A Batching and Scheduling Algorithm for the Diffusion Area in Semiconductor Manufacturing

Claude Yugma, Stéphane Dauzère-Pérès, Christian Artigues, Olivier Sibille

A Batching and Scheduling Algorithm for the Diffusion Area in Semiconductor Manufacturing

Claude Yugma[1] * Stéphane Dauzère-Pérès[1] Christian Artigues[2]
Alexandre Derreumaux[1] Olivier Sibille[3]

[1]École des Mines de Saint-Etienne, Centre Microlectronique de Provence - Site Georges Charpak, 880, Avenue de Mimet, F-13541 Gardanne, France

[2]LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse, France

[3]ATMEL, Zone industrielle, 13790 Rousset, France

Abstract

This paper proposes an efficient heuristic algorithm for solving a complex batching and scheduling problem in a diffusion area of a semiconductor plant. Diffusion is frequently bottleneck in the plant and also one of the most complex areas in terms of number of machines, constraints to satisfy and the large number of lots to manage.

The purpose of this study is to investigate an approach to group lots in batches and to schedule these batches on machines. The problem is modeled through a disjunctive graph formulation. A constructive algorithm is proposed and improvement procedures based on iterative sampling and Simulated Annealing are developed. Computational experiments, carried out on actual industrial problem instances, show the ability of the iterative sampling to significantly improve the initial solution. The Simulated Annealing enhances the results of the iterative sampling. The constructive algorithm has been embedded in a software and is currently being used.

Keywords: Batch, Scheduling, Disjunctive graph, Local search, Simulated Annealing, Wafer fabrication.

1 Introduction

Semiconductor wafer fabrication can be described as a multistage process with re-entrant flows. The processing is done layer by layer. Each layer requires several steps of processing such as chemical-mechanical polishing, diffusion, film deposition, photolithography, implant (doping) and etching. For each of the product types, and depending on the technology, a wafer goes through more than 400 process steps over a period of a few weeks. Wafer fabrication planning and scheduling is a complex task due to the large number of products and machines involved. It is further complicated by additional constraints such as re-entrant flow of operations Kumar [Kum94], setup issues, preventive maintenances, random machine breakdowns, Ovacik and Uzsoy [OU97], Sze [Sze01], etc. The importance of scheduling on the performance of semiconductor wafer fabrication facilities (fabs) is known for many years see Wein [Wei88] and Sze [VS06].

In this paper, we focus on an important part of the manufacturing process. Among the complex operations involved in the fabrication of a wafer, the diffusion phase is of critical importance since the batching decisions that are involved may affect the performance of the entire wafer fab Ibrahim et al. [ICN+03] and Mönch and Habenicht [MH03]. The processing time of the operations in the diffusion area can be large (10h) compared to other operations in the fab. Mehta and Uzsoy [MU98] state that optimizing batching operations results in good performance measures of the whole production process. Lots regularly arrive in the diffusion area and the diffusion phase is primarily used to alter the type and

*Corresponding author e-mail: Claude.Yugma@emse.fr

level of conductivity of semiconductor materials.
The purpose of this article is to develop efficient methods to partition lots in batches and to schedule batches on machines in the diffusion area of a semiconductor plant while taking into account numerous constraints and optimizing three main production criteria: maximize the number of operations (moves), maximize the batch sizes and minimizing the total tardiness.

The remaining sections of this paper are organized as follows. In Section 2, we provide some background on existing related batch and scheduling problems. In Section 3, the problem is stated. We present in Section 4, a disjunctive graph representation of the problem. The method for computing an initial solution and improvement procedures based on iterative sampling and Simulated Annealing are described in Section 5. Experimental results on real problem instances and comparison with a literature algorithm are given and discussed in Section 6. Section 7 concludes the paper with recommendations for further research.

