Virtual cellular manufacturing

Nomden, Gert

CHAPTER 4

Single-machine case

Group Technology exploits similarities in product and process design to meet the diversity of customer demand in an economic way. In this paper we consider one of the implementations of this concept – family-based dispatching. Intrinsic to family-based dispatching is the grouping of similar types of products for joint processing. In this way the number of set-ups may be reduced. Consequently, lead-time performance of the shop can be improved. We extend existing rules for family-based dispatching by including data on upcoming job arrivals. Typically, this type of data resides in the minds of the operators, or is stored in a shop-floor control system. Its availability allows for (1) better estimates of the composition of a process batch for a family, (2) the consideration of families for which no products are available at the decision moment, and (3) the possibility to start set-ups in anticipation of future job arrivals. The potential of including forecast data in decision-making is demonstrated by an extensive simulation study of a single-machine shop. Results indicate the possibility of significant improvements of flow time performance.

4.1 Introduction

Mass customisation and increasing competition force manufacturers in various industries towards higher levels of responsiveness. Group Technology provides one of the answers to meet these demands. The key idea underlying this concept is the exploitation of similarities in product and process design. In this article we address this issue by studying family-based dispatching rules for shop floor control. Essentially, these rules strive to improve lead time performance by reducing set-up time. This is realised by grouping products that share similar requirements with respect to system set-up, i.e., families, for joint dispatching. The application of these rules is considered particularly beneficial in small batch discrete parts manufacturing (Hyer and Wemmerlöv, 2002; Wisner and Sifer, 1995). In these environments frequencies of set-ups tend to be high, while set-up times may be lengthy.

This chapter has been published as:

NOMDEN, G., ZEE, D.J. VAN DER, SLOMP, J. (2008), Family-based dispatching: anticipating future jobs. *International Journal of Production Research*, Volume 46, Issue 1, Pages 73–97.

Family-based dispatching rules extend conventional dispatching rules by applying a two-phase approach. According to this approach the choice of a job to be processed next is preceded by the choice of the product family (Mosier et al., 1984). Product families are prioritised using shop-floor data as an input. So far, research efforts concentrated on using local data for priority setting, i.e. information on jobs readily available at the shop floor, such as queue lengths per family, set-up times and processing times (Ruben et al., 1993). However, in many cases, alternative sources of relevant data may be present. In this article we consider one such source: forecast data on jobs to arrive at the shop in the nearby future. Our research is motivated by a case study concerning the manufacturing of centrifugal pumps. In deciding on what to produce next, operators apply family based dispatching rules. Decisions are made using local information on queue contents, next to data on parts yet to arrive at the station, see Figure 4.1. The latter data may concern operators' own observations of work in process for the shop, as well as monitoring data stored in the shop floor control system. The availability of this information allowed operators (1) to exploit the current setup of a system to a greater extent by waiting for a job that is about to arrive, and (2) start setup activities prior to job arrivals. Other studies confirm the relevance of using forecast data for shop floor control (Fleig and Schneider 1998, Fowler et al. 2000). In general, opportunities for exploiting this type of data are expected to increase. This follows from the development and application of on-line monitoring systems of manufacturing processes (for example Ovacik and Uzsoy, 1994; Solomon et al., 2002).

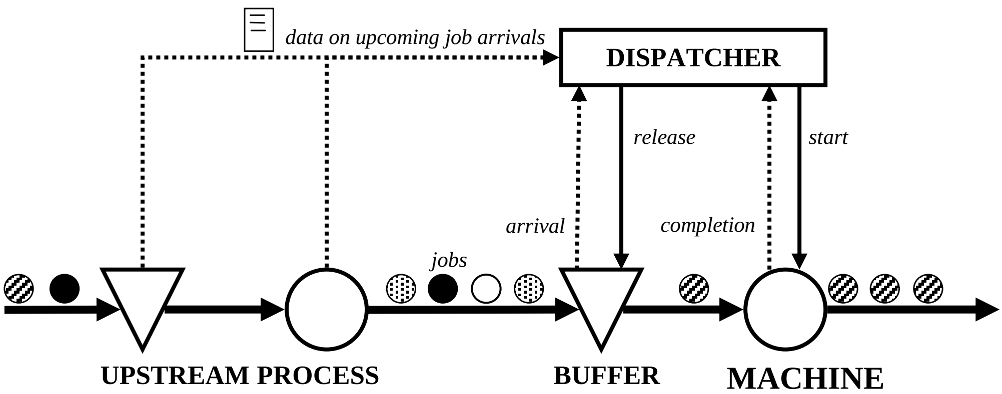

Figure 4.1: Family-based dispatching using data on upcoming job arrivals

We foresee a number of benefits concerning the use of forecast data for family-based dispatching. Firstly, family priority setting becomes more accurate, as insights on actual processing batches for each family are improved. Note that a processing batch may involve both jobs in queue, and jobs that arrive during processing.

Secondly, knowledge of future arrivals allows the dispatcher to act in anticipation. This refers to the possibilities (1) to initiate set-up activities prior to job arrivals, or (2) leave the system idle until new jobs arrive that require the current system set-up. Clearly, lead time performance may benefit from such flexibility.

In order to realise the foreseen benefits of forecast data for dispatching decisions we propose a number of extensions to existing rules. We do so, starting from a generic framework that captures the general structure of the control system for family-based dispatching. To study the benefits of the extended rules we designed a simulation study. In this study we compare and analyse performance of the extended rules with existing, non-extended, rules for a variety of settings with respect to system load, set-up to run-time ratios and number of part/product families. These settings are chosen such that they reflect our own observations of industrial settings and case studies reported in the literature. Specific attention is paid to the sensitivity of rule performance to the information horizon, i.e. the moment up to which data are available on future arrivals, and forecast errors.

This paper is structured as follows. In Section 2 we survey the literature and outline our research contributions. Next, in Section 3, we generalise existing literature in terms of a decision framework for family-based dispatching. The framework serves as a basis for describing extensions of current heuristics in order to deal with forecast data. To demonstrate the potential of using forecast data for family-based dispatching we conduct an extensive simulation study. Its design and outcomes will be discussed in Sections 4 and 5. Finally, conclusions are summarised in Section 6.

4.2 Literature review

Scheduling research that incorporates the notion of set-up times received significant attention. It is recognised that processing items with similar manufacturing requirements consecutively can improve flow performance significantly. So far, the main focus has been on the so-called deterministic rules, which start from the idea that all jobs are known beforehand. For an overview see Allahverdi et al. (1999). Unfortunately, practice often asks for a dynamic approach that is capable of dealing with limited information, and allows for regular updates of the job list. Moreover, rules should allow for timely answers. Analytic approaches, like those implemented for deterministic rules, often do not work out under these circumstances, due to problem size, and complexity. Therefore one often relies on heuristic approaches (McKay et al., 1988; Stuber, F., 1998). In this section we will survey literature for heuristic rules for family-based dispatching. Family-based dispatching starts from the idea that products can be grouped in families based on their similarities with respect to set-up requirements. It may be considered as an element of the Virtual Cellular Manufacturing concept (VCM). VCM starts from the idea that grouping efficiencies can be realised using family-oriented scheduling/dispatching rules (Kannan and Ghosh, 1996a).

