Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

SCHEDULING JOB FAMILIES ON NON-IDENTICAL PARALLEL MACHINES WITH TIME CONSTRAINTS

Ali Obeid
Stéphane Dauzère-Pérès
Claude Yugma

Department Manufacturing Sciences and Logistics
Centre Microélectronique de Provence
École des Mines de Saint-Étienne
Site Georges Charpak
880 Avenue de Mimet
Gardanne, F-13541, France

ABSTRACT

This paper studies the scheduling of lots (jobs) of different product types (job families) on parallel machines, where not all machines are able (i.e. are qualified) to process all job families (non-identical machines). This is known in the literature as scheduling with machine eligibility restrictions. A special time constraint, associated with each job family, should be satisfied for a machine to remain qualified for processing a job family. This constraint imposes that there must be at most a given time interval (threshold) between processing two jobs of the same job family, on a qualified machine. A machine is considered to be qualified to process a certain job if and only if, at a given time instant t, the time threshold corresponding to the job family of the job is not violated. This problem comes from semiconductor manufacturing, when Advanced Process Control constraints are considered in scheduling problems, as for example in the photolithography area. To solve this problem, a Time Indexed Mixed Integer Linear Programming (MILP) model was proposed and solved in a previous paper. A new adapted model will be provided in this paper. A bicriteria objective function, that includes scheduling and qualification criteria, is considered. Dedicated heuristics are proposed. Numerical experiments are conducted to compare heuristic and exact solutions.

1 INTRODUCTION

Semiconductor manufacturing is getting more and more competitive and industries are looking for strategies to improve productivity, decrease cost and enhance quality. Advanced Scheduling and Advanced Process Control (APC) systems support these objectives. Scheduling means assigning jobs to machines and sequencing jobs on machines to minimize some given objectives under a set of constraints. Hence, optimized scheduling helps to increase productivity. Process control is widely used to enhance the quality of products by compensating for process drifts and adjusting machine parameters. The collection of data at both machine and process levels helps in the detection of current process drifts and/or machine degradation, as well as in the prediction of possible faults. Scheduling and control could be considered as mutually related issues in semiconductor manufacturing. For example, to control, we may need information on scheduling, and to schedule in an effective way, we need information on which machines each operation can be processed.

Scheduling of lots has a direct impact on equipment utilization, cycle times, delivery times, etc. For example, effective scheduling decisions would send tasks to the right machines so as to avoid idle times and improve machine utilization. Moreover, semiconductor fabrication plants have characteristics that make scheduling a very complex issue (see (Kumar 1993) or (Moench, Fowler, Dauzère-Pérès, Mason, and Rose 2011) for instance). Advanced Process Control (APC) aims at controlling processes and equipment to reduce variability, to increase equipment efficiency, to collect and classify information on equipment, etc. APC is usually associated to the combination of Statistical Process Control (SPC), Fault Detection and Classification (FDC), Run to Run control (R2R), and more recently Virtual Metrology (VM).

In semiconductor manufacturing facilities (fabs), a wafer is the chip holder at the end of the manufacturing process. Lots contain 25 wafers or less and are processed in various work areas with different characteristics. In this paper, lots will be called jobs, and lots of the same product type will be called job family. Given the re-entrant nature of manufacturing processes, scheduling is often locally optimized in each work area. (Kubiak, Lou, and Wang 1996) study the problem of scheduling a reentrant job shop with different job families. They show that the shortest processing time (SPT) job order is optimal for the single machine reentrant shop under certain assumptions. An example is scheduling in the photolithography area that can be seen as a scheduling problem on parallel machines with job family setups (also called s-batching). A setup is required before starting the first job of a family, but no setup is necessary between two jobs of the same family. For example, the change of reticles in the photolithography workshop necessitates a family dependent setup time. Although research has been performed on this problem, very little has been done to integrate APC constraints. In R2R control for instance, a R2R controller uses data from past process runs to adjust settings for the next run as presented for example in (Musacchio, Rangan, Spanos, and Poolla 1997). Note that a R2R controller is associated to one machine and one job family. A machine can usually process a limited number of job families, that are said to be *qualified* on the machine. Machines are thus non-identical. In addition, in order to keep its parameters updated and valid, a R2R control loop should regularly get data. This imposes an additional constraint on scheduling, since jobs of the same family have to be scheduled within a maximum time interval on each machine on which the family is qualified. The value of the time threshold depends on several criteria such as the process type (critical or not), the equipment type, the stability of the control loop, etc. If this time constraint is not satisfied, a qualification run is required to be able to process again the job family on the machine. We assume in our problem that this qualification run cannot be performed within the scheduling horizon. Figure 1 illustrates this time constraint. In the first two cases, a job of a given family is started during the time interval corresponding to its family, and hence the machine will still be qualified to process jobs of the same family for another time interval. The third case represents the situation where a machine is no longer qualified to process such a job family, and this is because no job is scheduled during the considered interval.