2 Previous related work

The operations of batching and scheduling jobs is a common practice in a manufacturing system especially in semiconductor manufacturing systems, see Mathirajan et al. [MS06a]. For examples reasons for batching are the reduction of setups, the ability of machines to process several jobs simultaneously, etc. With the classification given by authors by Mathirajan et al. [MS06a], we notice that there is not much literature on scheduling batch problems taking into account the re-entrance features of the system as for example we can name Cigolini et al. [CPPZ02], Mason and Oey [MO03] [OM01]. The work proposed by Mönch et al. [MFDP$^+$09] presents a scheduling problems in semiconductor manufacturing. Typical scheduling problems found in semiconductor manufacturing are identified in particular batching problems.
The problem addressed in this paper deals with:

- Multiple machines at each stage. The authors in Mehta and Uzsoy [MU98] presented a problem of minimizing total tardiness on a batch processing machine with incompatible job families. They proposed a dynamic programming to solve the problem. The paper of Balasubramanian et al. [BMFP04] extends solutions of Mehta and Uzsoy [MU98] to a batching problem with incompatible jobs on parallel machines which aims at minimizing weighted total tardiness. The authors in Kim et al. [KJC10] focused on a problem of scheduling wafer lots on diffusion workstations in a semiconductor wafer fabrication facility. In their considered diffusion workstation, there are multiple identical machines, and each of them can process a limited number of wafer lots at a time. Their scheduling problem at the diffusion workstation is a problem of scheduling multiple job families on identical parallel batch-processing machines.

- Multiple stages. The described problem for the diffusion area takes into account all the different stages of this area which can reach 4. In the literature, the number of stages does not generally exceed 2. The author Su [Su03] considered a hybrid two-stage flow shop with a batch processor in stage 1 and a single processor in stage 2. This is also the case in articles Sun and Min [SM01] and Sung and Kim [SK03] which consider a batch problem on 2 stages. Oulamara et al. [OFKK09] studied a two-machine flow shop problem with conventional and batching machines in the first and second stage, respectively, and arbitrary job compatibilities.

- Multiple criteria. The goal of the paper is to simultaneously optimize three indicators. These indicators are the number of lots going through the line (to maximize), the number of wafer in a batch (to maximize) and the waiting time of lots (to minimize). Generally, articles in the literature tackled scheduling batch problems by considering one single indicator to optimize. Uzsoy in [Uzs95] tackled the problem with the objective of minimizing the completion time of jobs, Hung [Hun98] has an objective of maximizing the batch processing machine utilization. Perez et al. [PFC05] studied the problem of minimizing the total weighted tardiness on a single machine and Mathirajan and Sivakumar [MS06b] focused on the minimization of the total weighted tardiness on heterogeneous batch processing machines under condition of dynamic arrival of jobs, incompatible job families and non-identical job sizes. Few articles deal with different criteria simultaneously. Pfund et al. [PBF$^+$08] adapted the Shifting Bottleneck Heuristic to facilitate the multi-criteria optimization of Makespan,

cycle time and total weighted tardiness using a desirability function.

Furthermore, in our problem, setup times (loading and unloading) are not depending on the sequence and the complexity is added by the maximum times lags between batching operations.

Our problem is a complex variant of job shop problem. These types of problems can be found in Mason et al. [MFC02] and Ovacik and Uzsoy [OU97]. These problems are frequently analyzed using a method known as the shifting bottleneck (SB) procedure see for example Adams et al. [ABZ88]. Several aspects, like identifying appropriate sub problem solution procedures can be found in Demirkol and Uzsoy [DU00] and Uzsoy and Wang [UW00]. In Mason et al. [MFC02], a scheduling problem in semiconductor manufacturing close to the one tackled in this paper is solved by a modified SB heuristic. We propose to model solutions of the considered batching and scheduling problem through a variant of the disjunctive graph proposed in Mason et al. [MFC02].

3 Problem description

The diffusion area defines a batching and scheduling problem of wafer lots on two types of equipment: cleaning and furnaces machines. These resources are able to perform several lots simultaneously. Each lot requires one or more consecutive operations on the equipment and each operation has a recipe (specification on a process on how it should be executed on a tool; This pertains to requirements of maintaining proper temperature, pressure, and metal composition, among others) which determines its duration and the set of machines that are able to process it. On a 24-hour basis, each operation has to be assigned to an equipment, and included into a batch, i.e. a set of operations of the same recipe that are processed simultaneously by the equipment. The constraints in the diffusion area are divided into three types: equipment constraints, process constraints and line management constraints.

3.1 Constraints

Some of these constraints are common to the two types of resources while others are dedicated to furnaces.

