# QUALIFICATION MANAGEMENT WITH BATCH SIZE CONSTRAINT

Mehdi Rowshannahad<sup>1,2</sup>

<sup>1</sup>Department of Manufacturing Sciences and Logistics Ecole des Mines de Saint-Etienne – CMP 13541 Gardanne, FRANCE Stéphane Dauzère-Pérès<sup>1</sup>

<sup>2</sup>Soitec Parc Technologique des Fontaines 38190 Bernin, FRANCE

# ABSTRACT

Qualification and batch size constraints are two important characteristics of many toolsets in semiconductor manufacturing. In this paper, we consider the impact of batch size constraint on two flexibility measures used for qualification management. We propose several solution approaches that consider batch sizes when balancing the workload on machines in terms of number of products or process time. Using numerical experiments conducted on real fab data, the solution approaches are compared and the solutions are analyzed. Conclusions are drawn on the impact of batch size on qualification management.

# **1 INTRODUCTION**

The semiconductor manufacturing industry is one of the most complex and dynamic industries in the world. The wide range of constraints, together with expensive manufacturing costs, make an efficient planning of the fabrication facilities (called "fabs") very challenging. In the course of production, each wafer undergoes operations at each workstation (named as "toolset"). A toolset is a collection of parallel machines (or "tools") performing similar operations. For instance, a collection of furnaces, with eventually different specifications (throughput, batch size, hardware and software) constitute the "Thermal Treatment" toolset. Each operation is associated with a *recipe* which is the machine instructions to obtain the desired process. It may specify the temperature value for each ramp up and ramp down phases together with their duration and gas pressure. In order to perform an operation on a machine, its recipe must be *qualified* beforehand on the machine. Due to machine hardware and software restrictions, all operations cannot be performed on all machines. In other words, all recipes cannot be qualified on every machine of the toolset. When a recipe is not authorized to be qualified on a machine, it is called to be unqualifiable for that machine. On the other hand, if it is *authorized* to process a recipe on a machine, but the recipe is not yet qualified on the machine, it is called to be qualifiable. Finally, if a qualifiable recipe is already qualified on a machine, it is called to be *qualified*. The ideal configuration would be that all recipes are qualified on all machines, so that we can freely allocate the production volume of any recipe to any machine. However, in practice, this is impossible due to several reasons. As already stated, because of process restrictions, some of the recipe-to-machine qualifications are not authorized. Qualifying "qualifiable recipes" on machines is time consuming and also costly, causing scheduled downtimes. Therefore, only a few number of qualifications can be performed.

Qualifications have a direct impact on production capacity. If a recipe is not qualified on a machine, it is not possible to allocate the production volume of this recipe to the machine in question. Therefore, a poor qualification configuration causes loss of capacity. This effect is intensified for a bottleneck toolset.

The impact of *qualification management* in semiconductor industry has been pointed out in Johnzén et al. (2007). Several models and approaches have been proposed to find the best qualifications. Aubry, Espinouse, and Jacomino (2006) introduce a branch and bound method to solve a bi-objective optimization model which maximizes the robustness level with a fixed number of extra qualifications. Aubry et al. (2008)

propose a mixed integer linear program to find the qualification configuration at minimum cost in order to balance the workload on the toolset while meeting demand. Ignizio (2009) proposes an optimization model using boolean variables to balance the toolset workload while suggesting new qualifications for a photolithography toolset. Johnzén, Dauzère-Pérès, and Vialletelle (2011) define three *flexibility measures* to propose the best possible qualification(s) to the decision maker. One measure aims at helping the production to hedge against possible machine breakdowns. The two other measures aims at balancing the workload in terms of number of products or in terms of production time. In the following, we describe in more detail these flexibility measures.

Wafers of the same reference travel together in units called *lots*. Each lot consists of normally 25 wafers. Depending upon the machine, lots are grouped to form *batches*. The lots of a batch undergo the same process. Batching reduces the initial setup cost. Therefore it is mostly appropriate for processes with high setup costs, for instance long-run processes such as those in furnaces. Besides, batching assures a uniform processing condition for all lots. In some cases, lots are ungrouped when a machine processes less than 25 wafers at each *production run*. In all the cited qualification management approaches, the batch size has not been considered. For the toolset studied in this paper, the batch size may vary between 4 to 10 lots at each production run (equivalent to 100 to 250 wafers per run)! The wide range of the batch size motivates the study of its impact on qualification management and hence capacity planning. As the machine specifications are different, the batch size of the same recipe may vary depending upon the machine. In view of the fact that the throughput of each machine differs from recipe to recipe, an efficient qualification management considering these constraints optimizes the capacity utilization of the toolset while reducing the variability and the cycle time.

In this paper, we investigate the impact of the batch size on the recipe-to-machine allocation used in qualification management and toolset capacity. In Section 2, two of the flexibility measures defined originally in Johnzén, Dauzère-Pérès, and Vialletelle (2011) are briefly recalled. Then, in Section 3, several solution approaches for these flexibility measures considering batch size constraint are proposed. In Section 4, numerical experiments on real fab data for a "Thermal Treatment" toolset are discussed. Finally, we draw conclusions and mention some research perspectives in Section 5.