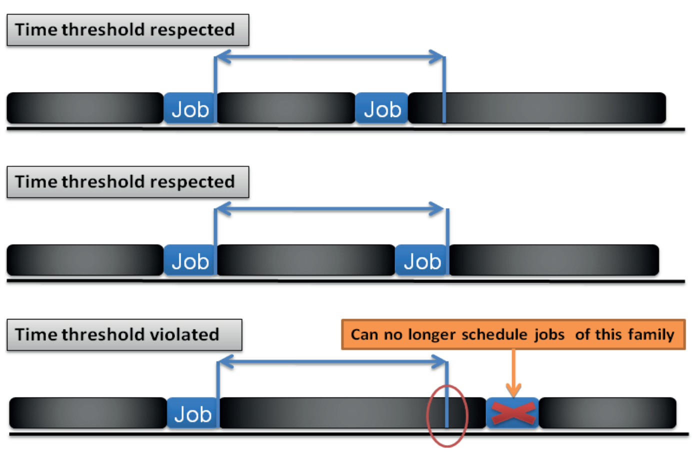

Figure 1: Time constraint.

The rest of this paper is organized as follows. In Section 2, we provide a literature review. Section 3 proposes a new MILP based on the one proposed in (Obeid, Dauzère-Pérès, Yugma, and Ferreira 2010).

Section 4 discusses the complexity of the problem, which is shown to be NP-hard. In Section 5, the heuristics we developed to solve the problem for large instances are presented. Computational experiments conducted on randomly generated data sets are shown and discussed in Section 6. Section 7 concludes and provides some perspectives for further research.

2 LITERATURE REVIEW

There are very few articles which deal with scheduling decisions while integrating APC constraints. The impact of APC on scheduling performances is analyzed by (Li and Qiao 2008). They also study the scheduling of job families on parallel machines. However, they consider that machines are identical, that qualification runs can be scheduled and that the threshold between two jobs of the same family is given in number of jobs. We consider non-identical parallel machines and assume that qualification runs cannot be scheduled and will be performed after the scheduling horizon. The problem becomes more complicated, since the assignment of jobs to machines is critical to avoid qualification runs. Finally, we consider a threshold expressed in time instead of number of jobs. Both threshold types are actually relevant and are related. Another example of integration of APC constraints in scheduling decisions can be found in (Detienne, Dauzère-Pérès, and Yugma 2010), where measurement operations are optimally scheduled to minimize the risk of losing products in jobs. Hence, we address a new scheduling problem in which there is a time constraint on jobs of the same family, i.e. the time interval between two consecutive jobs of the same family should be smaller than a given threshold. As mentioned above, this constraint is inspired from the needs of APC systems, and in particular Run-To-Run (R2R) control loops for a given product type on a machine, that require to regularly collect data for product types on machines. In what follows, we briefly review some existing topics which are related to our problem.

2.1 Dynamic Deadlines / Due Dates

A deadline d is a point in time by which the task (job) must absolutely complete. The deadline can be hard, soft, or firm. Scheduling with dynamic deadlines exists in the literature under various topics and rarely in the domain of semiconductor manufacturing. Topics concerning mobile communications are one of the fields where studies about dynamic deadlines can be found.

(Somasundara, Ramamoorthy, and Srivastava 2007) study the problem of Mobile Element Scheduling (MES). The mobile element visits the nodes of a wireless sensor network to collect their data before their buffers are full. In addition, as soon as a node is visited, its deadline (time before which it should be revisited to avoid buffer overflow) is updated. Thus, deadlines are *dynamically* updated as the mobile element performs the job of data gathering. The idea is to find a schedule for a controlled mobile element so that there is no data loss due to buffer overflow. Other examples of dynamic deadlines problems in the same domain include mobile element for data collection, battery charging, and calibration (Kallapur and Chiplunkar 2010). (Caccamo, Lipari, and Buttazzo 1999) address the problem of scheduling hybrid tasks in a shared resource environment (hard periodic and soft aperiodic) with dynamic deadlines. They develop an algorithm which finds the optimal solution for a schedule of hybrid tasks on shared resources. The problem basically is a problem of task scheduling in computer operating systems. The problem of lot release control and scheduling in wafer fabs producing multiple products with due dates, was tackled in (Kim, Kim, Lim, and Jun 1998). The authors suggest several new rules to minimize the mean tardiness. They show that new dispatching rules work better in terms of tardiness of orders other than existing rules such as the EDD (earliest due date) rule and other well-known dispatching rules for multi-machine scheduling.

By analogy, the idea of dynamic deadlines exists in our problem under the form of job family thresholds, where each family threshold creates a deadline at the machine qualification level. This threshold is dynamically updated with a fixed value once a job of a given family is scheduled on a qualified machine.

