The latter category of heuristics is meant to facilitate a second series of experiments, see Section 5.1. The inclusion of these heuristics allows comparing the added value of implementing alternative rules for priority setting and machine coordination. Note that heuristic names reflect the choice of rules made. For example, ‘PARJ’ refers to the logic underlying the family priority rule for M-SPT (Jensen, Malhotra, and Philipoom 1998), see Table 1. According to this logic, family priorities are related to the backend of the queue concerning those jobs with longest processing times (see Section 4.1). Alternatively, ‘PAR’ identifies the rule proposed in this article for adjusting family priorities for those jobs starting processing at alternative machines during set-up, see Section 4.2. Likewise, ‘AU’ clarifies that two rules have been implemented for machine coordination, i.e. ‘A’ that allows families to gain access to at most one machine, unless there are no competing families in queue, and ‘U’ that suggests to release ‘excess’ machines as soon as a family becomes available that has no access to a machine, compare Sections 2 and 4.

All heuristics are tested for their performance for various shop configurations. Shop configurations differ with respect to the set-up to run-time ratio, i.e. mean set-up time divided by mean processing time, shop work load, number of machines and number of job families. Choice of experimental factors is motivated by their estimated impact on shop and rule performance, as it has been indicated by previous research.

Three levels for the set-up to run-time ratio four settings are considered: 0.25, 0.50 and 1.00. Many researchers indicate how flow time performance of heuristics is very sensitive to the level of the set-up to run-time ratio, see, for example, Mahmoodi, Dooley, and Starr (1990), Mahmoodi and Dooley (1991), Ruben, Mosier, and Mahmoodi (1993), and Shambu, Suresh, and Pegels (1996).

Alternative work load levels of 75 and 90%, including both processing and set-ups, are chosen by adapting the mean inter-arrival time. The levels have been determined using FCFAM as a benchmark. Arrival rates have been established for each setting of the number of job families and the set-up to run-time ratio. Wemmerlov and Vakharia (1991) indicate that shop work load has a major impact on (relative) performance of family-based dispatching heuristics.

Figure 1. Estimated marginal means of mean job flow times for existing and proposed heuristics.

Two levels for the number of machines are considered: 4 and 8. Respective numbers are chosen in conformity with previous research on parallel machine shops (Jensen, Malhotra, and Philipoom 1998; Kannan 1998; Chern and Liu 2003; Nomden and van der Zee 2008). The number of families ranges from 4 to 16. This range is in conformity with many other studies, see, for example, Jensen, Malhotra, and Philipoom (1996), Marsh, Shafer, and Meredith (1999), and Wemmerlov (1992).

5.3 Simulation modelling

Plant SimulationTM 10.1 (Siemens PLM Software 2011) is used to carry out the simulation experiments. A total of 30 runs are considered for each experiment. The length of the warm-up period is determined using the Welch procedure (Law and Kelton 2000). In accordance with the outcomes of the procedure, the warm up period and run length are set at 10,000 and 60,000 time units, respectively. SPSS 22.0 for Windows (IBM 2014) was used for performing analysis of variance (ANOVA) for analysing effects of alternative shop configurations on heuristics' performance.

6. Analysis of simulation results

In this section, we will analyse the outcomes of the simulation study, starting from the research issues identified in Section 5.1. Outcomes of the initial series of experiments are mentioned in Table 4. The table shows mean job flow times for each heuristic across all shop configurations. ANOVA has been used for analysing effects of alternative shop configurations on heuristics' performance, see Table 5 and Figures 1 and 2. All effects are significant ($p < 0.001$).

Figure 2. Estimated marginal means of mean job flow time for alternative settings of number of machines, work load, set-up to run-time ratio and number of families.

Figure 3. Estimated marginal means of mean shop utilisation for existing and proposed heuristics for alternative work load settings.

Tukey's HSD (honest significant difference) test has been applied to establish best performing heuristics for each shop configuration. Outcomes for best performing heuristics are printed in bold, see Table 4.