Rules for family-based dispatching concern an extension of conventional dispatching rules. Dispatching decisions are made assuming a two phase approach, in which the choice of a family precedes the choice of a specific job – from the set of jobs available for the family (Mosier et al., 1984). Essentially, alternative rules for family-based dispatching only differ with respect to family priority setting, as the choice of individual jobs is made assuming the use of conventional dispatching rules, see Blackstone et al. (1982) or Chang et al. (1996) for an overview. Let us now consider existing heuristic rules, using Table 4.1 as a starting point for discussion.

Table 4.1 supplies an overview of rules for family-based dispatching. The notation that has been adopted for defining family priority settings can be found in Appendix A. Its columns specify rule name, a definition of family priority setting, and related references. Family priority setting may relate to a number of data sources:

- Local data, i.e. queue lengths per family, family set-up times, and/or job processing times;
- Forecast data, i.e. information on jobs yet to arrive at the shop;
- Historical data, i.e. summaries of data on completed jobs in terms of distributions, averages and spreads.

Let us start our discussion with those rules that adopt local data as an input to decision-making. Next we will consider rules that use alternative sources of data, i.e., forecast data and historical data in somewhat more detail. The first rule mentioned in Table 4.1 is the FCFAM rule. This rule shows similarities with the well-known first come first serve rule (FCFS). It prioritises families by considering the earliest entry moment of the jobs available in queue for a family. The rules MJ, MS, and WORK all relate decisions to one specific category of local data – queue lengths, set-up times and processing times. Rules that use multiple categories of local data are SWORK (set-up and processing time), MAS and ECON (set-up time and queue length), MAP/AVE(SPT) (processing time and queue length), and MASP/APT (set-up and processing time, queue length).

Basically, most definitions for family priority setting mentioned in Table 4.1 may be related to the concept of the Weighted Shortest Processing Time rule (WSPT). This well-known dispatching rule sequences jobs by considering workload for a job relative to its weight, see for example Pinedo (1995). For family-based dispatching this concept is extrapolated towards composite jobs, i.e., the sets of jobs available for specific families. Alternative rules differ with respect to the way:

- Workload is estimated for a process batch (set-up times, processing times, queue lengths);
- Weights are set (queue lengths, uniform).

Typically, these choices may be related to data availability. In principle, possible choices with respect to workload and weight, allow for a limited set of priority settings. Most of them are covered in the literature, see Table 4.1. However, two alternative rules are left, for which we could not find any trace in the literature. We refer to them as MAS and SWORK. SWORK uses data on family set-ups and job processing times, while MAS relates family set-ups to the respective number of items in queue.

More recently, researchers have tried to improve the performance of family-based dispatching procedures by incorporating other sources of data. Kannan and Ghosh (1996a) propose two look-ahead dispatching schemes called VCM3 and VCM5. Both rules start from the idea that the operator is informed about the set of families which, next to having one or more jobs in the queue, also have jobs being processed in a preceding manufacturing stage. VCM3 assigns priority to this set of families, over those families that do not have jobs in a preceding stage. Within this set family selection is realized using the MJ rule. VCM5 considers whether the family that corresponds to the current machine set-up has one or more jobs in process in a preceding stage. If this is the case, the machine is forced to remain idle until the moment the respective job arrives. Otherwise, family selection is realized using the MJ rule. The authors included these rules in an attempt to increase permanence of virtual cells by creating more continuous links between manufacturing stages. The authors only report minor performance improvements, which may be attributed to the rather coarse manner in which future arrivals are counted with. We come back to this point in Section 5.

Mahmoodi and Martin (1997) strive to improve family priority setting by including future arrivals that may be predicted using historical data on demand rates. More in particular, their LPTMM heuristic foresees in the possibility to extend workload estimates by including jobs that are predicted to arrive during set-up and

#	Rule	Family choice	Source
1	FCFAM	$\arg\min_{j} t_{i,j}$	(Flynn, 1987)
2	MJ	$\arg\max_{j} q_j$	(Frazier, 1996)
3a	VCM3	$\arg\max_{j} \ \alpha + q_j$ with $\alpha = \begin{Bmatrix} \text{M if } l(H)_j > 0 \\ 0 \text{ otherwise} \end{Bmatrix}$	(Kannan and Ghosh, 1996a)
3b	VCM5	$\arg\max_{j} \ \alpha + q_j$ with $\alpha = \begin{Bmatrix} \text{M if } l(H)_{j0} > 0 \\ 0 \text{ otherwise} \end{Bmatrix}$	(Kannan and Ghosh, 1996a)
4	MS	$\arg\min_{j} s_{j_0,j}$	(Mahmoodi et al. 1990)
5	WORK	$\arg\max_{j} \sum_{i=1}^{q_j} p_{i,j}$	(Mosier et al. 1984)
6	SWORK	$\arg\max_{j} (s_{j_0,j} + \sum_{i=1}^{q_j} p_{i,j})$	This paper
7	LPTMM	$\arg\max_{j} \ (s_{j_0,j} + \frac{q_j}{\mu_j}) \cdot \frac{\mu_j}{\mu_j - \lambda_j}$	(Mahmoodi and Martin, 1997)
8	ECON	$\arg\max_{j} \begin{Bmatrix} \text{if } j = j_0 : q_j \cdot \bar{s}_j \\ \text{if } j \neq j_0 : [(q_j - 1) \cdot \bar{s}_j - q_{j_0} \cdot \bar{s}_{j_0} \end{Bmatrix}$	(Mosier et al. 1984)
9	MAS	$\arg\min_{j} \frac{s_{j_0,j}}{q_j}$	This paper
10	MAP	$\arg\min_{j} \frac{\sum_{i=1}^{q_j} p_{i,j}}{q_j}$	(Mosier et al. 1984)
11	MASP	$\arg\min_{j} \frac{(s_{j_0,j} + \sum_{i=1}^{q_j} p_{i,j})}{q_j}$	(Jensen et al. 1996)

Abbreviations:
1) FCFAM – First Come FAMily; 2) MJ – Most Jobs; 3 a, b) VCM3/VCM5 – Virtual Cellular Manufacturing 3/5; 4) MS – Minimum Set-up time; 5) WORK – largest WORKload; 6) SWORK – largest WORKload including Set-up time; 7) LPTMM – Longest Processing Time Mahmoodi Martin; 8) ECON – ECONomic trade-off; 9) MAS – Minimum Average Set-up time; 10) MAP – Minimum Average Processing time; 11) MASP – Minimum Average Set-up plus Processing time

Table 4.1: Overview of family-based dispatching rules

processing. LPTMM showed favourable performance in situations with an oscillating demand rate. Reddy and Narendran (2003) extend the work of Mahmoodi and Martin, by including the notion of economic lot sizes. Their rule prioritises product families for which the preset economic lot size is met earliest. Their heuristic performed well for configurations with a high shop load and high set-up to run-time ratio.

From our survey of literature we derive two observations:

- Forecast data are used for family priority setting to a limited extent. Only Kannan and Ghosh (1996a) adopt such data in rule construction;
- The set of rules for family-based dispatching that has been studied so far seems incomplete. We mention two alternative rules that employ specific categories of local data for family priority setting: MAS, and SWORK.