2.2 Parallel Machines with Objectives

Parallel machine scheduling problems are frequent in semiconductor manufacturing. A wafer fab can be modeled as a complex job shop (Mason, Fowler, and Carlyle 2002), which contains unrelated parallel machines with sequence-dependent setup times and dedications, parallel batch machines, re-entrant flows, and ready times of the jobs (Moench, Fowler, Dauzère-Pérès, Mason, and Rose 2011). Classical scheduling objective functions include the minimization of makespan (C_{max}), total weighted tardiness ($\sum w_j T_j$), etc (Pinedo 2009). The makespan is schedule dependent when there are m machines in parallel. In such problems, scheduling with LPT (longest processing time first) is usually used as a scheduling rule, but it does not guarantee an optimal solution. Another classical scheduling objective function is the minimization of the total completion time ($\sum C_j$). Rules such as SPT (shortest processing time first) are used to tackle such problems. When the total weighted completion time ($\sum w_j C_j$) is to be minimized on parallel machines, the problem is NP-Hard.

2.3 Setup Times in Scheduling

A setup is a non-productive period of time which usually models operations to be carried out on machines after processing a job to leave them ready for processing the next job in the sequence. An extended survey on scheduling problems with setup times or costs was done in (Allahverdi, Ng, Cheng, and Kovalyov 2008). The authors provide an extensive review of the scheduling literature on models with setups covering more than 300 papers. They classify, throughout their paper, scheduling problems into those with batching and non-batching considerations, and with sequence independent and sequence dependent setup times. They also categorize the literature according to shop environments, including single machine, parallel machines, flow shop, no-wait flow shop, flexible flow shop, job shop, open shop, and others.

In addition, scheduling jobs on parallel machines with sequence-dependent family setup times is also studied in (Eom, Shin, Kwun, Shim, and Kim 2002). The authors propose a three-phase heuristic to minimize the total weighted tardiness of a set of tasks with known processing times, due dates, weights and family types for parallel machines. However, they consider the case of identical machines in a liquid crystal display (LCD) manufacturing process where the setup time is longer than the processing time. Moreover, (Schaller, Gupta, and Vakharia 2000) study the problem of scheduling a flowline manufacturing cell with sequence dependent family setup times. The objective is to minimize the makespan. The authors show that the problem is NP-hard in the strong sense and they develop several heuristic algorithms. In this paper, we consider non-identical parallel machines and minimize a bi-criteria objective function where the sum of completion times is one criterion.

3 MATHEMATICAL MODELING

3.1 Definition

Our objective is to schedule, on an horizon discretized in T periods, a set N of jobs of different families on a set M of parallel machines. The set of job families is denoted F, and $f(i)$ is the family of job i. We assume that the processing times p_f of all jobs in family f are equal. Machines are not qualified to process all jobs families. The qualification of a machine may be lost at a certain point in time due to a change in the level of confidence on the machine. A setup time s_f on a machine is necessary to change from one job of a family f to another job of family f', where $f \neq f'$. Finally, Run-To-Run control constraints are considered through a parameter γ_f, which corresponds to the maximum time interval (called time threshold in the sequel) between the processing of two jobs of family f on a qualified machine. Usually, if this constraint is not satisfied, a qualification run will be required to qualify again the machine for f. In the sequel, we consider that the machine will not be available to process any job of family f if the qualification cannot be maintained. The objective is to optimize a scheduling criterion, the sum of the completion times, while minimizing the number of disqualifications of families on machines, i.e. a bicriteria scheduling problem.

We first recall in Section 3.2 the notations used in the two time indexed mathematical models introduced in (Obeid, Dauzère-Pérès, Yugma, and Ferreira 2010). A new modified family based model is then presented in Section 3.3. In this new model, the effect of the time horizon on the total number of qualification losses is eliminated.

3.2 Notations

The parameters are:

T: Number of periods in the time horizon,
N: Set of jobs,
M: Set of machines,
F: Set of job families,
n_f: Number of jobs in family f,
$M(f)$: Set of qualified machines to process jobs in family f ($M(f) \subset M$),
p_f: Processing time of jobs in family f,
s_f: Setup time of jobs in family f,
γ_f: Time threshold for job family f.

The decisions variables are:

$x^m_{f,t} = 1$ if a job of family f starts at period t on machine m, and 0 otherwise,
C_f: Sum of the completion times of jobs in f,
$y^m_{f,t} = 1$ if the time threshold is not satisfied for family f on machine m at period t, i.e. a qualification run is required, and 0 otherwise,
$Y^m_f = 1$ if the time threshold is not satisfied for family f on machine m at the end of horizon, and 0 otherwise.

It is important to recall that, if the time threshold γ_f is not satisfied for job family f on machine m from a time instant t, we assume that the qualification run required on machine m cannot be performed within the time horizon. In this case, we suppose that no job in family f can be processed on m after t.