### **2 PROBLEM DEFINITION**

In order to determine the best recipe-to-machine configuration, Johnzén, Dauzère-Pérès, and Vialletelle (2011) proposed indicators depending on different criteria. In this paper, we discuss two of these indicators (out of three) which try to balance the workload on the toolset. These indicators, named *flexibility measures*, vary between zero and one (if expressed in percentage, between 0% and 100%). The flexibility measures are used by the decision maker to perform the best qualification(s). The extra qualification which yields the highest flexibility for the entire toolset recipe-to-machine configuration is a better alternative of qualification. In order to calculate the flexibility gain for each qualifiable recipe, at first, flexibility is calculated for the current recipe-to-machine configuration. Then, the flexibility of the entire toolset is calculated by virtually qualifying each qualifiable recipe on each machine while the flexibility value is saved in a matrix. By subtracting the flexibility after each qualification from the flexibility value of the current configuration, we obtain the flexibility gain for each qualifiable recipe-to-machine alternative.

### 2.1 Flexibility Measures

In the following sections, *WIP and Time flexibility measures*, which indicate how well the workload can be balanced on the toolset, are recalled. To balance the workload on the toolset, optimization problems must be solved. The following notations are used in the paper.

#### Parameters

- *R* Total number of recipes to be processed,
- *M* Total number of machines in the toolset,

- $WIP_r$  Total production volume of recipe r,
- $TP_{r,m}$  Throughput rate of recipe r on machine m (number of wafers per hour),
- $BS_{r,m}$  Batch size of recipe r on machine m (number of wafers per production run),
- $Q_{r,m}$  Qualification state of recipe *r* on machine *m*,
  - $=\begin{cases} 1 & \text{if recipe } r \text{ is already qualified on machine } m, \\ 0 & \text{otherwise.} \end{cases}$
  - $\gamma$  Workload balancing exponent ( $\gamma \ge 1$ ).

#### **Decision variables**

$WIP_m$	The total production volume assigned to machine <i>m</i> ,
WIP <sub>r,m</sub>	The production quantity of recipe $r$ assigned to machine $m$ ,
$C_m$	The total amount of production time on machine <i>m</i> .

### 2.1.1 WIP Flexibility Measure

*WIP flexibility*, denoted as  $F^{WIP}$ , tries to balance the workload on the toolset in terms of the production volume or WIP. Here we recall the definition from Johnzén, Dauzère-Pérès, and Vialletelle (2011):

$$F^{WIP} = \frac{(\sum_{m=1}^{M} WIP_m/M)^{\gamma}}{\sum_{m=1}^{M} (WIP_m)^{\gamma}/M} = \frac{M \times (\sum_{m=1}^{M} \sum_{r=1}^{R} WIP_{r,m}/M)^{\gamma}}{\sum_{m=1}^{M} \sum_{r=1}^{R} (WIP_{r,m})^{\gamma}} \in (0,1].$$

 $F^{WIP}$  is a convex function with  $\gamma > 1$ . It reaches 1 when the nominator and the denominator are equal. It means that when the total production volume  $(\sum_{m=1}^{M} WIP_m = \sum_{r=1}^{R} \sum_{m=1}^{M} WIP_{r,m} = \sum_{r=1}^{R} WIP_r)$  is equally distributed over all machines (represented in nominator) or in other words, if the machines are equally loaded (represented in denominator).

### 2.1.2 Time Flexibility Measure

*Time flexibility*, denoted as  $F^{Time}$ , tries to balance the workload on the toolset in terms of production time. The definition from Johnzén, Dauzère-Pérès, and Vialletelle (2011) is:

$$F^{Time} = \frac{C_{ideal}}{\sum_{m=1}^{M} (C_m)^{\gamma}} = \frac{C_{ideal}}{\sum_{m=1}^{M} \left(\sum_{r=1}^{R} \frac{WIP_{r,m}}{TP_{r,m}}\right)^{\gamma}} \in (0,1].$$

The constant  $C_{ideal}$  is the minimum value of  $\sum_{m=1}^{M} C_m$  (total amount of the production times of every machine in the toolset) when all of the qualifiable recipe-to-machine couples are qualified. By maximizing  $F^{Time}$ , as  $C_{ideal}$  is a constant, the denominator, which is the sum of production times ( $\sum_{m=1}^{M} C_m$ ), is minimized.  $F^{Time}$  (a convex function with  $\gamma > 1$ ) can at best reach 1 when the denominator (the sum of the production times) equals  $C_{ideal}$ , i.e. for the best qualification configuration.  $C_{ideal}$  is defined as follows:

$$C_{ideal} = \min \sum_{m=1}^{M} (C_m)^{\gamma}$$
  
with  $Q_{r,m} = 1$   $\forall r \in \{1, ..., R\}, \forall m \in \{1, ..., M\}$ 

#### **Balancing Exponent**

The *balancing exponent* ( $\gamma$ ) is an adjustment factor in  $F^{Time}$  and  $F^{WIP}$ . By increasing  $\gamma$  in  $F^{Time}$ , we decrease the total production time by allocating production volumes to slower machines. Hence by

increasing  $\gamma$  in  $F^{Time}$ , the total process time on the toolset increases while the maximum production time on high-loaded machines decreases. It is worth to notice that by increasing  $\gamma$ , the nominal value of both  $F^{Time}$  and  $F^{WIP}$  also decreases.  $\gamma$  has no impact on workload balancing in terms of WIP ( $F^{WIP}$ ).