Finally, we found that available studies on family based dispatching offer limited overview and insight with respect to rule performance. Typically, alternative sets of rules are studied in a non-incremental way, not based on an underlying framework for control, and - more in particular – family priority setting. This finding, together with our wish to provide a sound and insightful basis for exploring the relevance of forecast data for family based dispatching, motivated us to adopt the single machine shop for doing research. In line with this approach we include the rules MAS and SWORK in our study for reasons of completeness.

4.3 Family-based dispatching – dealing with forecast data

In this section we discuss extensions of rules for family-based scheduling in order to deal with forecast data on upcoming job arrivals. We will do so starting from a single machine shop. After describing basic shop characteristics, a framework for decision making will be defined. The framework captures the general structure of the control system for family-based dispatching. It is meant to structure the discussion, to clarify the way rule extensions are made, and to serve as a first indication of the meaning of the extended rules for shop control and performance.

4.3.1 Shop description

The shop consists of a single machine and a buffer. Incoming products are stored in the buffer until the time they are released to the machine and processed. For reasons of simplicity and clarity of understanding, we associate each job with a single product. 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 product family requires a specific machine set-up. This so-called major set-up is associated with a set-up time $s_{j0,j}$. 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, $s_{j0,j}$ = 0 for $j = j_0$. Product related, so-called minor set-ups, are assumed to be included in job processing times ($p_{i,j}$), with i identifying individual jobs being available within a family.

4.3.2 A framework for decision making – use of forecast data

Let us now discuss a framework for family-based dispatching. The framework distinguishes between criterion, information base, triggers for decision making, decision options, and decision structure.

Criterion

As an objective we consider the minimisation of average flow time per job in the long run, so for a large number of jobs (N). Average flow time per job (MFT) is defined as:

$$MFT = \frac{\sum_{j \in J} \sum_{i=1,2,\ldots} ft_{i,j}}{N} \qquad (1)$$

$$\text{with } ft_{i,j} = w_{i,j} + p_{i,j}$$

In computing flow time for a job i belonging to family j ($fl_{i,j}$) we distinguish between waiting time ($w_{i,j}$) and job processing time ($p_{i,j}$).

Information base

At each decision moment the dispatcher may have several sources of shop floor data at his disposal. We consider:

- Local data, like queue lengths per family, family set-up times, job arrival times and job processing times;
- Forecast data, that is information on future job arrivals;
- Historical data, i.e., summaries of data on completed jobs in terms of distributions, averages and spreads.

Typically, forecast data will be limited to a certain information horizon (H). Also forecasts tend to be error prone. We will come back to both issues in Section 5.

Triggers

Two types of events govern shop dynamics: job arrivals and job completion. Both events generate new information for the dispatcher. As such they correspond to decision moments (t_0), at which a dispatcher may be triggered for making a decision on job release. Note that new or updated information on future arrivals is not considered as a decision moment.

Decision options

Essentially a few decision options are open for the planner:

- Switch the machine set-up to meet requirements of an other family, and release jobs available for this family following a conventional dispatching rule;
- Determine a next decision moment in anticipation of future arrivals, see Triggers;
- Do nothing – wait for a next arrival.

Note how the latter situation refers to a setting, where the planner does not act in anticipation. This may be the case if he only uses local data for decision-making, or set-ups require the presence of the job. In this article we do not assume such a restriction with respect to the family related, so-called major set-up.

Decision structure

Rules for family-based dispatching foresee in a two phase approach for decision making. In the first phase the need for changing system set-up is addressed. An important issue to consider here is whether or not the dispatcher should follow an exhaustive strategy, i.e. postpone decision-making on the choice of family until all jobs available for a family are processed. Prior research showed that in most cases an exhaustive strategy is most beneficial for flow time performance (Frazier, 1996; Mahmoodi and Dooley, 1991). Therefore we will only consider exhaustive strategies in the remainder of this article. In case a decision has to be made on the next family for which the system should be set-up, the use of a priority rule is assumed, compare Section 2.

In the second phase, a job is chosen for release among those jobs available for the family. For this phase conventional dispatching rules are employed, like for example the First Come First Serve rule (FCFS), and Shortest Processing Time rule (SPT). Note how products arriving during processing may force a reordering of the queue – depending on the characteristics of the dispatching rule.

Let us now discuss the extension of existing rules to deal with forecast data. As a starting point for our discussion we will consider family priority setting for the MASP rule:

$$j^* = \underset{j \in J; q_j > 0}{\arg\min} \frac{s_{j_0,j} + \sum_{i=1}^{q_j} p_{i,j}}{q_j} \tag{2}$$

According to this rule, system set-up is related to the choice of family j^* for which a minimum weighted workload is foreseen. Hereby workload is estimated by the sum of family set-up time ($s_{j0,j}$) and cumulative processing time ($\Sigma p_{i,j}$). Weights are related to queue length (q_j). Note that families for which the queue is empty are not considered in the evaluation. Alternatively, the extended MASP rule is formulated as:

$$\begin{aligned} &j^* = \underset{j \in J; q_j + l(H)_j > 0}{\arg\min} \frac{s_{j_0,j} + it_j + \sum_{i=1}^{q_j + l(H)_j} p_{i,j}}{q_j + l(H)_j} \\ &\text{with} \\ &it_j = 0 \text{ if } q_j > 0 \\ &it_j = \max(t_{1,j} - (t_0 + s_{j_0,j}), 0) \text{ if } q_j = 0 \end{aligned} \tag{3}$$

This rule reflects the advantages associated with inclusion of forecast data in decision making. Firstly, family priority setting is more accurate as estimates for workload and batch size are more precise. This follows from the fact that next to jobs in queue also jobs expected to arrive up to the information horizon (H) may be included in the process batch. The latter jobs are quantified as $l(H)_j$. Note that we assume an exhaustive strategy for batch formation here, see above. Secondly, families for which queue length equals zero, but for which a job will arrive soon, may be included in the dispatching decision ($q_j + l(H)_j > 0$). Finally, the possibility of pro-active set-ups may be considered for these families, i.e., start a set-up prior to actual job arrival. The latter two extensions imply the possibility of inserting idleness (it_j) for the system, i.e., not using the system for some time. Note how the computation of inserted idleness foresees in the possibility of a family set-up before the next arrival ($t_{1,j}$).

Using the above discussion on the extended MASP rule as a starting point, alternative rules for family-based dispatching can be adapted in a straightforward manner.

Fixed factors		
Family mix	Equal	
Inter-arrival time distribution	Negative exponential	
Set-up time distribution	Normal*(Standard deviation=0.25)	
Processing time distribution	3-Erlang(Mean=1.0)	
Experimental factors		
Number of families (F)	2;4;8;16	
Set-up to runtime ratio (S)	0.25;1.0	
	S= 0.25	S= 1.0
Mean inter-arrival times for S (I)	2.01[a]	2.60[a]
	1.77	2.28
	1.54	1.92
	1.31	1.57
	1.08[b]	1.23[b]
Job selection	SPT	
Family-based dispatching rules (R)	FCFAM LPTMM MAP MAS MASP MJ MS SWORK WORK	
Forecast error (A)	Normal (Mean=0) Standard deviation of 0;0.2·Mean inter-arrival time	
Rule extensions (E)		
0: Conventional family-based dispatching, no forecast data		
1: Update of family priorities with forecast data		
2: 1+ inserted idleness allowed		
3: 2+ pro-active set-ups allowed		

*Only positive values

a) 55% utilisation with R=FCFAM, F=2, Job selection=SPT

b) 95% utilisation with R=FCFAM, F=2, Job selection=SPT

Table 4.2: Overview of experimental factors and levels

4.4 Design of the simulation study

A simulation study was designed to consider the potential of forecast data for family-based dispatching. In this section we discuss the set-up of experiments, by describing alternative shop configurations in terms of fixed, and experimental factors. We conclude by mentioning simulation details.