3.3 A New Mathematical Programming Model

The new model is given below. Although only Constraint (6) is new compared to the model introduced in (Obeid, Dauzère-Pérès, Yugma, and Ferreira 2010), all other constraints are recalled.

$$\sum_{m\in M(f)}\sum_{t=1}^{T-p_f+1} x^m_{f,t} = n_f \qquad \forall f \in F \tag{1}$$

$$\sum_{m\in M(f)}\sum_{t=1}^{T-p_f+1} t.x^m_{f,t} + n_f(p_f-1) \leq C_f \qquad \forall f \in F \tag{2}$$

$$\sum_{\tau=t-p_f+1}^{t} x^m_{f,\tau} \leq 1 \qquad \forall t = 1\dots T, \forall f \in F, \forall m \in M(f) \tag{3}$$

$$\sum_{\tau=t-p_f-s_{f'}+1}^{t} x^m_{f,\tau} + n_f.x^m_{f',t} \leq n_f \qquad \forall t = 1\dots T, \forall (f,f') \in F\times F \tag{4}$$

$$\text{s.t. } f \neq f', \forall m \in M(f)\cap M(f')$$

$$\sum_{\tau=t-\gamma_f+1}^{t} x^m_{f,\tau} + y^m_{f,t} \geq 1 \qquad \forall f \in F, \forall m \in M(f), \forall t = \gamma_f \dots T \tag{5}$$

$$y^m_{f,t-1} - 1 + \frac{1}{T-(t-1)}\sum_{\tau=t}^{T}\sum_{f'\in F}\sum_{m'\in M(f')} x^{m'}_{f',\tau} \leq Y^m_f \qquad \forall t = 2\dots T, \forall f \in F, \forall m \in M(f) \tag{6}$$

$$x^m_{f,t} \in \{0,1\} \qquad \forall t = 1\dots T, \forall f \in F, \forall m \in M(f) \tag{7}$$

$$y^m_{f,t} \in \{0,1\} \qquad \forall t = 1\dots T, \forall f \in F, m \in M(f) \tag{8}$$

$$Y^m_f \in \{0,1\} \qquad \forall f \in F, m \in M(f) \tag{9}$$

Constraint (1) guarantees that n_f jobs are scheduled for family f in the scheduling horizon. Constraint (2) is used to determine C_f. Constraints (3) and (4) model the fact that only one job of a family f is processed at a time on a machine. Constraint (3) is written for jobs of the same family, i.e. for which setup time is not required, whereas Constraint (4) is associated to pairs of jobs of two different families for which setup times are necessary. Constraint (5) ensures that either the time threshold is always satisfied for a job family f qualified on machine m, or a qualification run is necessary, i.e. $y^m_{f,t} = 1$. Constraint (6) guarantees that, if a machine is disqualified at period t, then it is also disqualified in the following periods. It also ensures that the number of machine qualification losses at the end of time horizon T, i.e. $\sum_{f\in F}\sum_{m\in M(f)} Y^m_f$, is independent of the time horizon. This sum is considered as one criterion of the objective function. In the model proposed in (Obeid, Dauzère-Pérès, Yugma, and Ferreira 2010), the number of machine qualification losses qualifications was dependent on the time horizon. This is because, to avoid losing a machine qualification, it was necessary to maintain this qualification on the machine from time 0 to time T. In Constraint (6), it is no longer necessary to maintain a qualification on the machine if no job is started on any machine in the remainder of the horizon, i.e. $\frac{1}{T-(t-1)}\sum_{\tau=t}^{T}\sum_{f'\in F}\sum_{m'\in M(f')} x^{m'}_{f',\tau} = 0$. Hence, the number of machine qualification losses does not depend on T (if T is large enough). Constraints (7), (8) and (9) ensure that variables $x^m_{i,t}$, $y^m_{f,t}$ and Y^m_f are binary.

3.4 Objective Function

A bicriteria objective function is minimized that is a weighted sum of two types of criteria. The first type corresponds to a scheduling criterion which is the sum of completion times of families $\sum_{f\in F} C_f$. The second type is associated to the number of machine disqualifications $\sum_{f\in F}\sum_{m\in M(f)} Y^m_f$. The objective function is defined as follows: $\alpha\sum_{f\in F} C_f + \beta\sum_{f\in F}\sum_{m\in M(f)} Y^m_f$, where α and β are weights that model the trade-off between both criteria. However, in this paper, we consider a lexicographical order where the number of qualification runs is prioritized over a pure scheduling criterion, i.e. β is chosen large enough compared to α ($\alpha = 1$, $\beta = |N| * T$), so that improving the scheduling criterion is not preferable to an additional disqualification. According to the $\alpha|\beta|\gamma$ notation introduced to classify scheduling problems by (Graham, Lawler, Lenstra, and Kan 1979), this problem is noted $P_m|ST_{si,b}|\sum C_j$.