Figure 1 clarifies how the choice of heuristic influences study outcomes in terms of the marginal means of mean job flow times for all shop configurations studied. It shows how proposed heuristics clearly outperform existing heuristics. Relative gains, as indicated by overall results for best performing heuristics for each category of heuristics, i.e. FCFAM-C and MASP_SNSPT_PAR, are about 18%, also compare Table 4 (final row). Largest differences among respective heuristics for specific settings are found for a high set-up to run-time ratio (18–39%), see Table 4. Best performing heuristics are MAS_PAR, SNSPT_PAR, MASP_AD_PAR and MASP_HY_PAR. Differences among the latter heuristics are small for most shop configurations (0–2%). Hence, for many shop configurations no single best heuristic can be established. For low set-up to run-time ratios SNSPT_PAR performs best. For moderate set-up to run-time ratios MAS_PAR, SNSPT_PAR, MASP_AD_PAR or MASP_HY_PAR may be favoured. MAS_PAR indicates best performance for high-set-up to run-time ratios.

Figure 2 supplies further information on the way the choice of shop configuration influences heuristics' relative performance in terms of two-way interaction effects. An increase in the number of machines goes together with a reduction of flow times reported for the heuristics (Figure 2(a)). This is as expected, as a larger pool of available machines tends to increase synergetic effects on flow times (Hopp and Spearman 2008). Apart from causing a general downward shift of mean job flow times, an increase of the number of machines also impacts on relative differences among heuristics – which tend to become somewhat smaller.

Work load settings have a clear effect on heuristics' performance (Figure 2(b)). As expected, an increase of work load causes higher mean job flow times for all heuristics. Moreover, relative differences among heuristics increase for higher arrival rates. The latter effect may be explained by the way heuristics create set-up efficiencies. Clearly, under high work loads set-up efficiencies pay off, as they tend to have a direct impact on the (many) items in queue. Whereas existing heuristics allow for the possibility to dispatch jobs group wise, proposed heuristics offer further possibilities for increasing set-up efficiencies, by adjusting queue contents for those items being withdrawn from the queue during

Figure 4. Estimated marginal means of mean job flow times for heuristics' extensions concerning family priority setting and machine coordination.

set-up (all proposed heuristics), and/or including set-up times in priority setting (MASP_PAR, MAS_PAR, MASP_AD_PAR, MASP_HY_PAR, SNSPT_PAR). Note that these differences with respect to heuristics' construction and performance are clearly reflected in Figure 2(b) (WL = 75 vs. WL = 90) and underpinned in Figure 3. The latter figure clarifies how shop utilisation benefits from set-up efficiencies realised by proposed heuristics.

Effects of longer set-up times on heuristics performance are quite similar to those found for higher work loads (Figure 2(c)). In both cases, heuristics' capabilities in creating set-up efficiencies pay off. Effects even tend to be somewhat stronger, as clarified by the performance for MAS_PAR, that stresses set-up efficiencies by a single focus on weighted set-up times in family priority setting, compare Table 2. For moderate and high set-up to run-time ratios it is

(among) the best performing heuristic(s), compare Figure 2(c) and Table 4. However, for a low set-up to run-time ratio SNSPT_PAR may be preferred. Note that other experimental factors seem to have little effect on relative performance differences among best performing heuristics, i.e. MAS_PAR, MASP_AD_PAR, MASP_HY_PAR, SNSPT_PAR. However, the set-up to run-time ratio allows to differentiate between respective heuristics.

An increase of the number of families results in higher mean job flow times (Figure 2(d)). This is as expected. After all, under these circumstances, there are fewer opportunities for grouping jobs family wise – as queue length per family tends to go down. In turn, the small batch sizes also explain the strong performance for those heuristics that consider set-up times in priority setting, i.e. MASP_PAR, MAS_PAR, MASP_AD_PAR, MASP_HY_PAR and SNSPT_PAR. After all, if batch sizes are becoming smaller, it is even more important to reduce the time spent on set-ups.