4.4.1 Experimental design

As a starting point for discussing the experimental design we adopt Table 4.2. It specifies fixed factors, and experimental factors. Three series of experiments are foreseen. In the first series of experiments (I) we evaluate system performance for settings in which no forecast data is available/used. We consider existing rules for family-based dispatching, and the newly identified rules MAS and SWORK. On the other hand we will not consider the ECON rule, VCM3 and VCM5 here, because of their poor performance in previous studies (Frazier, 1996; Kannan and Ghosh, 1996a; Mahmoodi et al., 1992; Russell and Philipoom, 1991). For VCM3 and VCM5, however, we carried out a series of pilot experiments (not reported in this paper). They confirmed findings of Kannan and Ghosh on rule performance, which indicate that both VCM3 and VCM5 perform no better than the MJ rule – which is included in our study. Also experiments made clear that performance for VCM3 and VCM5 is very sensitive to the length of the information horizon. Recall, that the distinguishing feature of both rules relative to the MJ rule is their pre-selection of families, building on family presence in the preceding manufacturing stage. In our experiments we modelled this characteristic by considering the family types of jobs expected to arrive within the information horizon. Typically, a long horizon will not result in better performance for VCM3, as it boils down to MJ – after all most families will be pre-selected under these circumstances. For VCM5 a long horizon may result in worse performance, due to the direct link of horizon length and forced machine idleness. For shorter horizons, we could find no significant improvements of VCM3 and VCM5 relative to MJ.

The outcomes of the first series of experiments serve as a benchmark for a second series of experiments (II). In the latter series the presence and use of forecast data on future arrivals is assumed. To be able to exploit these data, existing control rules are extended. Also, we will consider the way specific extensions, like inserted idleness, and the possibility to schedule set-ups prior to actual arrivals, contribute to system performance.

Finally, in the third series (III) sensitivity of the extended rules is considered by studying system performance for (1) alternative settings of the information horizon (H), and (2) forecast error (A) in estimates of job arrival times.

4.4.2 Fixed factors

We consider a single machine shop with an unlimited buffer, see Section 3. Jobs are assumed to arrive according to a negative exponential distribution. Further, all part families have an equal share in the product mix. Job processing times are drawn from a third order Erlang distribution, with average 1. Assumptions with respect to family mix and arrival distribution are in line with most research in this field, see for example Jensen et al. (1996) or Wemmerlöv (1992). For the processing time distribution several choices have been considered. Where Wemmerlöv and Vakharia (1991) assume processing times to be almost constant, Reddy and Narendran (2003) use a negative exponential distribution. By employing a third order Erlang distribution we hold an intermediate position, in line with the research by Mahmoodi and Martin (1997).

4.4.3 Experimental factors

In this study 9 rules for family-based dispatching (R) are tested for a wide variety of system configurations. Shop configurations are defined by the settings for the set-up to runtime ratio (S), the number of product families (F) and the mean inter-arrival time (I). For all rules, jobs belonging to the same family are ordered according to shortest processing time (SPT).

The set-up to run-time ratio equals average set-up time divided by average processing time. Set-up times are determined using a bounded normal distribution with spread 0.25, leaving negative values out. Set-up times are drawn every time a set-up has been executed. For the set-up to run-time ratio two settings are foreseen: 0.25 (low), and 1.0 (high). Many other authors adopt similar settings, see for example Frazier (1996), Andrés et al. (2004) or Russell and Philipoom (1991).

The number of families ranges from 2 to 16. This range is in conformity with many other studies, see for example Jensen et al. (1996), Marsh et al. (1999), and Wemmerlöv (1992). The relevance of this factor follows from its foreseen impact on the number of set-ups.

Alternative shop load levels are chosen by adapting the mean inter-arrival time (I). They have been determined using a shop configuration with two product families (F=2), and FCFAM as a control rule (R=FCFAM) as a benchmark. The benchmark configuration was used to determine two levels of shop load for each setting of the set-up to runtime ration. The levels correspond with a 55% and 95% shop utilisation respectively, which includes both processing and set-ups. Three intermediate levels for shop load are found by interpolating average arrival intervals in a linear way. For example, if longest and shortest average arrival intervals derived from the aforementioned configurations would equal 30 and 10 time units respectively, intermediate settings would be 15, 20 and 25 time units. Previous research indicated that shop load has a major impact on (relative) performance of rules for family-based dispatching (Wemmerlöv and Vakharia, 1991).

Forecast errors are implemented using a normal distribution for modelling deviations from the actual arrival moment. We distinguish between the default case – no forecast error, i.e. a standard deviation of zero, and a setting with forecast errors equal to 0.2 times the mean inter-arrival time.

4.4.4 Simulation modelling

The software package that was used to carry out the simulation experiments is EM-Plant (EM-Plant TM 7.0, Stuttgart : Tecnomatix). The principles of object-oriented design underlying this language make it a flexible and efficient tool for model building. The performance for each heuristic was estimated using the replication deletion method, compare for example Hoover and Perry (1986), Law and Kelton (2000). A total of 35 runs were considered for each experiment. Each run concerned 15 000 jobs. The length of the warm-up period was determined using the Welch procedure, see Law and Kelton (2000). In accordance with the outcomes of the procedure it was set at 2500 jobs.

4.5 Simulation results

In the previous section the design of the simulation study has been discussed. In this section we will analyse the outcomes of the study. First we present results for the first series of experiments. This concerns shop configurations that assume local control, i.e., no forecast data on future job arrivals are included in dispatching decisions. Next, we consider the potential of using forecast data in decision-making, starting from a similar set of shop configurations. Finally, we consider robustness of the extended rules with respect to the length of the information horizon, and forecast errors.

Outcomes for all experiments are summarised in Appendix B, see Tables B1 and B2. Results for configurations that are close to instability are left out for further analysis, i.e., configurations for which system utilisation exceeds 98%. The results are analysed by considering two three-way Analyses of Variance (ANOVA), one for each ratio of set-up to run-time. They concern the mean inter-arrival time (I), the number of product families (F) and the family-based dispatching rule (R). We checked if any assumption underpinning ANOVA was violated and found no problems. ANOVA is robust with respect to inequality of variance, because we have equal cell sizes (Hair Jr. et al., 1995). We assumed normality of outcomes since kurtosis and skewness were generally near zero (Lindman, 1974).

4.5.1 Default settings – no forecast data

In Table 4.3 outcomes of the ANOVA are presented. They make clear that design issues like the number of product families (F) and the total product volume assigned to the system (I), i.e. shop load, have a large impetus on shop performance. Note also the interaction effect for the number of families and the shop load (FxI). These outcomes are in line with previous research, see for example Wemmerlöv (1992).