4 COMPLEXITY

We recall that a machine is said to be *qualified* to process a given job family if it satisfies the necessary conditions to process this job. Our problem consists of scheduling $|N|$ jobs of $|F|$ job families (where $|N|$ and $|F|$ are the cardinality of the sets N and F, respectively) on $|M|$ non-identical parallel machines (each machine has its own set of qualifications), with s_f as the setup time of family $f \in F$ and p_f as the associated processing time. A *time interval* is associated with each family during which at least one job of this family must be scheduled. We called this time constraint a *threshold*. The value of a family threshold is given by γ_f, and n_f is the number of jobs in family f. Initially, we must send a job of family f to a qualified machine m during the interval $[0, \gamma_f]$. Otherwise, the machine will no longer be available to process such a job family (the machine is disqualified). The objective function of our problem is bi-criteria, in which the aim is to minimize both the sum of completion times of families $(\sum_{f \in F} C_f)$ and the number of machine disqualifications $(\sum_{f \in F} \sum_{m \in M(f)} Y_f^m)$. Let us consider the following special case of our problem where the setup time is set to zero for all job families $(s_f = 0, \forall f \in F)$. The threshold associated to each job family defined by γ_f is considered as the deadline d_f of a given family. We recall that all the jobs which belong to the same family have the same deadline (threshold), and these deadlines are considered on all the machines to be equal to the time horizon (T). Hence, it is no longer possible to lose the qualification of any machine. We assume that all machines are qualified to process all job families. Hence, the qualification part in the objective function has no effect, and the objective function becomes the classical known function of minimizing the sum of completion times. Moreover, the machines are identical in terms of qualifications. Therefore, the problem is reduced to the problem of scheduling n jobs on m arbitrary machines, with p_f as the processing time of job family f which is the same for all jobs in the family, and $(\sum_{f \in F} C_f)$ as an objective function. (Webster 1997) proves that this problem is unary NP-hard, and therefore our problem is also NP-hard.

5 HEURISTICS

5.1 List Heuristics

The jobs of different families on the parallel machines are scheduled by using two priority rules: Earliest Time Threshold (ETT) and Shortest Processing Time (SPT). The Earliest Time Threshold rule first schedules jobs of the family with the earliest threshold, i.e. the most urgent family first to keep the machine qualified.

5.2 Recursive Heuristic

The general idea of this heuristic is to schedule jobs by accepting each time one qualification loss or more. More precisely, we consider the solution obtained by the list heuristic, and reapply the heuristic after changing the initial qualification scheme. The perturbations in the qualification scheme are chosen from the set of disqualified machines in the solution obtained by the list heuristic. In other words, after scheduling the jobs using the list heuristic, we examine the resulting solution to verify whether the machines are still capable (qualified) to execute the jobs, i.e. whether thresholds are satisfied or not. If this is not the case, we change the data to accept some threshold violations, and reapply the list heuristic recursively to improve the solution. Note that this iterative heuristic could also be applied on any solution obtained from other heuristics, such as the ones described in the next sections. These recursive versions are currently being tested.

5.3 Scheduling-centric Heuristic

The main goal of the scheduling-centric heuristic is to minimize setup times. Recall that a setup time is necessary when two jobs of different families are scheduled consecutively on a machine. The heuristic starts by placing first the jobs of the family with the shortest threshold. When adding a job implies that the threshold of another family will be violated on the machine, the heuristic shifts to a job of this other family. Figure 2 shows the mechanism of this heuristic. Jobs of the same family are scheduled consecutively until a time instant where it is no longer possible to schedule more jobs of the same family without losing the qualification of another family on this machine.

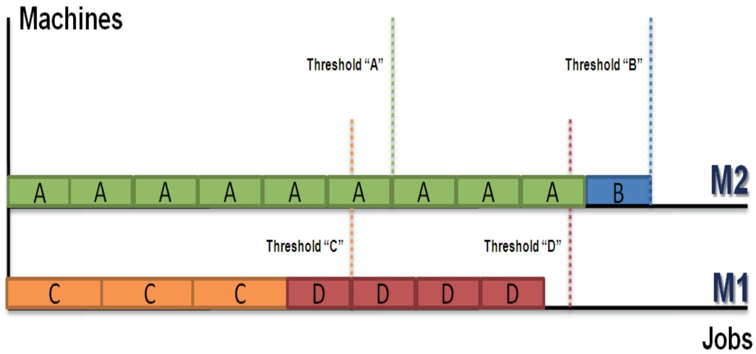

Figure 2: Example of the scheduling-centric heuristic.

5.4 Qualification-centric Heuristic

The main objective of the qualification-centric heuristic is to minimize the total number of violations of the time constraint of each family on the machines on which the family is qualified. To do this, the qualified family which still has a job to schedule and with the shortest remaining time threshold is first selected on a machine. A job in the family is scheduled on the machine, and its completion time is updated. The remaining time threshold of the family is then reset on the machine. If it is no longer possible to schedule a job of a family before its remaining time threshold, then the machine is disqualified for the family.