#### 2.2 Optimization Model

In this paper, we propose solution approaches for workload balancing for both the WIP and Time flexibility measures. In addition, recipe-to-machine WIP allocations must be an integer multiple of the batch size of that recipe-to-machine, except for at most one recipe-to-machine allocation for each recipe. The latter is because it is not always possible to find batch sizes for recipe r so that their sum is exactly  $WIP_r$ . Let us denote by  $Y_{r,m}$  a binary variable which is equal to 1 if the quantity of recipe r assigned to machine m is not an integer multiple of the batch size and 0 otherwise. Let us also denote by  $WIPC_{r,m}$  the allocation of recipe r to machine m which can be continuous, i.e. not an integer multiple of the batch size.

In the model below, (1a) is the objective function which is either  $F^{WIP}$  or  $F^{Time}$ . Constraint (1b) ensures that all of the production volume of each recipe is allocated. Constraint (1c) stipulates that recipe r is only allocated to machines m that are qualified for r, i.e. such that  $Q_{r,m} = 1$ . Constraint (1d) ensures that the allocation  $WIP_{r,m}$  of recipe r on machine m is an integer multiple of the batch size  $BS_{r,m}$  of recipe r on machine m.  $\frac{WIP_{r,m}}{BS_{r,m}}$  ( $\in \mathbb{N}^+$ ) is the number of production runs of recipe r on machine m. Constraint (1e) and Constraint (1f) ensure that there is only one variable  $WIPC_{r,m}$  which is strictly positive for each recipe r.

$$\max F \tag{1a}$$

Subject to

М

$$\sum_{m=1} (WIP_{r,m} + WIPC_{r,m}) = WIP_r \qquad \forall r \qquad (1b)$$

$$WIP_{r,m} \le Q_{r,m}WIP_r \qquad \forall r,m \qquad (1c)$$
$$WIP_{r,m} \qquad \forall r,m \qquad (1c)$$

$$\frac{\partial T}{\partial S_{r,m}} \in \mathbb{N}^+ \qquad \qquad \forall r,m \qquad (1d)$$

$$\sum_{m=1}^{m} Y_{r,m} \le 1 \qquad \qquad \forall r \qquad (1e)$$

$$WIPC_{r,m} \le Q_{r,m}WIP_rY_{r,m}$$
  $\forall r,m$  (1f)

$$Q_{r,m}, Y_{r,m} \in \{0,1\} \qquad \qquad \forall r,m$$

$$WIP_r, WIP_{r,m}, WIPC_{r,m} \ge 0 \qquad \forall r, m$$

Note that  $\gamma$  takes another importance in  $F^{Time}$  when considering batch sizes. By varying  $\gamma$ , the WIP allocation of recipes to machines may change. Since the batch size of each recipe on each machine may be different, if the workload balancing is done under batch size constraint, significant changes in  $\gamma$  can play an important role on recipe-to-machine qualification configuration.

#### 2.3 Complexity Analysis

Due to lack of space, we cannot detail the analysis but it is possible to show that the problem with  $F^{WIP}$  or  $F^{Time}$  as a criterion is NP-hard in the strong sense. This is because, even if all recipes are qualified on all machines, i.e.  $Q_{r,m} = 1 \ \forall r, m$ , then the problem can be simplified to the scheduling problem on parallel machines  $P||C_{max}$ , which is known to be NP-hard in the strong sense (Garey and Johnson 1979). The transformation is done by noting that  $F^{WIP} = 1$  or  $F^{Time} = 1$  means that  $\sum_{r=1}^{R} WIP_{r,m}$  is the same for each machine m, i.e.  $\sum_{r=1}^{R} WIP_{r,m} = \frac{\sum_{r=1}^{R} WIP_{r,m}}{M}$ . The decision problem "Is there a solution to the qualification problem such that  $F^{WIP} = 1$  or  $F^{Time} = 1$ " can be transformed to solving a related scheduling problem  $P||C_{max}$  and checking that  $C_{max} = \frac{\sum_{r=1}^{R} WIP_{r,m}}{M}$ .

### **3** SOLUTION APPROACHES

After briefly recalling the main ideas of the algorithm proposed in Johnzén, Dauzère-Pérès, and Vialletelle (2011) for real-valued resolution for the WIP flexibility measure  $F^{WIP}$ , we develop several solution approaches that consider the batch size of each recipe on each machine. Concerning the Time flexibility measure  $F^{Time}$ , we do not recall the resolution method for real-valued variables proposed in Johnzén, Dauzère-Pérès, and Vialletelle (2011) due to lack of space. We use one heuristic proposed for  $F^{WIP}$  to obtain recipe-to-machine WIP allocations that satisfy batch sizes from an initial optimal real-valued solution.

## 3.1 Workload Balancing in terms of WIP

In order to find the optimal workload balance on the toolset in terms of the production volume (WIP), we consider two main approaches. The first approach is a modified version of the algorithm proposed in Johnzén, Dauzère-Pérès, and Vialletelle (2011) which solves the problem while considering batch sizes. Secondly, we propose a simple heuristic (which is later adapted to *Time flexibility*) which takes the optimal real-valued solution as an initial solution and then, according to the batch size of each recipe-to-machine couple, generates a solution considering batch size.

### **3.1.1 Real-Valued Resolution**

For balancing the production volume on the toolset, we maximize  $(F^{WIP})$  while distributing the production quantity of each recipe among the qualified machines for that recipe. The problem is formalized in (1) without Constraint (1d). The real-valued resolution proposed in Johnzén, Dauzère-Pérès, and Vialletelle (2011) is composed of two algorithms. The first algorithm constructs a real-valued initial solution and the second algorithm progressively improves the initial solution to obtain either the optimal or a precise-enough near-optimal real-valued solution.