3.1.1 Equipment constraints

- **Common constraints**
  Dedicated equipment: Any equipment is able to process a limited set of recipes.
  Maximum batch size: Any equipment defines a maximum batch size corresponding to the capacity of that equipment.
  Loading and unloading times: A time may be needed to load and unload a batch on the equipment.
  Unavailability periods: The equipment may be unavailable during some periods (defined by time windows) due to qualification, repair, maintenance etc.
  In process jobs: The equipment may be occupied at the beginning of the time horizon by in-process operations that have to be completed before the equipment becomes available.

- **Specific furnace constraints**
  Minimum time between two batches on an furnace: Furnaces must be inspected after completing each batch.

3.1.2 Process constraints

- **Common constraints**
  Precedence: Operations must be performed following the manufacturing process of the lot. The operations of a lot are chained and no operation can start before the end of its predecessor, except for the first operation of each lot.
  Minimum time lag: There is a fixed handling and transport time between every two successive operations of a lot.
  Release dates: They correspond to the arrival times of lots at the cleaning machines and furnaces. A

lot cannot be scheduled before its release date. Because the diffusion area is a stage of the complex global production process, release dates are estimated by a simulation tool.
Fixed recipe: Each operation of a lot is associated with a recipe, i.e. the lot should be processed on resources that are qualified for the corresponding recipe. This implies that all lots in the same batch must be processed with the same recipe.
Process time: The process time of a batch on an equipment depends on the recipe.
Maximum time lag: A time limit is given for two successive operations x and y of a lot. The difference between the starting time of y and the completion time of x cannot exceed this limit. Maximum time lags depend on the operations x and y. They can correspond to the maximum time limit between cleaning and furnace equipment.

3.1.3 Line management constraints

- **Specific furnace constraints**
  Minimum time between two batches of same recipe: There is a minimum time between the beginning of two batches of the same recipe on two different furnaces.

3.2 Objective

In semiconductor fabs, several indicators are used to measure the performance. The interested reader can refer to [MT05] for more details. We describe thereafter the indicators that are relevant in our study.

- *Number of moves* (to maximize): It corresponds to the number of completed operations on the planning horizon, which can be compared to the target number fixed by the production managers.
- *Batching coefficient* (to maximize): Defined on the planning horizon, it is calculated as the number of moves divided by the sum of the number of batches performed on each machine, times the maximum capacity of that machine. Note that the denominator is the number of lots that could be performed if the equipment was loaded up to their maximum capacities.
- *X-factor* (to minimize): This indicator is used to evaluate the waiting times of lots in the diffusion area in order to reduce the cycle times. For a given lot, this factor is calculated as the total staying time of the lot in the diffusion area over its processing time.

These three indicators will be used to evaluate solutions in the fab.

Other indicators could be relevant. Among them, the *Work-In-Process* (WIP), is defined as the number of wafers being in the fab at a given period, either in a production state or in a non-production state (e.g. transport and waiting). The law of Little [Lit61] establishes that if a system is stable and stationary then the average WIP is proportional to the average cycle time. The *throughput* is defined as the outgoing number of wafers of the fab per unit of time. The evolution of the throughput in time makes it possible to know if the system is stable i.e. to know if there is no accumulation of lots in the fab or if the system evolves according to forecasts. Glassey and Resende [GR88], observed that there is a relation between the increasing of the throughput and the output of a fab. The cycle time drastically increases when throughput is close to the maximal capacity of the fab. Hence for both WIP and throughput indicators, the X-factor indicator is adequate.

The goal is to optimize the various performance measures, while taking into account the numerous complex constraints.
The next subsection described the mathematical formulation of the problem.

3.3 Problem formulation

The scheduling problem can be formulated as follows. For sake of clarity, the above-described *Unavailability periods*, *In-process jobs* and *Minimum distance between batches of the same recipe* constraints are not included. In Section 4, we describe how we tackle these characteristics.