After scheduling all the available jobs on the qualified machines while focusing on maintaining machine qualifications, we then try to reduce the impact of setup times between job families by scheduling when possible jobs of the same family consecutively. To do this, we consider the last job of the sequence on a machine, and try to schedule this job as early as possible with a job of the same family. We then check whether the machine qualifications are still valid in the resulting schedule. If this is the case, then the change is accepted, otherwise the new schedule is rejected. The process is repeated for all machines and for each last job until no change is possible. Figure 3 illustrates this heuristic. Each time a job is completed, the heuristic checks all the updated thresholds of the families, and chooses a job of the family with the shortest current threshold.

The other phase of this heuristic is to try to shift jobs or groups of jobs of the same family (batch) between machines (inter-machine job exchange). We start by taking the last job/batch of a certain machine and then try to place it on another machine, besides a job of the same family so as to compensate for the setup time.

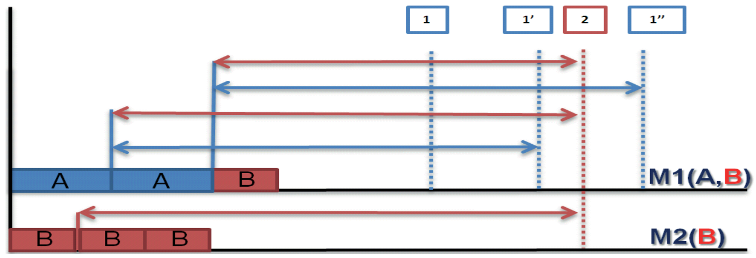

Figure 3: Example of qualification-centric heuristic.

6 COMPUTATIONAL EXPERIMENTS

To test the mathematical programming model and the heuristics, test instances were randomly generated. The different parameter values to generate the instances were chosen so that the basic problem pre-requisites are respected. The time thresholds of job families were set sufficiently large with respect to their associated processing times. This was done to give a minimal bias to find a solution, since short thresholds may lead to a very fast disqualification of the machines. Hence, it might not be possible to process all available jobs, since jobs cannot be sent to disqualified machines. We considered that $Max(p_f) \leq Min(\gamma_f)$. The initial family/machine qualification scheme was defined so that each family has at least one machine on which it can be processed, and each machine is qualified to process at least one job family.

Setup times were not chosen too large so as the risk of disqualifying a machine due to a set-up time insertion is acceptable. We considered that $Max(s_f) \leq Min(p_f)$. In addition, the time horizon was taken as the sum of all processing times, plus the setup time multiplied by the number of jobs per family. This is an extreme case where all jobs are scheduled on one single machine and where, each time a job is scheduled, a setup time is required.

The model was tested using a standard solver (FICO Xpress-MP), on an Intel Xeon processor of 2.50 GHz and 3 GB of RAM. Exact solutions were obtained for several types of instances that were generated with 20, 40, 60 and 80 jobs, 2, 3, 4 and 5 families, and 2, 3, 4 and 5 machines. The heuristics were tested on the same computer. The execution time for computing exact solutions is limited to 600 seconds.

Tables 1 and 2 present the results on various test instances. In the second column, the instance is represented as the number of jobs, the number of families and the number of machines ($|J| - |F| - |M|$). The values of the sum of losses in machine qualifications as well as the sum of completion times is provided for each heuristic. The exact solution obtained for the bicriteria objective function with $\alpha = 1$ and $\beta = |N| * T$ (i.e. $\sum_{f \in F} C_f + |N| * T * \sum_{f \in F} \sum_{m \in M(f)} Y_f^m$) is in the column "Optimum". Taking these values for α and β means that minimizing the number of machine qualification losses is prioritized over the sum of completion times.

Note that, for example in Instance 4, the sum of completion times obtained by the qualification-centric heuristic is better than in the "Optimum" solution for the same instance. However, there is one qualification loss in the solution obtained by the heuristic, and no qualification is lost in the "Optimum" solution. Other tests were conducted with $\alpha = 1$ and $\beta = 1$, where we found that the sum of completion times is smaller than in the previous case. For example, taking again Instance 4, the number of machine qualification losses is increased to 2, while the sum of completion times is minimized and is equal to 206.

We also notice that the gap between the results of the sum of completion times obtained by the heuristics and the exact method is still rather large. However, comparing the heuristics with each other, we notice that the results of the scheduling-centric heuristic are in most cases better than those of other heuristics in terms of the sum of completion times. On the other hand, as expected, the qualification-centric heuristic gives better results in terms of number of machine qualification losses. Moreover, the exact solutions for

Table 1: Heuristics applied on test instances and exact solutions (Instances 1 to 15).