The algorithm used to construct the real-valued initial solution equally distributes the WIP of each recipe on their corresponding qualified machines. This yields a feasible and (almost always) non-optimal solution. The second algorithm takes this initial solution as an input and improves it to find the optimal solution. The full algorithm is described in Johnzén, Dauzère-Pérès, and Vialletelle (2011).

### **3.1.2 Full-Batch Resolution**

To consider batch sizes, two main approaches are proposed. First, we optimize the workload balance while considering batch sizes. Second, we introduce a simple but quite efficient heuristic which takes a real-valued optimal solution and makes it feasible by considering batch sizes.

#### Full-Batch Workload Balancing from Beginning

Here, we modify the algorithm described in Johnzén, Dauzère-Pérès, and Vialletelle (2011) to consider batch sizes. Several strategies are possible which are described along the section. We may consider batch sizes from the beginning, i.e. while constructing the initial feasible solution or only when improving the initial feasible solution, constructed considering or ignoring batch sizes.

**Constructing Full-Batch Initial Solution** For calculating a real-valued initial feasible solution, we equally divide the WIP of each recipe among all of the qualified machines for that recipe. To consider batch sizes, in each loop, we start allocating the WIP by making as many batches as possible, i.e.  $\lfloor \frac{WIP_{r,m}}{BS_{r,m}} \rfloor$ . The new WIP allocation is composed of one or more batches which represent one or more production runs of recipe *r* on machine *m*.

One case that often happens is that the overall production volume of each recipe does not correspond to an integer number of full batches, which is also difficult to consider from the beginning (when production planning is done). This is because the batch size of each recipe depends on the machine and we do not know

exactly in advance how much production volume of which recipe is allocated to which machine considering that the machine statuses and qualification configuration change over time. In this case, even by attributing  $WIP_{r,m}$  considering the batch size of recipe r on machine m ( $BS_{r,m}$ ), the rest of the production volume for each recipe r is not equal to a production run, i.e one full batch equivalent to  $BS_{r,m}$ . The question is to which of the qualified machines of recipe r, the remaining production volume must be assigned. We will allocate this volume to the less loaded machine among all of the qualified machines of recipe r. However it is better that, if possible, production volumes are determined from the beginning to form full batches. This is because processing only one wafer or a full batch of for instance 250 wafers will take the same process time. Besides, if we cannot make full batches, in some toolsets, including Thermal Treatment toolset, *filler wafers* must be used to fill the vacant places in the boats. By making full batches, we eliminate the cost of filler wafers and reduce the non productivity in the fab. Another case that must be considered is when the production quantity of a recipe is smaller than the smallest batch size of any machine qualified for this recipe. In this case, the production quantity of this recipe is produced only on the machine with the smallest total workload ( $WIP_m$ ).

- Step 0. Set recipe index r to 1 (r = 1).
- Step 1. For each machine in the toolset, determine the production volume allocation of recipe r to machines by equally distributing the production volume of recipe r (*WIP<sub>r</sub>*) among all of the qualified machines for r. If for each of the qualified machines for recipe r, i.e. where  $Q_{r,m} = 1$ , the calculated *WIP<sub>r,m</sub>* is more than a full batch (*WIP<sub>r,m</sub>*  $\geq BS_{r,m}$ ) and forms an integer multiple of the batch size (*BS<sub>r,m</sub>*), then attribute the WIP, else attribute only the integer part of the calculated *WIP<sub>r,m</sub>*  $\leq BS_{r,m}$ ]. However, if the calculated *WIP<sub>r,m</sub>* is less than a full batch (*WIP<sub>r,m</sub>*  $\leq BS_{r,m}$ ), then allocate *WIP<sub>r,m</sub>* to the less loaded machine among the machines qualified for recipe r. Step 2. Set r = r + 1, if  $r \leq R$ , then go to step 1.

**Full-Batch Optimization Algorithm** The structure of the algorithm is the same as the one proposed in Johnzén, Dauzère-Pérès, and Vialletelle (2011). The main difference is in the WIP allocations which are done according to batch sizes.

The algorithm begins with an initial solution  $S_0$ , generated by the initial WIP distribution algorithm (considering or ignoring batch sizes). Before beginning the algorithm, we present the following definition.

Let us define by  $\mathcal{LM}_r$ , the set of *loading machines* for recipe *r* to be the set of qualified machine(s) for *r*, with the smallest WIP quantity among all of the qualified machines for *r*, i.e.:

$$m^* \in \mathscr{LM}_r \Rightarrow WIP_{m^*} = \min_{\forall m \ s.t. \ Q_{r,m}=1} \{WIP_m\}$$

- Step 0. Start with initial solution  $S_0$ . The recipe index r and the iteration k are set to 1 (r = 1 and k = 1).
- Step 1. Remove the WIP of recipe *r* allocated initially to machine *m* (*WIP<sub>r,m</sub>*) from machine *m*,  $(WIP_m = WIP_m WIP_{r,m})$ . Then, the total WIP quantities on each machine *m* is sorted in a decreasing order.
- Step 2. Calculate the WIP allocation quantities  $(WIP_{r,m^*})$ , if we had distributed equally the WIP of *r* on the loading machine(s)  $m^*$  in  $\mathcal{LM}_r$  until:
  - Step 2(a). The WIP quantity on one (or more) loading machine(s)  $m^*$  in  $\mathcal{LM}_r$ , (i.e.  $WIP_{r,m^*}$ ) is equal to the WIP quantity of a machine m' qualified for recipe r and not in  $\mathcal{LM}_r$ :