In order to underpin our choices made with respect to heuristics construction, we consider two further issues. Firstly, we traded-off two alternative rules for adjusting family priorities for those job families having access to multiple machines at the same time. Figure 4(a) shows how existing heuristics developed for addressing the single machine case (HE = None) may benefit from (i) an extension by the priority rule proposed by Jensen, Malhotra, and Philipoom (1998) (HE = PARJ) and (ii) the implementation of the family priority rule proposed in this paper (HE = PAR). See Section 4.1 for details on respective extensions. Figure 4(a) clearly shows the added value of the latter extension in terms of the marginal means of mean job flow times over all shop configurations studied. Performance improvements found amount to 5–14% on average.

A second issue in constructing proposed heuristics concerned the choice of rules for machine coordination, see Figure 4(b). Three alternative rules for machine coordination have been considered, assuming the extension 'PAR' (also see above) to be implemented: (i) no restriction on machine use (HE = PAR), (ii) each family gains access to at most one machine, unless there are no competing families in queue (HE = PAR_A) and (iii) each family gains access to at most one machine, unless there are no competing families in queue; machine use is reconsidered after each job being completed (HE = PAR_AU). Results in Figure 4(b) show that only FCFAM_PAR, MJ_PAR, MAP_PAR and FCSPT_PAR may benefit from implementing a rule, i.e. (iv), for machine coordination. A likely explanation for this finding is in the choice of rules for family priority setting. Respective heuristics do not include family set-up times in priority setting. Hence, they lack a mechanism in family priority setting for preventing possible adverse effects of machine pooling on set-up frequencies. Therefore, additional rules for machine coordination are required to safeguard set-up efficiencies. Furthermore, results in Figure 4(b) clarify that there is no benefit in promptly releasing machines for use by a new family, once they have been assigned to a family (compare rule iii). In other words, set-ups, once executed, should be considered sunk costs.

7. Concluding remarks

In this article, we studied family-based dispatching rules for use in shops with parallel machines. A total of nine existing heuristics for family-based dispatching has been included in our study. Two extensions have been evaluated for all heuristics included in the study. Firstly, the choice of rule for family priority setting is considered. Simulation results indicate that a choice of rule that accounts for those jobs within the current family starting processing during set-up at other machines, should be preferred over other rules suggesting no or alternative adjustments. A second extension concerns the choice of rule for coordinating the number of machines in use by a single family. Experiments indicate that a rule that aims to restrict machine use to just one machine is only beneficial for shop performance, in case the rule adopted for family priority setting does not account for set-up times.

Best performing heuristics are SNSPT_PAR, MAS_PAR, MASP_AD_PAR and MASP_HY_PAR. For these heuristics, only the new rule for family priority settings, as suggested in this article, has been implemented. Simulation results indicate that respective heuristics outperform existing heuristics by up to 39% for specific settings. Highest performance gains for heuristics studied are found for high set-up to run-time ratios. Performance differences among best performing heuristics are typically small, 0–2%. For low set-up to run-time ratios SNSPT_PAR performs best. For moderate set-up to run-time ratios SNSPT_PAR, MAS_PAR, MASP_AD_PAR or MASP_HY_PAR may be favoured. MAS_PAR indicates best performance for high set-up to run-time ratios.

Many avenues for further research on family-based dispatching may be considered, concerning, for example, choice of shop configuration, use of information on upcoming job arrivals, selected criterions and methods used in rule construction. Alternative shop configurations may be studied, such as manufacturing networks (Robinson, Fowler, and Bard 1995; Van der Zee 2002). Furthermore, in this article we show how family priority setting may be improved by accounting for those jobs being processed at alternative machines during set-up. Knowing about job arrivals during set-up may further improve priority setting and – next – shop performance (Nomden, van der Zee, and Slomp 2008). Many other choices of criterions may be considered in extending heuristics. For example, heuristics adopting due date related

criteria may help in meeting the practical needs for on-time delivery. Also (multi-objective) cost related criteria may be investigated, thereby acknowledging the fact that each set-up typically causes a financial loss due to the use or consumption of resources and materials (Detti, Meloni, and Pranzo 2007; Ciavotta et al. 2008). Finally, in this article we study heuristics relying on (a set) of dispatching rules. One may wonder about the benefits of employing algorithms focussing at further improvement of the shop schedule, such as, for example, local search procedures (Pinedo 2005; Ciavotta, Meloni, and Pranzo 2009).