A set of jobs (lots) $\mathcal{J} = \{J_i | i = 1, \ldots, n\}$ has to be processed on a horizon T by a set of machines $\mathcal{M} = \{M_k | k = 1, \ldots, m\}$. Each job J_i is made of n_i operations such that each operation O_{ij} has a duration $p_{ij} > 0$ and a set $\mathcal{M}_{ij} \subseteq \mathcal{M}$ of machines (the furnaces or the cleaning machines) able to process it. Let $\mathcal{O}_k = \{O_{ij} \in \mathcal{O} | M_k \in \mathcal{M}_{ij}\}$ denote the set of operations that can be assigned to machine M_k. The

value of p_{ij} and the elements of the set $\mathcal{M}_{ij}$ are determined by the recipe of operation O_{ij} denoted ρ_{ij}. In general we have $\mathcal{M}_{ij} \subset \mathcal{M}$ since each machine cannot be configured for all recipes. Each operation O_{ij} has to be included in a batch on a resource $k \in \mathcal{M}_{ij}$. Each machine has a finite capacity R_k which gives the maximal number of lots in the same batch. On each machine k, S_k denote the setup time needed before starting a new batch, D_k denote the removal time needed after the completion of a batch and s_k denote the constant setup time needed between two different batches. s_k^0 denotes the initial setup time on machine k, depending on the state of the resource at time 0. Two consecutive operations O_{ij} and $O_{i(j+1)}$ of the same job are linked by minimal and maximal time lags. Once the batch of O_{ij} is completed and removed from k, the setup for the batch of $O_{i(j+1)}$ cannot start before a minimal time lag $\tau_{ij}^{\min}$ and has to start before a maximal time lag $\tau_{ij}^{\max}$. Let $\mathcal{O} = \{O_{ij} | i = 1, \ldots, n; j = 1, \ldots, n_i\}$ denote the set of all operations. Each job J_i has a relative priority c_i ($c_i < c_j$ means that J_i is more urgent than J_j). Each job corresponds to a number w_i of wafers produced when the job is completed.

Table 1 summarizes the notations used.

Notations	Description
$\mathcal{J} = \{J_i \| i = 1, \ldots, n\}$	Set of jobs (lots)
T	Horizon length
$\mathcal{M} = \{M_k \| k = 1, \ldots, m\}$	Set of machines
n_i	Number of operations of lot J_i
$\mathcal{O} = \{O_{ij} \| i = 1, \ldots n; j = 1, \ldots n_i\}$	Set of operations
O_{ij}	Operation j of lot J_i
p_{ij}	Duration of operation O_{ij}
$\mathcal{M}_{ij}$	Qualified machines to process O_{ij}
$\mathcal{O}_k$	Operations that can be processed on the machine M_k
ρ_{ij}	Recipe of O_{ij}
R_k	Capacity of machine M_k (batch max. size)
S_k	Setup before the operation on machine M_k
D_k	Setup after the operation on machine M_k
s_k	Delay inter batch
s_k^0	Initial setup time
$\tau_{ij}^{\min}$	Min. delay between O_{ij} and O_{ij+1}
$\tau_{ij}^{\max}$	Max delay between O_{ij} and O_{ij+1}
c_i	Priority of lot J_i
w_i	Number of wafers of J_i
$\mathcal{B} = \{B_{kq}\}_{k \in \{1,\ldots,m\}, q \in \{1,\ldots,\nu_k\}}$	Batch at position q on machine M_k
ν_k	Number of batches on machine M_k
$\mathcal{T} = \{t_{ij}\}_{O_{ij} \in \mathcal{O}}$	Set of start times
m_{ij}	Machine which processes O_{ij}
θ_i	Completion ratio of O_{ij}
$\mathcal{B}^T$	Batches started before T
Z_k	Number of qualified recipes on machine M_k
θ^T	Operations started before T
$N = \sum_{i=1}^{n} n_i$	Total number of operations

Table 1: Summary of notations used to formalize the problem

Finding a feasible solution for the problem lies in making four types of decisions:

D1 - Partition the operations into batches,

D2 - Select a resource to process each batch,

D3 - Order the batches on each resource,

D4 - And assign a start time to each batch.