Instance		Recursive		Sched-centric		Qual-centric		Optimum ($\alpha=1, \beta=\lvert N\rvert * T$)	
No.	$\lvert J\rvert-\lvert F\rvert-\lvert M\rvert$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$
1	20-2-3	**1**	901	2	**817**	**1**	850	0	723
2	20-2-4	1	951	1	**861**	**0**	909	0	755
3	20-2-5	1	855	**0**	**782**	**0**	**782**	0	624
4	20-3-2	**1**	224	**1**	217	**1**	**211**	0	319
5	20-3-3	1	741	1	672	**0**	**654**	0	567
6	20-3-4	1	508	**0**	**488**	**0**	**488**	0	388
7	20-3-5	**0**	701	**0**	**617**	**0**	641	0	526
8	20-4-2	**1**	214	**1**	214	**1**	**210**	0	202
9	20-4-3	**1**	237	2	372	**1**	**355**	0	280
10	20-4-4	**0**	525	**0**	**518**	**0**	525	0	439
11	20-4-5	**0**	570	**0**	**425**	**0**	470	0	385
12	20-5-2	**2**	238	**2**	**180**	**2**	238	0	180
13	20-5-3	**0**	**134**	**0**	**134**	**0**	**134**	0	126
14	20-5-4	**0**	174	**0**	**164**	**0**	174	0	142
15	20-5-5	**0**	502	**0**	**483**	**0**	493	0	372
Average		10/15	498	10/15	462	6/15	476	0	402

Table 2: Heuristics applied on test instances and exact solutions (Instances 16 to 30).

Instance		Recursive		Sched-centric		Qual-centric		Optimum ($\alpha=1, \beta=\lvert N\rvert * T$)	
No.	$\lvert J\rvert-\lvert F\rvert-\lvert M\rvert$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$	$\sum Y_f^m$	$\sum C_f$
16	40-4-2	**3**	790	**3**	**710**	**3**	712	0	704
17	40-4-3	**3**	1657	**3**	**1503**	**3**	1513	0	1193
18	40-4-4	3	1418	**0**	**1051**	**0**	1081	0	1052
19	40-4-5	3	1458	4	**1259**	**2**	1321	0	1105
20	60-3-2	**0**	5040	3	**2853**	**0**	2592	0	2229
21	60-3-3	**3**	4604	**3**	4514	**3**	**4350**	0	3865
22	60-3-4	**3**	3031	4	**2571**	**3**	2627	0	2614
23	60-3-5	3	4682	3	4407	**1**	**4399**	0	3141
24	60-4-3	**1**	4824	2	4375	**1**	**4008**	0	3453
25	60-4-4	6	2587	**5**	**2166**	6	2280	0	1766
26	60-4-5	3	3030	**0**	**2769**	1	2786	-	-
27	80-5-2	**3**	3174	**3**	2886	**3**	**2880**	-	-
28	80-5-3	**6**	2752	**6**	2540	**6**	**2118**	0	1819
29	80-5-4	8	3582	8	**2952**	**5**	3150	-	-
30	80-5-5	4	6685	**3**	**5518**	**3**	5743	-	-
Average		52/15	3288	50/15	2805	40/15	2771	0	2086

instances 26, 27, 29 and 30 could not be found by the exact solver in the given time limits (600 seconds).

The CPU time for each heuristic is almost the same since solutions are obtained instantaneously. This is why CPU times are not shown in Tables 1 and 2.

7 CONCLUSION AND PERSPECTIVES

In this paper, we discussed the problem of scheduling job families on non-identical parallel machines with time constraints. We modified a mathematical model and considered a bicriteria objective function: Sum of completion times and number of machine qualification losses. We developed heuristics that target each criterion of the objective function, and numerical results on randomly generated instances were presented. These results showed, as expected, that the scheduling-centric heuristic gives better results regarding the sum of completion times, and that the qualification-centric heuristic provides better solutions on the number of machine qualification losses. The solutions of the heuristics are compared with the exact solutions given by a standard solver.

An extension of this work is the development of more advanced heuristics, in particular metaheuristics. Another perspective is to adapt exact methods in order to find optimal solutions for larger instances than with the mathematical model.

REFERENCES

Allahverdi, A., C. Ng, T. Cheng, and M. Y. Kovalyov. 2008, June. "A Survey of Scheduling Problems with Setup Times or Costs". *European Journal of Operational Research* 187 (3): 985–1032.

Caccamo, M., G. Lipari, and G. Buttazzo. 1999. "Sharing Resources Among Periodic and Aperiodic Tasks with Dynamic Deadlines". In *Real-Time Systems Symposium, 1999. Proceedings. The 20th IEEE*, 284–293.

Detienne, B., S. Dauzère-Pérès, and C. Yugma. 2010. "Scheduling Jobs on Parallel Machines to Minimize a Regular Step Total Cost Function". *Journal of Scheduling*:1–16. 10.1007/s10951-010-0203-z.