Set-up to run-time ratio	Source	df	F-value	p-value
0.25	Family selection rule (R)	8	29.12	<0.01
	Number of families (F)	3	2058.51	<0.01
	Mean inter-arrival time (I)	4	30493.10	<0.01
	R x F	24	10.20	<0.01
	R x I	32	5.08	<0.01
	F x I	10	646.57	<0.01
R-squared=.961	R x F x I	80	3.02	<0.01
1.0	Family selection rule (R)	8	491.19	<0.01
	Number of families (F)	3	161713.30	<0.01
	Mean inter-arrival time (I)	4	126372.30	<0.01
	R x F	24	172.02	<0.01
	R x I	32	148.44	<0.01
	F x I	8	17781.72	<0.01
R-squared=.995	R x F x I	64	77.89	<0.01

Table 4.3: ANOVA output for default configurations – no forecast data (α=0.01)

The ANOVA results make clear that the choice of control rule appears to become more relevant for a higher number of product families (RxF interaction). Let us consider this effect in somewhat more detail. Essentially, dispatching heuristics differ only with respect to family priority setting. After all, they all adopt the same job sequencing rule: SPT. For higher numbers of product families, the workload per family decreases. At the same time the rule for family priority setting gains weight relative to the job selection rule, as queue length diminishes. Likewise, for lower numbers of families the common job selection rule becomes dominant – resulting in a similar performance. The workload of the manufacturing system appears to have less relevance to the choice of rule (RxI interaction). The effect is most apparent at high work loads.

The ANOVA results supply us with a general picture on the relevance of control rules for system performance. Let us now consider relative performance of rules in somewhat more depth by means of multiple pairwise comparisons. Table 4.4 ranks dispatching rules for their flow performance across all shop configurations considered in the ANOVA. The overall level of significance within each setting is 0.01, which is in line with choices made for related studies, for example, Mahmoodi et al.(1990), Ruben et al. (1993). We use the Bonferroni adjustment procedure to correct α for individual comparisons (Hair Jr. et al., 1995). The rankings make clear that best results are obtained for MAS and MASP. Worst results are indicated for WORK and SWORK. Other rules hold intermediate positions. Further, rankings for both settings of the set-up to run-time ratio appear to be about the same. The only exception is the MAP rule. In principle these outcomes are not surprising. MASP is best informed in making dispatching decisions - all categories of local shop data are considered, see Section 3. The fact that MAS is also performing well, stresses the relevance of using set-up data in making dispatching decisions. In this respect it is striking to see the relative performance of MAP – that focuses on processing times, but not on set-up time – getting worse for higher set-up to run-time ratios. Bad performance for WORK and SWORK may be explained from their similarity to the so-called Longest Processing Time rule (LPT). In general LPT has proven to result in lengthy flow times (Conway et al., 1967).

Set-up to run-time ratio (S)			
S=0.25		S=1.0	
SWORK	4.194	SWORK	4.785
WORK	4.172	WORK	4.751
FCFAM	4.032	MAP	4.738
LPTMM	4.002	MS	4.702
MS	4.001	FCFAM	4.681
MJ	3.950	LPTMM	4.608
MAS	3.908	MJ	4.536
MAP	3.892	MASP	4.482
MASP	3.817	MAS	4.465

Table 4.4: Rankings of family-based dispatching rules (R) for default settings – no forecast data (α=0.01)

Results in Table 4.4 indicate that there is no single rule performing best overall. For a low set-up to run-time ratio best performance is indicated for MASP, while for a high set-up to run-time ratio no significant difference can be found for MAS and MASP. Also, more detailed analysis shows that MAS performs best when a high set-up to run-time ratio is combined with a high utilization, see Tables B1 and B2 in the appendix. Moreover, these tables make clear that also other rules may outperform MASP for specific configurations. Another way of looking at rule performance concerns its associated cost of data collection. MAS only requires data on set-up times and queue length, where MASP includes data on job processing times. At the same time the speed of decision making on dispatching may benefit from this characteristic.

4.5.2 Extended rules – use of forecast data

In the previous subsection we discussed simulation outcomes for shop configurations that are controlled relying on local data only. Here we will consider the potential of extended rules that include the notion of forecast data on future job arrivals, see Table 4.5. Similar to Table 4.4 the figures in Table 4.5 relate to overall averages of flow times for the rules over all alternative configurations considered in the ANOVA. For each rule three incremental extensions (1-3) are considered for their contribution to system performance, see Section 3:

1. Family priority setting, by including knowledge of jobs arriving within the information horizon, for those families with a non-zero queue length;
2. Including those families in the decision for which the queue is empty at the decision moment, but for which a next job will arrive within the information horizon;
3. Allowing for pro-active set-ups, i.e. set-ups may be started prior to job arrival.

S	E	Family-based dispatching rule (R)							
		MASP	**MAS**	**FCFAM**	**MS**	**MAP**	**MJ**	**SWORK**	**WORK**
0.25	0	3.817	3.908	4.032	4.001	3.892	3.950	4.194	4.172
	1	3.823	3.901	4.059	3.984	3.865	3.962	4.211	4.170
	2	3.845	3.924	4.031	4.026	4.233	8.202	11.535	11.784
	3	3.698	3.737	3.854	3.848	4.078	8.643	10.676	11.881
1	0	4.482	4.465	4.681	4.702	4.737	4.536	4.785	4.751
	1	4.440	4.430	4.684	4.697	4.721	4.496	4.737	4.706
	2	4.358	4.369	4.679	4.590	4.886	8.776	12.051	12.442
	3	3.940	3.924	4.220	4.171	4.555	8.516	11.595	12.117

Table 4.5: Contributions of rule extensions (E) to overall performance (α=0.01)

A first insight that may be derived from Table 4.5 is that the use of forecast data in dispatching may significantly improve system performance – by up to 10% for the average case. However, large differences among rules exist. Rules that already performed well in the absence of forecast data, like MASP and MAS, realise the largest improvements. On the other hand, results for those rules that showed poor performance in the earlier experiments, like WORK and SWORK, are even worse now. The latter effect may be explained from their (increased) likeness with the LPT rule. Note that we do not consider the LPTMM rule in our evaluation, since the LPTMM rule is quite similar to the SWORK rule.

Another interesting outcome of the experiments presented in Table 4.5 concerns the relative contribution of the three types of rule extension to system performance. Extensions 1 and 2 result in improvements of average flow times by 0-3%. The most important contribution, however, is to be associated with extension 3. The allowance for pro-active set-ups results in a reduction of average flow times by 5-8%.

		Number of families (F)			
S	**I**	**E**	2	**E**	8
0.25	1.31	2	3.416	0	4.795
		0	3.394	2	4.750
		1	3.367	1	4.714
		3	3.342	3	4.661
	2.01	0	1.856	1	2.116
		1	1.849	0	2.113
		2	1.840	2	2.097
		3	1.745	3	1.915
1	1.57	0	3.558	0	9.144
		1	3.550	1	8.902
		2	3.411	2	8.880
		3	3.237	3	8.685
	2.60	1	2.309	0	3.541
		0	2.304	1	3.513
		2	2.209	2	3.469
		3	1.889	3	2.779

Table 4.6: Overall performance rankings for extensions (E) of MASP for various shop configurations (α=0.01)

Above we addressed the general case in evaluating the added value of forecast data for dispatching decisions. Let us now consider the resulting performance improvements in somewhat more detail. We do so by considering the extended MASP rule as an illustrative example. This rule is chosen for its good performance as well as its comprehensive use of shop-floor data. Rule performance is studied for alternative set-up to run-time ratios, shop loads, and number of families. The results in Table 4.6 confirm the earlier finding that rule extensions 1 and 2 only realise minor improvements of system performance. Contributions of pro-active set-ups tend to be of most relevance over all experiments. A clear effect is to be associated with the setting of the set-up to run-time ratio – a ratio of 0.25 corresponds with a reduction of average flow times from 0-10%, whereas for a ratio of 1.0 performance improvements of 5-20% are indicated. Most of these benefits are realised for lower shop loads. This may be explained by the fact that these circumstances offer most possibilities to act pro-actively. The number of families does not seem to influence relative performance too much. Only for higher shop loads a higher number of families tends to reduce relative gains. This may be explained by the fact that under these circumstances there is little room for performance improvement through better quality decision-making – process batches tend to be large and possibilities to act proactively are hardly available.