References

- Allahverdi, A., J. N. D. Gupta, and T. Alsdowaisan. 1999. "A Review of Scheduling Research Involving set-up Considerations." *Omega* 27 (2): 219–239.
- Allahverdi, A., C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov. 2008. "A Survey of Scheduling Problems with Setup Times or Costs." *European Journal of Operational Research* 187: 985–1032.
- Blackstone, J. H., D. T. Phillips, and G. L. Hogg. 1982. "A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations." *International Journal of Production Research* 20 (1): 27–45.
- Chern, C. C., and Y. L. Liu. 2003. "Family-based Scheduling Rules of a Sequence-dependent Wafer Fabrication System." *IEEE Transactions on Semiconductor Manufacturing* 16 (1): 15–25.
- Ciavotta, M., P. Detti, C. Meloni, and M. Pranzo. 2008. "A Bi-objective Coordination Setup Problem in a Two-stage Production System." *European Journal of Operational Research* 189: 734–745.
- Ciavotta, M., C. Meloni, and M. Pranzo. 2009. "Scheduling Dispensing and Counting in Secondary Pharmaceutical Manufacturing." *AIChE Journal* 55 (5): 1161–1170.
- Ciavotta, M., C. Meloni, and M. Pranzo. 2013. "Minimising General Setup Costs in a Two-stage Production System." *International Journal of Production Research* 51 (8): 2268–2280.
- Detti, P., C. Meloni, and M. Pranzo. 2007. "Minimizing and Balancing Setups in a Serial Production System." *International Journal of Production Research* 45 (24): 5769–5788.
- Flynn, B. B. 1987. "Repetitive Lots: The Use of a Sequence-dependent Set-up Time Scheduling Procedure in Group Technology and Traditional Shops." *Journal of Operations Management* 7 (1–2): 203–216.
- Frazier, G. V. 1996. "An Evaluation of Group Scheduling Heuristics in a Flow-line Manufacturing Cell." *International Journal of Production Research* 34 (4): 959–976.
- Gavett, J. W. 1965. "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility." *Management Science* 11 (8): B-166–B-176.
- Groover, M. P. 2008. *Automation, Production Systems, and Computer-Integrated Manufacturing*. 3rd ed. New York: Prentice Hall.
- Hopp, W. J., and M. L. Spearman. 2008. *Factory Physics*. 3rd ed. Boston, MA: McGrawHill.
- IBM. 2008. *SPSS 22.0 for Windows Software Package*. Chicago, IL.
- Jayamohan, M. S., and C. Rajendran. 2000. "New Dispatching Rules for Shop Scheduling: A Step Forward." *International Journal of Production Research* 38 (3): 563–586.
- Jensen, J. B., M. K. Malhotra, and P. R. Philipoom. 1996. "Machine Dedication and Process Flexibility in a Group Technology Environment." *Journal of Operations Management* 14 (1): 19–39.
- Jensen, J. B., M. K. Malhotra, and P. R. Philipoom. 1998. "Family-based Scheduling of Shops with Functional Layouts." *International Journal of Production Research* 36 (10): 2687–2700.
- Kannan, V. R. 1998. "Analysing the Trade-off between Efficiency and Flexibility in Cellular Manufacturing Systems." *Production Planning & Control* 9 (6): 572–579.
- Kochhar, S., and R. J. T. Morris. 1987. "Heuristic Methods for Flexible Flow Line Scheduling." *Journal of Manufacturing Systems* 6 (4): 299–314.
- Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. Boston, MA: McGraw-Hill.
- Mahmoodi, F., and K. J. Dooley. 1991. "A Comparison of Exhaustive and Non-exhaustive Group Scheduling Heuristics in a Manufacturing Cell." *International Journal of Production Research* 29 (9): 1923–1939.
- Mahmoodi, F., K. J. Dooley, and P. J. Starr. 1990. "An Investigation of Dynamic Group Scheduling Heuristics in a Job Shop Manufacturing Cell." *International Journal of Production Research* 28 (9): 1695–1711.
- Marsh, R. F., S. M. Shafer, and J. R. Meredith. 1999. "A Comparison of Cellular Manufacturing Research Presumptions with Practice." *International Journal of Production Research* 37 (14): 3119–3138.
- McKay, K. M., F. R. Safayeni, and J. A. Buzacott. 1988. "Job-shop Scheduling Theory: What is Relevant?" *Interfaces* 18 (4): 84–90.
- Mizrak, P., and G. Bayhan. 2006. "Comparative Study of Dispatching Rules in a Real-life Job Shop Environment." *Applied Artificial Intelligence* 20: 585–607.
- Mosier, C. T., D. A. Elvers, and D. Kelly. 1984. "Analysis of Group Technology Scheduling Heuristics." *International Journal of Production Research* 22 (5): 857–875.
- Nomden, G., and D. J. van der Zee. 2008. "Virtual Cellular Manufacturing: Configuring Routing Flexibility." *International Journal of Production Economics* 112 (1): 439–451.