$$WIP_{m^*} = WIP_{m'}$$

Then m' is added to the set of loading machines, i.e.  $\mathcal{LM}_r \equiv \mathcal{LM}_r \cup \{m'\}$ . Now, if the calculated WIP allocation  $(WIP_{r,m^*})$  constitutes at least one full batch, allocate  $WIP_{r,m^*} = \lfloor \frac{WIP_{r,m^*}}{BS_{r,m^*}} \rfloor$ , and go to Step 2. Else allocate  $WIP_{r,m^*}$  and go to step 2. Due to that fact that the production quantity per recipe may have not been determined (from beginning during production planning) to form full-batches, there is some production quantity remainder which does not constitute a full batch. The remainder  $(WIP_r - \sum_{m^*} WIP_{r,m^*})$ 

where  $WIP_{r,m^*} = \lfloor \frac{WIP_{r,m^*}}{BS_{r,m^*}} \rfloor$ ) is added in the last loop to the least-loaded loading machine in the loading machine set.

Step 2(b). If  $r \leq R$ , then go to Step 1.

Step 3. If  $F_k^{WIP} - F_{k-1}^{WIP} > 0$ , then k = k+1, r = 1 and go to step 1.

As the proposed algorithm (as well as the algorithms which follow) are based on progressive improvement of an initial solution, it is not possible to implement the condition  $F_k^{WIP} - F_{k-1}^{WIP} > 0$  of Step 3 in a strict manner. Therefore, the flexibility value of the previous solution  $S_{k-1}$  is compared with the flexibility value of the new solution  $S_k$  by defining an acceptable gap called  $\varepsilon$ .  $\varepsilon$  is a very small value, for example  $1 \times 10^{-20}$ . The algorithm iterates until precise-enough near-optimal solution is reached.

#### **Batch-Feasibility Algorithm**

When taking into account batch sizes, the problem becomes NP-hard in the strong sense for both flexibility measures as discussed in Section 2.3. This is why we propose a heuristic which takes as input the optimal real-valued workload balance, and makes it feasible by considering the batch size of each recipe on each machine. The resolution approach is different from the previous algorithms in the sense that we first choose the machine and then the recipe, whereas in the previous methods, a recipe is first chosen from which WIP values are assigned to its qualified machines. The advantages of our approach are that we already know the optimal solution of the real-valued problem and we have more flexibility in the problem resolution. Furthermore, the real-valued solution serves as an upper bound of the full-batch problem solution.

- Step 0. The optimal solution generated by the real-valued WIP distribution algorithm is the initial solution  $(S_0)$ .
- Step 1. In the whole toolset, find the most loaded machine  $(m^*)$ . From the set of all qualified recipes for this machine to which WIP values have been allocated  $(Q_{r,m^*} = 1 \text{ and } WIP_{r,m^*} > 0)$ , remove the load from the WIP allocation which is nearest (or in a second version: "farthest") to form a full batch. The removed value is equal to  $\{\frac{WIP_{r,m^*}}{BS_{r,m^*}}\}$ . Attribute this value to the less loaded machine which has already a *non-integer* WIP allocation for the same recipe (i.e.  $\{\frac{WIP_{r,m^*}}{BS_{r,m^*}}\} > 0$ ). Note that this last condition (i.e. non-integer WIP allocation) has been added to avoid cycling.
- Step 2. Once all the WIP allocations are redistributed (if necessary), the stop criterion is checked. If  $F_k^{WIP} F_{k-1}^{WIP} > 0$ , then k = k + 1, r = 1 and go to step 1.

Two versions of the algorithm have been implemented. In the first version, at the beginning of Step 1, the preliminary workload on the toolset is ordered decreasingly and then all WIP attributions are done using this order. In the second version, the ordering is redone each time a WIP allocation is performed. The numerical results in this paper are based on the second version which yields better results on average.

#### 3.2 Workload Balancing in terms of Process Time

In order to find the optimal workload balance for the *Time flexibility measure*  $(F^{Time})$  with real-valued variables, an optimization problem must be solved. The method used is called *Active set method* which

is a *Line Search* strategy for the minimization of unconstrained smooth functions (Johnzén 2009). An investigation of the Active set method, even brief, is out of the scope of this paper, therefore we refer the interested reader to textbooks and papers on the subject (e.g., (Nocedal and Wright 2006)).

As with the WIP flexibility, there are several approaches to determine a solution when batch sizes are considered. In the approach presented here, we use the same heuristic (*Batch-Feasibility Algorithm*) as described in Section 3.1.2 for WIP flexibility. *Batch-Feasibility Algorithm* takes the optimal real-valued load balancing solution of the Time flexibility and make it feasible with respect to the batch size constraint (1d).

Other approaches have also been investigated among which is to limit the step length in the Active Set method in order not to violate the constraint (1d). However, as the active set method is not explained in this paper, we present this approach together with the experimental results in another paper.

## **4 NUMERICAL EXPERIMENTS**

25 industrial instances of SOITEC are used for analyzing our solution approaches. The WIP and Time flexibility measures for the current recipe-to-machine qualification configuration and the qualification configuration after the best qualification are presented in Tables 1 and 3.  $\gamma$  is set to 6 in all our experiments. The choice of  $\gamma$  is based on historical experiments. The workload balancing obtained with  $\gamma = 6$  represents the best the actual toolset workload in the fab. As the optimal real-valued solutions for both WIP and Time flexibility measures are available, they are used as upper bounds to evaluate the quality of our algorithms with batch size constraints.