Decisions D1-D3 can all be represented by a family of batches $\mathcal{B} = \{B_{kq}\}_{k \in \{1,\ldots,m\}, q \in \{1,\ldots,\nu_k\}}$ where B_{kq} is the batch sequenced at position q on machine M_k. $\nu_k \in \{0, \ldots, |\mathcal{O}_k|\}$ denote the number of batches assigned to machine M_k. Decision D4 lead to a family of start times $\mathcal{T} = \{t_{ij}\}_{O_{ij} \in \mathcal{O}}$ assigned to the operations. Once a solution $\{\mathcal{B}, \mathcal{T}\}$ is determined, we have a machine assignment $\{m_{ij}\}_{O_{ij} \in \mathcal{O}}$ where m_{ij} denote the machine O_{ij} is assigned to, i.e. verifying that $\exists q \in \{1, \ldots, \nu_k\}$ such that $O_{ij} \in B_{m_{ij}q}$. To be feasible, a solution $\{\mathcal{B}, \mathcal{T}\}$ and its corresponding assignment $\{m_{ij}\}_{O_{ij} \in \mathcal{O}}$ have to satisfy the following constraints. The operations of the same batch must have the same recipe, i.e.

$$\rho_{ij} = \rho_{xy} \quad \forall B \in \mathcal{B}, \forall O_{ij}, O_{xy} \in B \tag{1}$$

Each operation must be assigned to a machine able to process its recipe.

$$m_{ij} \in \mathcal{M}_{ij} \quad \forall O_{ij} \in \mathcal{O} \tag{2}$$

The batch capacity cannot be exceeded and each batch includes at least one operation.

$$1 \leq |B_{kq}| \leq R_k \quad \forall B_{qk} \in \mathcal{B} \tag{3}$$

An operation appears in only one batch, i.e.

$$B \cap B' = \emptyset \quad B, B' \in \mathcal{B}, B \neq B' \tag{4}$$

All operations are included in a batch

$$\cup_{B \in \mathcal{B}} B = \mathcal{O} \tag{5}$$

The start time of the first operation of each lot cannot exceed the lot release date:

$$t_{i1} \geq r_i \quad \forall J_i \in \mathcal{J} \tag{6}$$

Each operation O_{ij}, $j > 1$, cannot start before a minimal time lag after the end of its preceding operation $O_{i(j-1)}$, which takes into account of the removal time of the batch of $O_{i(j-1)}$, the minimal time lag $\tau^{\min}_{i(j-1)}$ and the setup time of batch of O_{ij}:

$$\begin{gathered} t_{ij} - t_{i(j-1)} \geq D_{m_{i(j-1)}} + p_{i(j-1)} + \tau^{\min}_{i(j-1)} + S_{m_{ij}} \\ \forall i \in \{1, \ldots, n\}, \forall j \in \{2, \ldots, n_i\} \end{gathered} \tag{7}$$

Each operation O_{ij}, $j > 1$, has to start before a maximal time lag after the end of its preceding operation $O_{i(j-1)}$, which takes account of the removal time of the batch of $O_{i(j-1)}$, the maximal time lag $\tau^{\max}_{i(j-1)}$ and the setup time of batch of O_{ij}:

$$\begin{gathered} t_{ij} - t_{i(j-1)} \leq D_{m_{i(j-1)}} + p_{i(j-1)} + \tau^{\max}_{i(j-1)} + S_{m_{ij}} \\ \forall i \in \{1, \ldots, n\}, \forall j \in \{2, \ldots, n_i\} \end{gathered} \tag{8}$$

The start times of two operations of the same batch must be equal:

$$t_{ij} = t_{xy} \quad \forall B \in \mathcal{B}, \forall O_{ij}, O_{xy} \in B \tag{9}$$

An operation of a batch which is not at the first position on its machine cannot start before the end of the preceding batch on the machine, plus the necessary removal time of the preceding batch, plus the minimal setup time on the machine between two batches, plus the necessary setup time for the next batch.

$$\begin{gathered} t_{ij} - t_{xy} \geq p_{xy} + D_k + s_k + S_k \\ \forall B_{kq,q>1} \in \mathcal{B}, \forall O_{ij} \in B_{kq}, \forall O_{xy} \in B_{k(q-1)} \end{gathered} \tag{10}$$