- Nomden, G., D. J. van Der Zee, and J. Slomp. 2008. "Family-based Dispatching: Anticipating Future Jobs." *International Journal of Production Research* 46 (1): 73–97.
- Pickardt, C. W., and J. Branke. 2012. "Setup-oriented Dispatching Rules – A Survey." *International Journal of Production Research* 50 (20): 5823–5843.
- Pinedo, M. 1995. *Scheduling: Theory, Algorithms, and Systems*. Englewood Cliffs, NJ: Prentice-Hall.
- Pinedo, M. 2005. *Planning and Scheduling in Manufacturing and Services*. New York: Springer.
- Ponnambalam, S. G., P. Aravindan, and K. R. R. Reddy. 1999. "Analysis of Group-scheduling Heuristics in a Manufacturing Cell." *The International Journal of Advanced Manufacturing Technology* 15 (12): 914–932.
- Robinson, J. K., J. W. Fowler, and J. F. Bard. 1995. "The Use of Upstream and Downstream Information in Scheduling Semiconductor Batch Operations." *International Journal of Production Research* 33 (7): 1343–1369.
- Ruben, R. A., C. T. Mosier, and F. Mahmoodi. 1993. "A Comprehensive Analysis of Group Scheduling Heuristics in a Job Shop Cell." *International Journal of Production Research* 31 (6): 1343–1369.
- Russell, G. R., and P. R. Philipoom. 1991. "Sequencing Rules and Due Date Setting Procedures in Flow Line Cells with Family Setups." *Journal of Operations Management* 10 (4): 524–545.
- Sarin, S. C., A. Varadarajan, and L. Wang. 2011. "A Survey of Dispatching Rules for Operational Control in Wafer Fabrication." *Production Planning & Control* 22 (1): 4–24.
- Shambu, G., N. C. Suresh, and C. C. Pegels. 1996. "Performance Evaluation of Cellular Manufacturing Systems: A Taxonomy and Review of Research." *International Journal of Operations and Production Management* 16 (8): 81–103.
- Siemens PLM Software. 2011. *Plant SimulationTM 10.0*. Stuttgart.
- Stuber, F. 1998. "Approaches to Shop Floor Scheduling – A Critical Review." In *Shop Floor Control – A Systems Perspective*, edited by E. Scherer, 201–220. Berlin: Springer.
- Van der Zee, D. J. 2002. "Adaptive Scheduling of Batch Servers in Flow Shops." *International Journal of Production Research* 40 (12): 2811–2833.
- Van der Zee, D. J. 2010. "Non-exhaustive Family Based Dispatching Heuristics–Exploiting Variances of Processing and Set-up Times." *International Journal of Production Research* 48 (13): 3783–3802.
- Van der Zee, D. J., G. J. C. Gaalman, and G. Nomden. 2011. "Family Based Dispatching in Manufacturing Networks." *International Journal of Production Research* 49 (23): 7059–7084.
- Wemmerlov, U. 1992. "Fundamental Insights into Part Family Scheduling: The Single Machine Case." *Decision Sciences* 23 (3): 565–595.
- Wemmerlov, U., and A. J. Vakharia. 1991. "Job and Family Scheduling of a Flow-Line Manufacturing Cell: A Simulation Study." *IIE Transactions* 23 (4): 383–393.