Eom, D.-H., H.-J. Shin, I.-H. Kwun, J.-K. Shim, and S.-S. Kim. 2002. "Scheduling Jobs on Parallel Machines with Sequence-Dependent Family Set-up Times". *The International Journal of Advanced Manufacturing Technology* 19:926–932. 10.1007/s001700200105.

Graham, R., E. Lawler, J. Lenstra, and A. Kan. 1979. "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey". In *Discrete Optimization II, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver*, edited by E. J. P.L. Hammer and B. Korte, Volume 5 of *Annals of Discrete Mathematics*, 287 – 326. Elsevier.

Kallapur, P. V., and N. N. Chiplunkar. 2010. "Topology Aware Mobile Agent for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines". In *Advances in Computer Engineering (ACE), 2010 International Conference on*, 352–356.

Kim, Y., J. Kim, S. Lim, and H. Jun. 1998. "Due-date Based Scheduling and Control Policies in a Multiproduct Semiconductor Wafer Fabrication Facility". *Semiconductor Manufacturing, IEEE Transactions on* 11 (1): 155–164.

Kubiak, W., S. X. C. Lou, and Y. Wang. 1996, October. "Mean Flow Time Minimization in Reentrant Job Shops with a Hub". *Operations Research* 44 (5): 764–776.

Kumar, P. R. 1993. "Re-Entrant Lines". *Queueing Systems* 13:87—110.

Li, L., and F. Qiao. 2008. "The Impact of the Qual-run Requirements of APC on the Scheduling Performance in Semiconductor Manufacturing". In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, 242–246.

Mason, S. J., J. W. Fowler, and W. M. Carlyle. 2002. "A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops". *Journal of Scheduling* 5 (3): 247–262.

Moench, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations". *Journal of Scheduling*:1–17. 10.1007/s10951-010-0222-9.

Musacchio, J., S. Rangan, C. Spanos, and K. Poolla. 1997. "On the Utility of Run to Run Control in Semiconductor Manufacturing". In *Semiconductor Manufacturing Conference Proceedings, 1997 IEEE International Symposium on*, D9–12.

Obeid, A., S. Dauzère-Pérès, C. Yugma, and A. Ferreira. 2010. "Scheduling Job Families on Non-Identical Parallel Machines under Run-To-Run Control Constraints". *12th International Workshop on Project Management and Scheduling*:309—313.

Pinedo, M. L. 2009, August. *Planning and Scheduling in Manufacturing and Services*. Springer.

Schaller, J. E., J. N. D. Gupta, and A. J. Vakharia. 2000, September. "Scheduling a Flowline Manufacturing Cell with Sequence Dependent Family Setup Times". *European Journal of Operational Research* 125 (2): 324–339.

Somasundara, A., A. Ramamoorthy, and M. Srivastava. 2007. "Mobile Element Scheduling with Dynamic Deadlines". *IEEE Transactions on Mobile Computing* 6 (4): 395–410.

Webster, S. T. 1997, February. "The Complexity of Scheduling Job Families about a Common Due Date". *Operations Research Letters* 20 (2): 65–74.

AUTHOR BIOGRAPHIES

ALI OBEID received his masters degree in Industrial Control Engineering from the Universite de Technologie de Compiegne in Compiegne, France in 2008. Currently he is a Ph.D. student in Industrial Engineering at the Ecole des Mines de Saint-Etienne in Gardanne. His email address is obeid@emse.fr.

STÉPHANE DAUZÈRE-PÉRÈS is Professor at the Provence Microelectronics Center of the Ecole des Mines de Saint-Etienne, where he is heading the Manufacturing Sciences and Logistics Department. He received the Ph.D. degree from the Paul Sabatier University in Toulouse, France, in 1992; and his Habilitation á Diriger des Recherches from the Pierre and Marie Curie University, Paris, France, in 1998. He was a PostDoc Fellow at the Massachusetts Institute of Technology, U.S.A., in 1992 and 1993, and Research Scientist at Erasmus University Rotterdam, The Netherlands, in 1994. He has been Associate Professor and Professor from 1994 to 2004 at the Ecole des Mines de Nantes in France. He was invited Professor at the Norwegian School of Economics and Business Administration, Bergen, Norway, in 1999. Since March 2004, he is Professor at the Ecole des Mines de Saint-Etienne. His research mostly focuses on optimization in production and logistics, with applications in planning, scheduling, distribution and transportation. He has published more than 35 papers in international journals and 100 communications in conferences. His email address is dauzere-peres@emse.fr.

CLAUDE YUGMA is an Associate Professor at the Provence Microelectronics Center of the Ecole des Mines de Saint-Etienne, in the Manufacturing Sciences and Logistics Department. He received the Ph.D. degree from the Institut National Polytechnique de Grenoble, France, in 2003. His main work focuses on problems related to semiconductor manufacturing sciences as scheduling, Advanced Process Control, etc. His email address is yugma@emse.fr.