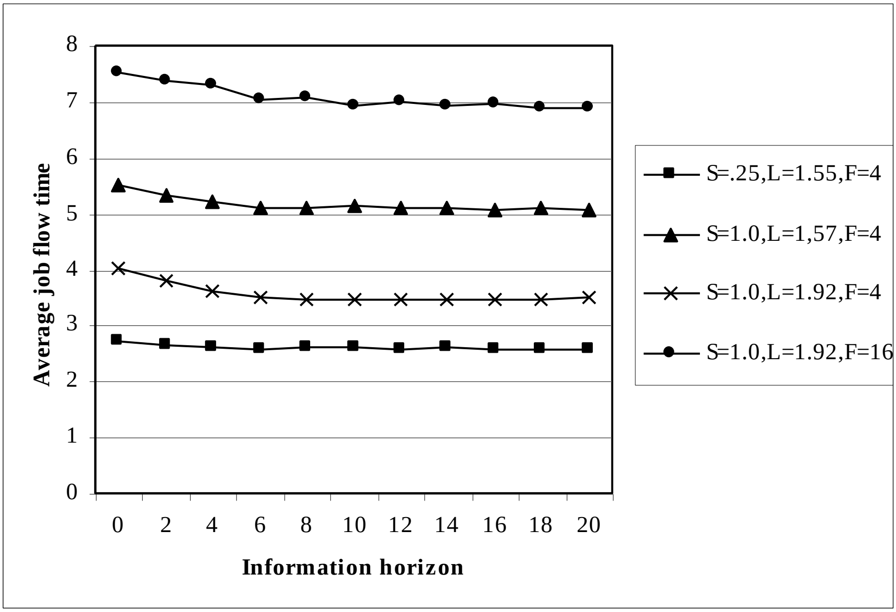

Figure 4.2 Performance of MASP for various settings of the information horizon (H)

4.5.3 Sensitivity analysis – information horizon and forecast error

In practice the use of data on future arrivals may imply investments in shop floor control systems. In principle, higher costs are associated with more precise data, or data that span a longer horizon. We study the issue of information horizon length by considering performance of the extended rules for various settings. Results for the MASP rule are displayed in Figure 4.2. They make clear that data collection on jobs arriving during the next 5 time units helps to realise maximum performance gains. This period corresponds to 5 times the average processing time. In fact most of the gains are already obtained during the first time units. In principle, this makes the rules quite suitable for practical application.

The effects of forecast error on system performance are shown in Table 4.7 for a limited set of rules. Rules concern the basic FCFAM rule, as well as the MAS and MASP rule, which indicated a good overall performance. In general, percentual performance differences caused by forecast errors may be considered small, i.e., less than 2% for most cases. MAS and MASP seem to be more sensitive to errors than FCFAM. This may be attributed to the extensive use of forecast data by the former rules. Not surprisingly, sensitivity increases with the horizon length. On the other hand, influence of the number of families and arrival intensity appears to be limited.

				Family-based dispatching rule		
S	**F**	**I**	**H**	**FCFAM**	**MAS**	**MASP**
0.25	4	1.55	1	1.289	-0.837	0.034
			2	1.243	1.834*	1.270
			4	0.176	1.796*	2.208*
			8	1.655	3.138*	3.015*
1	4	1.57	1	0.386	0.404	-0.164
			2	-0.137	0.830	1.647*
			4	1.160	1.309	0.698
			8	0.760	2.120*	1.802*
		1.92	1	0.788	0.391	-0.228
			2	0.923	1.408*	-0.114
			4	1.549*	1.990*	2.645*
			8	0.895	2.723*	3.526*
	16	1.92	1	-0.357	0.562	-0.750
			2	0.259	-0.418	1.307
			4	-0.686	2.553*	4.157*
			8	1.614	4.798*	4.539*

Table 4.7: Changes in flow performance caused by forecast errors in %.

(* indicates significant difference, α=0.01)

4.6 Conclusions

In this article we studied the use of family-based dispatching rules for shop control. More in particular, we considered the potential of including forecast data, on jobs yet to arrive at the shop, in decision-making. As a first step in our study we reviewed existing heuristics. Most of these rules follow a weighted shortest processing time scheme (WSPT) for making decisions on the product family to release next. Rule construction foresees in alternative choices with respect to the use of local data and weighting.

The second step in our study concerns the definition of a framework for decision-making. It highlights the essential elements of a control system, such as triggers, shop data, decision options and decision structure. It is used as a basis for generic rule extension towards the inclusion of data on future job arrivals. Developments in information technology, specifically real-time monitoring of manufacturing processes, enable this.

To study potential of the extended rules a simulation study has been carried out. Main findings are:

- Significant improvements of average job flow times are possible by employing near future arrival data. Rule extensions work out best for the MASP rule, i.e., the rule that includes family set-up time, job processing times, and queue length. A good alternative is the MAS rule. It has a similar structure, but excludes processing time from decision-making. On average both rules realise a reduction of up to 10% of average flow time per job;
- Most benefits of forecast data result from the possibility to start set-ups prior to the actual job arrival. In this way set-up effects on flow time are reduced. This adaptation explains about 80% of the performance improvements. The remaining 20% is related to:
    - More accurate family priority settings, as a more complete insight in actual processing batches is obtained;
    - The possibility to include families in the evaluation for which queue length equals zero, but for which a job will arrive soon;
- Most gains can be realised for low and moderate shop loads (improvements of up to 25%). Typically, under these circumstances it is often possible to act pro-actively by initiating set-ups prior to actual job arrivals;
- Maximum gains are realised using a short horizon of up to about 5 times the average job processing time;
- Rules appear to be quite robust with respect to forecast error.

Although this research has answered some questions with respect to the relevance of forecast data for family-based dispatching, many interesting directions for further research remain. For example, while our research focussed on flow time performance, it is worthwhile from a practical perspective to study due date and cost related criteria. Another interesting avenue concerns extensions of the model such as multiple, parallel machines, network configurations, and the restrictions set by secondary resources like tools and fixtures (for example Sienrich et al., 2001).

4.7 Acknowledgements

The authors would to thank the anonymous referees for their helpful comments and suggestions.

Appendix 4.A: Notation

Indexes:

i job identifier = 1,2,... for jobs in the system, ordered by arrival time

j family identifier = 1,2,...