Appendix 1

Table A1. Overview of family-based dispatching heuristics.

Name	Explanation ^a	References	Included in study ^b	Extensions considered ^c
APT-L	Average Processing Time with limited set-up duplication	Jensen, Malhotra, and Philipoom (1996)	Yes	–
ECON	ECONomic trade-off between changing queues	Mosier, Elvers, and Kelly (1984), Ruben, Mosier, and Mahmoodi (1993)	No	–
FCFAM	First Come FAMily	Flynn (1987)	Yes	FCFAM_PARJ, FCFAM_PAR, FCFAM_PAR_A, FCFAM_PAR_AU
FCFAM-C	–	Nomden and van der Zee (2008)	Yes	–
FCSPT ^d	–	Chern and Liu (2003)	Yes	FCSPT_PARJ, FCSPT_PAR, FCSPT_PAR_A, FCSPT_PAR_AU
FCFS-F	Family-Based FCFS	Chern and Liu (2003)	Yes	–
FCFS-L	FCFS with limited set-up duplication	Jensen, Malhotra, and Philipoom (1996)	Yes	–
MAS	Minimum Average Set-up time	Nomden, van der Zee, and Slomp (2008)	Yes	MAS_PARJ, MAS_PAR, MAS_PAR_A, MAS_PAR_AU
MAP	Minimum Average Processing time	Mosier, Elvers, and Kelly (1984)	Yes	MAP_PARJ, MAP_PAR, MAP_PAR_A, MAP_PAR_AU
MASP	Minimum Average Set-up plus Processing time	Jensen, Malhotra, and Philipoom (1996)	Yes	MASP_PARJ, MASP_PAR, MASP_PAR_A, MASP_PAR_AU
MASP_AD	ADaptive Minimum Average Set-up plus Processing time	Van der Zee (2010)	Yes	MASP_AD_PARJ, MASP_AD_PAR, MASP_AD_PAR_A, MASP_AD_PAR_AU
MASP_HY	HYbrid Minimum Average Set-up plus Processing time	Van der Zee (2010)	Yes	MASP_HY_PARJ, MASP_HY_PAR, MASP_HY_PAR_A, MASP_HY_PAR_AU
MJ	Most Jobs	Frazier (1996)	Yes	MJ_PARJ, MJ_PAR, MJ_PAR_A, MJ_PAR_AU
RepetitiveLots-FCFS	–	Kannan (1998)	Yes	–
SNSPT	Shortest Normalised Setup and Processing Time	Kochhar and Morris (1987)	Yes	SNSPT_PARJ, SNSPT_PAR, SNSPT_PAR_A, SNSPT_PAR_AU
SRPT-F	Family-Based Shortest Remaining Process Time	Chern and Liu (2003)	Yes	–
SST	Short Set-up Time	Gavett (1965)	No	–

^aEach heuristic is referred to by its acronym. If an explanation of the acronym is available in literature, it is mentioned here.

^bIndicates whether an heuristic is included in the simulation study or not.

^cBoth existing heuristics for the parallel machine case and heuristics that showed good performance for the single machine case have been included in the study. For all heuristics, two possible extensions have been considered: Adjusting family priorities: rules PARJ and PAR. Machine coordination: rules A and AU. All rules are explained in Section 4. Implementation of the respective rules is reflected in the naming convention for the extended heuristics. For example, FCFAM_PAR_AU clarifies how FCFAM has been extended by adjusting family priorities according to PAR, and implementing AU to coordinate machine use.

^dSRPT-F and FCSPT are identical.