An operation of a batch in the first position on its machine cannot start before the initial setup time for this batch (we assume $S^0_k \leq S_k + D_k + s_k, \forall M_k \in \mathcal{M}$):

$$t_{ij} \geq s^0_{m_{ij}} \quad \forall O_{ij} \in \mathcal{O} \tag{11}$$

By definition, a feasible solution includes each operation inside a batch. In our problem, the scheduling horizon is limited to a scheduling horizon T. Hence only those batches released in the interval $[0, T]$ must be taken into account. Several criteria are used to measure the quality of a feasible solution. The number of moves is the number of wafers produced in $[0, T]$:

$$f_{\text{mov}} = \sum_{\theta_{ij} \in \theta} w_i \theta_{ij} \tag{12}$$

where $\theta_{ij} = 1$ when the job is completed before T and $\theta_i \in [0, 1[$ denotes the completion ratio of job i before time T, otherwise.

$$\theta_{ij} = \frac{\sum_{O_{ij}, t_{ij}+p_{ij}<=T} p_{ij} + \sum_{O_{ij}, t_{ij}<T<t_{ij}+p_{ij}} (T - t_{ij})}{\sum_{j=1}^{n_i} p_{ij}}$$

The batching coefficient is the average ratio of the actual size of each batch divided by its maximal size. Let $\mathcal{B}^T = \{B \in \mathcal{B} | t_{ij} < T, \forall O_{ij} \in B\}$ denote the set of batches started before time period T.

$$\text{Batching coefficient} = \frac{\sum_k \sum_{B_{kq} \in \mathcal{B}^T} |B_{kq}|/R_k}{|\mathcal{B}^T|} \tag{13}$$

$$f_{\text{batch}} = \frac{\sum_k \sum_{B_{kq} \in \mathcal{B}^T} |B_{kq}|/(R_k + \frac{Z_k}{100})}{|\mathcal{B}^T|} \tag{14}$$

where Z_k means the total number of qualified recipes on machine M_k.
The weighting of the batching coefficient by the number of qualified recipes on the concerned machine makes it possible to use the least general-purpose machines initially.

The average X-factor is the average of the X-factor of each job weighted by the job priority. All the jobs do not have the same priority. Certain jobs are more important than others (important customers, tests to be carried out quickly, delay to catch up with, etc). Thus, a weight is assigned to the job to reflect its priority. Let $\mathcal{J}^T = \{J_i \in \mathcal{J} | t_{in_i} + p_{in_i} \leq T\}$ denote the set of jobs completed before T.

$$\text{X-fac} = \frac{\sum_{J_i \in \mathcal{J}^T} (t_{in_i} - r_i)/p_{in_i}}{|\mathcal{J}^T|} \tag{15}$$

$$f_{\text{X-fac}} = \frac{\sum_{J_i \in \mathcal{J}^T} c_i (t_{in_i} - r_i)}{|\mathcal{J}^T|} \tag{16}$$

The choice made together with the decision makers of the production unit is to combine these different objectives into a single one by maximizing the following weighted sum:

$$f = \alpha f_{\text{mov}} + \beta f_{\text{batch}} - \gamma f_{\text{X-fac}} \tag{17}$$

where α, β and γ are adjustable weights allocated to each objective function. The objective functions have been designed to integrate some requests of managers. In the calculation of f_{X-fac}, the delay of each lot is multiplied by its priority, in order to accelerate the urgent lots. In the calculation of the total batching coefficient, the batching coefficient of each equipment is multiplied according to the number of qualified recipes on this equipment. This leads to choose in priority the machines that are able to process less recipes, and thus to maintain the availability of the most flexible equipment.
The objective of the problem is to search for a feasible selection $\{\mathcal{B}, \mathcal{T}\}$ such that f is maximized. Note that, given a (feasible) solution $\{\mathcal{B}, \mathcal{T}\}$, f can be computed in $O(N)$ time where $N = \sum_{i=1}^{n} n_i$ is the total number of operations.

4 The disjunctive graph representation

The considered problem can be seen as an extension of the job-shop problem and, consequently, the disjunctive graph model can be used for batching and scheduling problems as proposed by Mason et al. [MFC02]. In their article, the authors consider a different objective function (total weighted tardiness). Sequence-dependent setup times and reentrant flows are also considered but there are no maximum time lags. Unfortunately, these constraints considerably increase the difficulty of the problem Gentner et al. [GNST04]. Let us explain how the problem is modeled using disjunctive graphs.
We define the disjunctive graph $G = (V, C, E)$ as follows.