Parameters:

j_0 current family, i.e., the family for which the machine has been set-up

H information horizon, specifying up to which moment forecasts on job arrivals are available

I_j set of jobs available for family j

J set of job families

N number of jobs processed

M large number

t_0 decision moment, i.e., the moment the dispatcher is triggered to make a decision

$p_{i,j}$ processing time of job i belonging to family j

q_j numbers of jobs in queue for family j at t_0

$s_{j0,j}$ set-up time required for changing the current set-up for family j_0 to meet requirements for family j

$t_{i,j}$ the arrival moment of job i for family j

λ_j arrival rate for family j

μ_j service rate for family j

Variables:

$ft_{i,j}$ flow time for job i belonging to family j

it_j inserted system idle time

$l(H)_j$ the number of forecasted jobs within family j that may be included in the process batch, following an exhaustive strategy

$w_{i,j}$ waiting time for job i belonging to family j

Appendix 4.B: Simulation results

S	I	F	E	FCFAM	LPTMM	MAP	MAS	MASP	MJ	MS	SWORK	WORK
				Family-based dispatching rule (R)								
0.25	1.08	2	0	9.687	9.278	9.664	9.246	8.994	9.003	8.928	9.330	9.428
			1	9.687	*	9.887	9.058	9.080	9.003	8.928	9.607	9.650
			2	9.626	*	9.358	9.554	8.884	9.810	9.092	10.014	10.071
			3	9.374	*	9.485	9.155	9.108	9.922	8.963	9.878	10.077
		4	0	12.942	12.752	13.682	13.124	13.480	12.846	14.486	13.287	12.888
			1	13.440	*	13.123	13.013	13.663	12.974	14.250	13.420	12.752
			2	13.156	*	15.728	12.682	13.782	14.672	15.081	15.677	15.689
			3	12.922	*	15.356	12.564	13.318	14.810	14.907	15.929	15.536
		8	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
		16	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
	1.31	2	0	3.381	3.316	3.367	3.362	3.394	3.350	3.395	3.325	3.326
			1	3.381	*	3.324	3.362	3.367	3.350	3.395	3.369	3.334
			2	3.335	*	3.366	3.361	3.416	4.252	3.370	4.322	4.425
			3	3.255	*	3.317	3.308	3.342	4.230	3.290	4.301	4.383
		4	0	4.208	4.187	4.041	4.162	3.994	4.190	4.220	4.346	4.364
			1	4.232	*	3.992	4.146	3.973	4.198	4.238	4.379	4.382
			2	4.195	*	4.230	4.150	3.969	6.459	4.205	7.392	7.583
			3	4.075	*	4.101	4.036	3.885	6.505	4.036	7.338	7.612
		8	0	5.225	5.083	4.811	4.886	4.795	5.046	5.154	5.558	5.613
			1	5.197	*	4.810	4.939	4.714	5.094	5.105	5.508	5.581
			2	5.263	*	5.625	4.965	4.750	9.645	5.109	13.058	13.226
			3	5.080	*	5.372	4.753	4.661	10.136	4.887	13.062	13.336
		16	0	6.551	6.525	5.837	5.849	5.489	6.258	6.017	7.301	7.261
			1	6.530	*	5.758	5.889	5.507	6.219	5.985	7.247	7.242
			2	6.480	*	7.312	5.949	5.857	14.706	5.916	23.254	23.763
			3	6.213	*	7.207	5.596	5.598	15.442	5.714	23.452	23.811
	1.55	2	0	2.416	2.423	2.416	2.415	2.418	2.413	2.419	2.412	2.414
			1	2.416	*	2.415	2.415	2.405	2.413	2.419	2.415	2.407
			2	2.419	*	2.443	2.428	2.416	3.410	2.422	3.522	3.633
			3	2.312	*	2.332	2.362	2.345	3.352	2.303	3.498	3.608
		4	0	2.865	2.856	2.748	2.822	2.735	2.844	2.842	2.931	2.963
			1	2.858	*	2.745	2.833	2.723	2.857	2.833	2.929	2.954
			2	2.856	*	2.869	2.848	2.725	5.403	2.799	6.511	6.721
			3	2.716	*	2.708	2.699	2.595	5.569	2.653	6.581	6.783

Table 4.B1: Full factorial results for S=0.25

0.25		8	0	3.293	3.275	2.948	3.140	2.950	3.245	3.165	3.553	3.519
			1	3.290	*	2.945	3.180	2.941	3.253	3.158	3.542	3.484
			2	3.258	*	3.285	3.173	2.972	8.397	3.157	12.180	12.451
			3	3.087	*	3.104	2.962	2.792	9.083	2.922	12.336	12.600
		16	0	3.648	3.769	3.142	3.360	3.106	3.612	3.398	4.159	4.097
			1	3.648	*	3.158	3.383	3.099	3.647	3.415	4.131	4.104
			2	3.630	*	3.689	3.438	3.161	13.115	3.400	22.264	22.770
			3	3.423	*	3.540	3.146	2.937	14.188	3.146	22.790	23.076
	1.78	2	0	2.067	2.067	2.050	2.050	2.055	2.057	2.069	2.061	2.065
			1	2.067	*	2.056	2.050	2.054	2.057	2.069	2.064	2.052
			2	2.055	*	2.061	2.073	2.048	3.140	2.049	3.289	3.394
			3	1.949	*	1.962	1.984	1.944	3.058	1.939	3.250	3.361
		4	0	2.359	2.360	2.263	2.336	2.256	2.340	2.342	2.429	2.422
			1	2.359	*	2.269	2.335	2.263	2.349	2.338	2.418	2.414
			2	2.355	*	2.348	2.336	2.263	5.139	2.319	6.404	6.601
			3	2.190	*	2.189	2.196	2.106	5.343	2.164	6.474	6.695
		8	0	2.608	2.628	2.401	2.535	2.399	2.587	2.524	2.774	2.762
			1	2.608	*	2.399	2.528	2.397	2.589	2.505	2.771	2.743
			2	2.583	*	2.581	2.536	2.396	8.083	2.521	12.123	12.430
			3	2.395	*	2.380	2.327	2.216	8.886	2.306	12.451	12.715
		16	0	2.766	2.874	2.503	2.634	2.473	2.759	2.631	3.096	3.069
			1	2.769	*	2.501	2.647	2.468	2.753	2.648	3.093	3.052
			2	2.782	*	2.770	2.681	2.480	12.702	2.641	22.547	23.066
			3	2.550	*	2.554	2.409	2.261	14.021	2.395	22.893	23.337
	2.01	2	0	1.857	1.854	1.860	1.859	1.856	1.851	1.861	1.854	1.858
			1	1.857	*	1.856	1.859	1.849	1.851	1.861	1.852	1.849
			2	1.859	*	1.860	1.859	1.840	3.028	1.854	3.199	3.350
			3	1.740	*	1.756	1.773	1.745	2.930	1.738	3.155	3.300
		4	0	2.079	2.095	2.030	2.067	2.026	2.084	2.069	2.125	2.134
			1	2.083	*	2.028	2.067	2.020	2.092	2.069	2.122	2.124
			2	2.083	*	2.077	2.083	2.011	5.040	2.061	6.433	6.706
			3	1.917	*	1.903	1.920	1.858	5.298	1.894	6.567	6.780
		8	0	2.250	2.277	2.123	2.205	2.113	2.248	2.202	2.378	2.363
			1	2.257	*	2.127	2.213	2.116	2.247	2.200	2.378	2.359
			2	2.255	*	2.244	2.221	2.097	8.048	2.198	12.400	12.681
			3	2.042	*	2.029	2.016	1.915	8.865	1.983	12.715	13.006
		16	0	2.369	2.425	2.172	2.291	2.165	2.363	2.289	2.564	2.547
			1	2.375	*	2.174	2.293	2.167	2.363	2.290	2.556	2.571
			2	2.365	*	2.345	2.301	2.144	12.588	2.278	23.033	23.553
			3	2.133	*	2.113	2.059	1.938	13.933	2.031	23.504	23.849