The WIP flexibility measure obtained with each approach and for each instance is presented in Table 1. It is composed of four blocks which show the flexibility measure for the current recipe-to-machine configuration and the flexibility for the new configuration after the best qualification. Note that in all tables blocks and columns are counted from left. The first block of results (second and third columns) shows the optimal real-valued solutions for flexibility measures. The second block (fourth and fifth columns) shows the flexibility measures taking batch sizes into account, with a (non-optimal) real-valued initial solution. The third block (sixth and seventh columns), like the second block, shows the flexibility measures taking batch sizes into account, with a (non-optimal) full-batch initial solution. The fourth and last block (eighth and ninth columns) shows the flexibility measures taking batch sizes into account using the Batch Feasibility Algorithm. Bold values correspond to the best values obtained among the three heuristics for each instance and for either the current configuration or the new configuration.

Although in most cases, the WIP flexibility is very close with or without batch size constraints, the difference is quite large in Instance 12 (from 79.24% down to 64.54% for the current configuration and from 93.88% down to 75.30% for the configuration after the best qualification). This result could be due to the fact that our solution approaches considering batch sizes does not necessarily find the optimal solution. Looking at Table 1, it is also clear that the two full batch approaches (blocks 2 and 3) perform similarly and strongly dominate the Batch Feasibility Approach.

An important question is whether the best qualification proposed when solving the problem with real values is also relevant when solving the problem with batch size constraints. Table 2 compares the WIP flexibility measure for the best qualification proposed by our batch solution approaches with the best qualification when real values are considered but solved with our batch solution approaches.

Table 2 is composed of three blocks. The first block (second, third and fourth columns) corresponds to the results using the Full-Batch Optimization Algorithm with a real-valued initial solution. The second block (fifth, sixth and seventh columns) corresponds to the results using again the Full-Batch Optimization Algorithm, but with an initial solution determined considering batch size constraints. The third and last block (eighth, ninth and tenth columns) give the results using the heuristic Batch Flexibility Algorithm. Each block contains three columns. The first column (called "Best") contains the WIP flexibility measure value for the recipe-to-machine configuration after the best (new) qualification using its corresponding block solution approach (for instance, the first block solution approach is Full-Batch Optimization Algorithm

Instance number	Real-value		Full-batch (initial: real-value)		Full-batch (initial: full-batch)		Batch-feasibility	
	Current	Best	Current	Best	Current	Best	Current	Best
1	74.43%	87.69%	74.22%	86.53%	74.22%	86.53%	72.20%	85.46%
2	53.88%	67.55%	53.70%	67.22%	53.70%	67.22%	50.47%	65.86%
3	68.83%	78.13%	68.54%	77.61%	68.54%	77.61%	66.60%	76.37%
4	53.49%	70.88%	53.29%	70.66%	53.29%	70.66%	52.86%	69.81%
5	5.08%	33.01%	5.07%	31.46%	5.07%	31.46%	5.07%	5.09%
6	72.73%	98.33%	72.28%	96.96%	72.28%	96.96%	67.99%	93.69%
7	7.25%	14.78%	7.04%	14.23%	7.04%	14.23%	6.81%	14.16%
8	11.01%	20.98%	10.56%	20.41%	10.56%	20.41%	10.32%	19.55%
9	97.17%	97.17%	90.33%	91.72%	90.33%	91.72%	45.97%	64.39%
10	61.76%	62.18%	56.58%	60.88%	56.58%	60.88%	57.44%	59.87%
11	29.09%	33.83%	28.88%	32.87%	28.88%	32.87%	27.09%	31.63%
12	79.24%	93.88%	64.54%	75.30%	64.54%	75.30%	40.24%	75.11%
13	25.30%	53.03%	24.41%	49.36%	24.41%	49.36%	22.96%	45.80%
14	49.96%	49.96%	47.69%	49.01%	47.69%	49.01%	45.87%	47.28%
15	32.99%	36.86%	32.08%	33.05%	32.08%	33.05%	20.22%	33.61%
16	44.81%	69.53%	39.87%	63.63%	39.87%	63.63%	32.87%	46.71%
17	28.19%	47.20%	24.73%	44.62%	24.73%	44.62%	22.11%	29.28%
18	31.11%	38.50%	27.01%	37.32%	27.01%	37.32%	15.84%	26.75%
19	15.01%	24.72%	14.77%	23.75%	14.77%	23.75%	13.41%	16.56%
20	20.17%	37.41%	19.22%	35.13%	19.22%	35.13%	17.46%	32.51%
21	19.62%	19.78%	16.95%	18.76%	16.95%	18.76%	18.33%	19.14%
22	17.83%	21.86%	17.37%	21.34%	17.37%	21.34%	16.57%	21.35%
23	32.80%	33.16%	31.55%	32.23%	31.55%	32.23%	25.69%	28.69%
24	30.24%	32.63%	26.05%	31.22%	26.05%	31.22%	27.16%	30.05%
25	46.75%	68.86%	45.56%	67.67%	45.56%	67.67%	43.73%	66.40%

Table 1: Comparing solution approaches for WIP flexibility measure.

using real-valued initial solution). The second column (called "Best (RV)") is the WIP flexibility measure value for the recipe-to-machine configuration after the best (new) qualification proposed when using the (optimal) real-valued resolution approach. The third and last column of each block shows whether the next (new) qualification of the block's approach is the same as the (new) qualification proposition using the (optimal) real-valued resolution approach. It is interesting to note that there are various instances for which the proposed qualifications differ and which lead to substantial differences of WIP flexibility value, such as Instances 5 (31.46% and 0.70%), 9 (91.72% and 83.59%) and 12 (75.30% and 69.25%). This illustrates that not considering batch sizes may induce less effective qualification decisions. It is interesting to note that the Full-Batch Optimization Algorithm is quite efficient, considering the gap with the results of the real-valued resolution. Besides the algorithm yields the same results regardless of the initial solution (ignoring or considering batch size for the WIP flexibility measure, the Full-Batch Optimization Algorithm is the most efficient.