- V is a set of nodes where there is one node per operation, denoted V_{ij}, plus a dummy start node denoted 0.

- C is the set of conjunctive arcs representing the release dates and minimal and maximal time lags. There is an arc from node 0 to node V_{i1} of each job J_i. There is an arc from V_{ij} to $V_{i(j+1)}$ and an arc from $V_{i(j+1)}$ to V_{ij} , for each consecutive operations O_{ij} and $O_{i(j+1)}$ of each job J_i.
- E is the set of disjunctive arcs which represent the decisions of the problem. There are two opposite conjunctive arcs $(V_{ij}, V_{xv})^k$ and $(V_{xy}, V_{ij})^k$ for any machine $k \in \mathcal{M}$ and for any pair of operations $O_{ij}, O_{xy} \in \mathcal{O}_k$, $O_{ij} \neq O_{xy}$. These arcs represent the sequencing or the batching decision concerning O_{ij} and O_{xy} on machine k.

Let $\mathcal{B}$ denote a partial or complete batching for the problem satisfying at least Constraints (1) through (4). $\mathcal{B}$ is a complete batching if Constraint (5) is also verified, otherwise it is a partial batching. Recall that $\mathcal{B}$ also defines the machine assignment m_{ij}, for all $O_{ij} \in \cup_{B\in\mathcal{B}} B$. We assume $m_{ij} = 0$ if O_{ij} is not batched in $\mathcal{B}$, i.e. if $O_{ij} \notin \cup_{B\in\mathcal{B}} B$.

$\mathcal{B}$ unambiguously defines a selection $\mathcal{S}$ as follows. For each distinct operations O_{ij} and O_{xy} such that $m_{ij} = m_{xy} = k \neq 0$, select arc $(V_{xy}, V_{ij})^k$ if O_{xy} is in a batch sequenced before the batch of O_{ij}, select arc $(V_{ij}, V_{xy})^k$ if O_{ij} is in a batch sequenced before the batch of O_{xy}, and select both arcs $(V_{ij}, V_{xv})^k$ and $(V_{xy}, V_{ij})^k$ if O_{ij} and O_{xy} are assigned to the same batch.

Once a selection is computed, we define a graph $G(\mathcal{S}) = (V, C \cup \mathcal{S})$ where arcs $C \cup \mathcal{S}$ are valued as follows (we assume $s_0^0 = s_0 = S_0 = D_0 = 0$):

- Each arc from 0 to the first operation O_{i1} is valued by $\max(r_i, S^0_{m_{ij}})$, the maximal value between the release date of job i and the initial setup time for machine m_{i1}.
- Each arc from V_{ij} and $V_{i(j+1)}$ is valued by $D_{m_{ij}} + p_{ij} + \tau^{\min}_{ij} + S_{m_{i(j+1)}}$, the value of the minimal time lag between O_{ij} and $O_{i(j+1)}$ plus the setup and removal times linked to the assignment of the operations.
- Each arc from $O_{i(j+1)}$ and O_{ij} is valued by $-(D_{m_{ij}} + p_{ij} + \tau^{\max}_{ij} + S_{m_{i(j+1)}})$, the (negative) value of the maximal time lag between O_{ij} and $O_{i(j+1)}$ plus the necessary setup and removal times.
- Each arc $(V_{ij}, V_{xv})^k$ is valued either by $p_{ij} + D_k + s_k + S_k$ if the opposite arc is not selected or by 0 if the opposite arc is selected. Indeed, in the first case this arc represents the decision to sequence O_{xy} after O_{ij} on machine k whereas in the second case, both arcs $(V_{ij}, V_{xv})^k$ and $(V_{xv}, V_{ij})^k$ represent the synchronization of the operations included in the same batch.

We can state that the (partial) solution represented by the (partial) batching $\mathcal{B}$ and its selection $\mathcal{S}$ is feasible if and only if longest path problems from node 0 to each node V_{ij} in $G(\mathcal{S})$ have a solution. In the positive case and if $\mathcal{B}$ is complete, a feasible schedule $\mathcal{T}$ can be obtained by setting t_{ij} to the length of the longest path from 0 to V_{ij}. Furthermore $\mathcal{T}$ is the best schedule compatible with $\mathcal{B}$ one can obtain when the objective is to maximize f.