- indicates unstable system; * equal to performance E=0

				Family-based dispatching rule (R)								
S	**I**	**F**	**E**	FCFAM	LPTMM	MAP	MAS	MASP	MJ	MS	SWORK	WORK
1.0	1.23	2	0	6.170	6.066	6.228	6.061	6.035	6.055	6.038	6.124	6.120
			1	6.167	*	6.145	6.061	6.083	6.024	6.038	6.077	6.096
			2	6.091	*	6.048	5.910	5.862	6.429	6.219	6.466	6.599
			3	5.944	*	5.987	5.842	5.759	6.342	6.090	6.397	6.581
		4	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
		8	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
		16	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
	1.57	2	0	3.534	3.526	3.557	3.526	3.558	3.551	3.575	3.521	3.536
			1	3.534	*	3.550	3.526	3.550	3.551	3.575	3.548	3.530
			2	3.547	*	3.493	3.388	3.411	4.223	3.419	4.234	4.355
			3	3.298	*	3.324	3.214	3.237	4.122	3.209	4.077	4.284
		4	0	5.666	5.530	5.922	5.506	5.528	5.492	5.910	5.677	5.685
			1	5.674	*	5.855	5.425	5.449	5.461	5.899	5.616	5.619
			2	5.671	*	5.973	5.315	5.354	7.261	5.788	8.154	8.452
			3	5.425	*	5.925	5.048	5.114	7.026	5.638	7.767	8.163
		8	0	9.280	8.553	10.964	8.526	9.144	8.518	10.430	9.129	9.169
			1	9.301	*	10.912	8.331	8.902	8.339	10.363	8.988	8.994
			2	9.333	*	11.569	8.304	8.880	11.902	10.177	15.179	15.539
			3	9.133	*	12.082	7.963	8.685	11.457	10.450	14.555	15.008
		16	0	-	-	-	-	-	-	-	-	-
			1	-	-	-	-	-	-	-	-	-
			2	-	-	-	-	-	-	-	-	-
			3	-	-	-	-	-	-	-	-	-
	1.92	2	0	2.843	2.834	2.826	2.830	2.835	2.830	2.836	2.830	2.829
			1	2.843	*	2.828	2.830	2.832	2.830	2.836	2.842	2.824
			2	2.834	*	2.785	2.707	2.710	3.756	2.705	3.796	3.977
			3	2.512	*	2.534	2.478	2.451	3.590	2.428	3.580	3.858
		4	0	4.132	4.074	4.109	4.012	4.018	4.040	4.138	4.159	4.176
			1	4.132	*	4.109	3.986	3.967	3.999	4.154	4.103	4.145
			2	4.125	*	4.154	3.910	3.861	6.481	3.953	7.589	7.904
			3	3.695	*	3.836	3.476	3.496	6.262	3.607	7.194	7.659

Table 4.B2: Full factorial results for S=1.0

	8	0	5.846	5.672	5.771	5.477	5.483	5.551	5.818	6.035	5.931
		1	5.843	*	5.759	5.394	5.386	5.478	5.803	5.946	5.847
		2	5.855	*	6.067	5.366	5.320	10.620	5.624	14.431	14.884
		3	5.400	*	5.798	4.855	4.891	10.272	5.165	13.904	14.433
	16	0	8.332	8.072	8.282	7.457	7.527	7.808	8.061	8.787	8.626
		1	8.331	*	8.294	7.376	7.387	7.686	8.062	8.633	8.477
		2	8.264	*	9.258	7.393	7.392	17.024	7.932	26.644	27.347
		3	7.826	*	8.975	6.867	6.949	16.604	7.387	25.921	26.635
2.28	2	0	2.496	2.496	2.492	2.483	2.497	2.494	2.499	2.495	2.496
		1	2.496	*	2.489	2.483	2.491	2.494	2.499	2.495	2.493
		2	2.492	*	2.460	2.388	2.380	3.611	2.372	3.691	3.876
		3	2.111	*	2.166	2.109	2.083	3.409	2.042	3.426	3.746
	4	0	3.422	3.399	3.374	3.354	3.314	3.371	3.411	3.468	3.451
		1	3.422	*	3.359	3.335	3.302	3.353	3.412	3.438	3.426
		2	3.415	*	3.367	3.272	3.217	6.319	3.265	7.620	7.992
		3	2.863	*	2.913	2.786	2.729	6.077	2.753	7.222	7.731
	8	0	4.385	4.378	4.229	4.188	4.119	4.281	4.268	4.549	4.505
		1	4.381	*	4.226	4.166	4.095	4.244	4.262	4.488	4.458
		2	4.377	*	4.388	4.134	4.028	10.279	4.181	14.534	15.018
		3	3.740	*	3.775	3.486	3.425	10.011	3.505	14.023	14.681
	16	0	5.400	5.556	5.074	5.024	4.850	5.289	5.119	5.869	5.717
		1	5.420	*	5.061	4.989	4.844	5.254	5.127	5.800	5.665
		2	5.455	*	5.396	5.036	4.801	16.346	5.050	26.935	27.551
		3	4.714	*	4.696	4.249	4.068	16.049	4.247	26.095	27.063
2.6	2	0	2.304	2.308	2.310	2.308	2.304	2.309	2.308	2.312	2.310
		1	2.304	*	2.313	2.308	2.309	2.309	2.308	2.307	2.310
		2	2.310	*	2.291	2.218	2.209	3.588	2.204	3.702	3.895
		3	1.900	*	1.957	1.911	1.889	3.337	1.851	3.392	3.742
	4	0	3.046	3.053	3.013	2.998	2.993	3.041	3.040	3.096	3.084
		1	3.043	*	3.008	2.994	2.981	3.021	3.041	3.080	3.068
		2	3.056	*	3.024	2.943	2.889	6.317	2.923	7.736	8.142
		3	2.452	*	2.497	2.412	2.366	6.046	2.360	7.341	7.890
	8	0	3.725	3.754	3.604	3.608	3.541	3.672	3.649	3.873	3.814
		1	3.728	*	3.596	3.597	3.513	3.641	3.637	3.846	3.802
		2	3.707	*	3.665	3.560	3.469	10.225	3.557	14.798	15.359
		3	3.005	*	2.989	2.858	2.779	9.939	2.809	14.190	14.939
	16	0	4.324	4.459	4.045	4.084	3.974	4.278	4.127	4.634	4.563
		1	4.332	*	4.027	4.073	3.952	4.251	4.129	4.590	4.542
		2	4.324	*	4.229	4.068	3.936	16.039	4.066	27.304	28.183
		3	3.495	*	3.422	3.230	3.119	15.718	3.193	26.429	27.456

- indicates unstable system; * equal to performance E=0