Table 3 has the same format as of Table 1 for the two first blocks and as of Table 2 for the third (the last) block. It presents the results obtained with the only solution approach studied in this paper for workload balancing with batch sizes for the Time flexibility measure, i.e. the Batch Feasibility Algorithm. According

Instance	Full batch (initial: real value)			Full batch (initial: full batch)			Batch feasibility		
number	Best	Best (RV)	Match	Best	Best (RV)	Match	Best	Best (RV)	Match
1	86.53%	86.53%	*	86.53%	86.53%	*	85.46%	85.46%	*
2	67.22%	67.22%	*	67.22%	67.22%	*	65.86%	65.86%	*
3	77.61%	77.61%	*	77.61%	77.61%	*	76.37%	74.56%	
4	70.66%	70.66%	*	70.66%	70.66%	*	69.81%	69.81%	*
5	31.46%	0.70%		31.46%	0.70%		5.09%	3.41%	
6	96.96%	96.96%	*	96.96%	96.96%	*	93.69%	93.69%	*
7	14.23%	14.07%		14.23%	14.07%		14.16%	14.09%	
8	20.41%	20.41%	*	20.41%	20.41%	*	19.55%	19.55%	*
9	91.72%	83.59%		91.72%	83.59%		64.39%	46.92%	
10	60.88%	60.88%	*	60.88%	60.88%	*	59.87%	59.72%	
11	32.87%	32.87%	*	32.87%	32.87%	*	31.63%	31.63%	*
12	75.30%	69.25%		75.30%	69.25%		75.11%	66.39%	
13	49.36%	49.36%	*	49.36%	49.36%	*	45.80%	45.80%	*
14	49.01%	47.31%		49.01%	47.31%		47.28%	46.83%	
15	33.05%	33.05%	*	33.05%	33.05%	*	33.61%	33.61%	*
16	63.63%	63.63%	*	63.63%	63.63%	*	46.71%	46.71%	
17	44.62%	44.62%	*	44.62%	44.62%	*	29.28%	29.28%	*
18	37.32%	37.32%	*	37.32%	37.32%	*	26.75%	21.62%	*
19	23.75%	23.75%	*	23.75%	23.75%	*	16.56%	13.80%	
20	35.13%	35.13%	*	35.13%	35.13%	*	32.51%	29.29%	
21	18.76%	18.13%		18.76%	18.13%		19.14%	18.42%	
22	21.34%	21.34%	*	21.34%	21.34%	*	21.35%	21.07%	
23	32.23%	32.23%	*	32.23%	32.23%	*	28.69%	26.79%	
24	31.22%	31.22%	*	31.22%	31.22%	*	30.05%	24.75%	
25	67.67%	67.54%		67.67%	67.54%		66.40%	64.56%	

Table 2: Comparing best qualifications for WIP flexibility measure.

\*: Proposed qualifications are identical.

to the first two blocks of Table 3 (second, third, fourth and fifth columns), in many more instances than for the WIP flexibility measure, the differences can be quite large between the Time flexibility measures obtained with real values and with batch sizes, such as in Instance 11 (from 99.22% down to 43.64% for the current configuration and from 99.55% down to 49.50% for the configuration after the best qualification). Referring to the third and last block of Table 3 (columns six and seven), again in more instances than for the WIP flexibility measure, the differences can be quite large between the Time flexibility measure of the best qualification with batch sizes and the Time flexibility measure obtained by running the Batch Feasibility Algorithm for the best qualification with real values, such as in Instance 13 (44.68% and 15.85%). It must be noted that the large gaps could also be explained by the fact that our Batch Feasibility Algorithm is not always very efficient. Other more efficient approaches are being investigated.

## **5** CONCLUSION AND FUTURE RESEARCH

Batching is one of the main constraints in the semiconductor manufacturing industry. In this paper, we have proposed different workload balancing algorithms (in terms of both production quantities and production time) considering batch sizes. The workload allocations are used in qualification management. Using real

Instance	Real value		Batch fe	easibility	Batch feasibility		
number	Current	Best	Current	Best	Best (RV)	Match	
1	40.91%	72.68%	40.43%	66.89%	66.89%	*	
2	34.29%	49.75%	32.83%	45.60%	45.60%	*	
3	85.53%	93.43%	82.92%	88.27%	88.27%	*	
4	33.89%	38.44%	32.94%	36.84%	36.84%	*	
5	34.42%	63.71%	33.44%	61.90%	61.90%	*	
6	82.58%	94.82%	80.03%	92.72%	92.72%	*	
7	98.00%	99.99%	62.11%	81.03%	78.64%		
8	29.34%	41.40%	13.37%	38.72%	25.56%		
9	28.02%	49.44%	17.36%	34.27%	19.42%		
10	67.82%	85.11%	20.89%	49.74%	49.74%	*	
11	99.22%	99.55%	43.64%	49.50%	49.19%		
12	81.14%	96.45%	49.58%	57.28%	31.66%		
13	48.91%	68.81%	45.06%	44.68%	15.85%		
14	70.41%	97.50%	29.66%	47.30%	47.30%	*	
15	99.64%	99.96%	37.27%	58.74%	58.74%	*	
16	74.42%	88.54%	27.43%	27.72%	19.60%		
17	61.70%	76.55%	13.68%	32.44%	25.99%		
18	75.33%	92.15%	38.60%	52.36%	52.36%	*	
19	74.70%	83.46%	26.95%	51.60%	26.86%		
20	69.94%	84.04%	54.98%	61.85%	47.13%		
21	71.49%	85.33%	13.32%	41.37%	26.21%		
22	95.20%	98.17%	45.70%	59.68%	42.71%		
23	97.70%	99.13%	42.96%	78.09%	67.44%		
24	96.03%	98.01%	58.23%	68.45%	43.24%		
25	92.74%	95.00%	26.52%	55.44%	17.04%		

Table 3: Comparing solution approaches Comparing best qualifications for the Time flexibility measure.

\*: Proposed qualifications are identical.

fab data, we have demonstrated that ignoring the batch size constraint may lead to inappropriate qualification decisions which in turn leads to a non-optimal toolset capacity utilization.

Having the near-optimal recipe-to-machine WIP allocations considering the batch size, it may be possible to use the workload balancing tables as an input for a scheduler or a dispatcher of lots.

Various perspectives are possible. The Batch-Feasibility Algorithm might be improved by taking the flexibility measure value (objective function value) into account. This is achievable by adding another condition while re-allocating the workload in step one. This means that we calculate the flexibility value if we have had distributed the (removed) WIP for the machines of the same recipe having a non-integer WIP value. And if there are several alternatives, the WIP allocation is chosen which yields the largest flexibility measure.

Another more general perspective for qualification management is to propose flexibility measures which consider the yield for each recipe on each machine and propose relevant qualifications taking this constraint into account. Another aspect which may be considered directly or indirectly in flexibility measures, is the recipe priority. This means that the model gives more priority to the recipe related to more important products or "hot lots". Auxiliary resources management (for example in lithography toolset) is also a subject related to qualification management which can enrich future studies.

## ACKNOWLEDGMENTS

This study has been done within the framework of a joint collaboration between SOITEC (Bernin, France), and the Center of Microelectronics in Provence of the École des Mines de St-Étienne (Gardanne, France). The authors are grateful to Bernard Cassini (SOITEC) for his support and would like to thank the ANRT (*Association Nationale de la Recherche et de la Technologie*) which has partially financed this study.

## REFERENCES

- Aubry, A., M.-I. Espinouse, and M. Jacomino. 2006. "Robust Load-Balanced Configuration with Fixed Costs for the Parallel Multi-Purpose Machines Problem". 2006 International Conference on Service Systems and Service Management: 990–995.
- Aubry, A., A. Rossi, M.-L. Espinouse, and M. Jacomino. 2008. "Minimizing Setup Costs for Parallel Multi-Purpose Machines under Load-Balancing Constraint". *European Journal of Operational Research* 187 (3): 1115–1125.
- Garey, M. R., and D. S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman.
- Ignizio, J. P. 2009. "Cycle Time Reduction via Machine-to-Operation Qualification". *International Journal* of Production Research 47 (24): 6899–6906.
- Johnzén, C. 2009. Modeling and Optimizing Flexible Capacity Allocation in Semiconductor Manufacturing. Ph. D. thesis, Department of Manufacturing Sciences and Logistics - Center of Microelectronics in Provence, École des Mines de St-Étienne.

Johnzén, C., S. Dauzère-Pérès, and P. Vialletelle. 2011. "Flexibility Measures for Qualification Management in Wafer Fabs". *Production Planning & Control* 22 (1): 81–90.

Johnzén, C., S. Dauzère-Pérès, P. Vialletelle, and C. Yugma. 2007. "Importance of Qualification Management for Wafer Fabs". In *Conference*, 2007. ASMC 2007. IEEE/SEMI, 166–169.

Nocedal, J., and S. J. Wright. 2006. Numerical Optimization. 2nd ed. Springer.

## **AUTHOR BIOGRAPHIES**

**MEHDI ROWSHANNAHAD** holds an MSc in "Science and Technology of Management" from the University of Montpellier II in France, and an MSc in "Industrial Engineering" from the Grenoble National Polytechnics Institute. He is currently working at Soitec (Bernin, France) and is a PhD student at the Center of Microelectronics in Provence of the École des Mines de St-Étienne. His research interests are optimization, capacity and production planning, and scheduling. His email address is rowshannahad@emse.fr.

**STÉPHANE DAUZÈRE-PÉRÈS** is Professor and Director of the Center of Microelectronics in Provence of the EMSE. He received the Ph.D. degree from the Paul Sabatier University in Toulouse, France, in 1992; and the H.D.R. from the Pierre and Marie Curie University, Paris, France, in 1998. He was a Postdoctoral Fellow at the M.I.T., 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 École des Mines de Nantes in France. Since March 2004, he is Professor at the École des Mines de Saint-Étienne. His research interests broadly include modeling and optimization of operations at various decision levels in manufacturing and logistics, with a special emphasis on semiconductor manufacturing. He has published more than 45 papers in international journals and has contributed to more than 100 communications in national and international conferences. He has coordinated multiple academic and industrial research projects, and also five conferences. His email address is dauzere-peres@emse.fr.