The problem can be formulated as follows: Find the batching $\mathcal{B}$ verifying Constraints (1) through (5) such that the corresponding selection $\mathcal{S}$ is feasible and maximizes f.

As stated in the previous sections, the actual data issued from the fab have other characteristics that have been tackled, such as in-process jobs and machine down times. We can model easily these characteristics thanks to the disjunctive graph formulation. They can be both represented by operations with fixed start times on the involved machine. A start time t can be fixed by linking the node with node 0 by two opposite arcs valued by t and $-t$. As stated in Section 3, the actual problem also includes an important line management constraint: a minimum time between the start time of two batches of the same recipe scheduled on two different batches has to be respected. This can also be tackled through the disjunctive graph representation. A fictitious machine can be associated to each recipe and disjunctive arcs linking two operations of the same recipe can be defined. Then, whenever the two operations are batched on two different machines (furnaces), the disjunctive arc has to be oriented. Once oriented the arc is valued by the minimum required distance.

5 Solution Methods

We propose a two-phase heuristic method and a metaheuristic to solve the problem. The first phase of the heuristic is a constructive heuristic based on successive job insertions. The second phase is a local

search method which aims at improving the initial solution. Both phases are based on the evaluation of a (complete or partial) selection through longest path calculations. For the metaheuristic, we propose a Simulated Annealing algorithm.

5.1 Evaluation of a partial or complete batching

Any partial or complete batching $\mathcal{B}$ and its selection $\mathcal{S}$ can be evaluated through the calculation of start time t_{ij}, equal to the longest path from 0 to V_{ij} in $G(\mathcal{S})$ of each operation O_{ij}. To compute such longest paths, since the graph includes arcs with negative weights, we use the Bellman-Ford algorithm which has a $O(N|\mathcal{S} \cup C|)$ time complexity. If the algorithm finds a path of positive length, then the partial or complete solution is unfeasible. Otherwise the algorithm returns start times t_{ij} and the objective function value f can be determined.

5.2 Computing an initial solution by a priority rule-based constructive heuristic

The initial solution (selection) is computed by a job insertion method. The jobs are first sorted in a list L according to the order: jobs involving maximal time lags first, then increasing release dates, then job priority.

The method starts with an empty batching. Then, the jobs are taken in the order given by the list and inserted one by one in the current batching. The insertion of J_i is made as follows. Let $\mathcal{B} = \{B_{kq}\}_{k \in \mathcal{M}, q \in 1,\ldots,\nu_k}$ denote the current batching including jobs located before J_i in $\mathcal{L}$. For each operation O_{ij} of J_i and for each resource $k \in \mathcal{M}_{ij}$, there are $2\nu_k + 1$ insertion positions of O_{ij} in the batch sequence of machine M_k: indeed, for each batch B_{kq}, we may insert O_{ij} inside batch B_{kq} or create a new batch at any position. Each of these insertion positions is evaluated with the algorithm described in the previous section and the one that maximizes f is kept to update $\mathcal{B}$. If none of the insertion positions is feasible for an operation O_{ij}, then all these partial solutions violate a maximal time lag and there exists insertion positions that violate only maximal time lag between $O_{i(j-1)}$ and O_{ij} (the last position on each resource of $\mathcal{M}_{ij}$, for instance). Hence we have to remove $O_{i(j-1)}$ from $\mathcal{B}$, the previous operation of job J_i, and insert it at a later insertion position. If this is not feasible, they are not scheduled and delayed from the list of operations. Note that at most $\sum_{j=1}^{n_i} \sum_{k \in \mathcal{M}_{ij}} 2\nu_k + 1$ insertion positions are tested per inserted lot. Thereafter, we will call by Priority-rule Based Insertion Algorithm (PBIA), the algorithm described above.

5.3 Improving the initial solution by iterative sampling

Iterative random sampling consists in iteratively applying the constructive algorithm presented in Section 5.2 and randomly changing each time the input operation list. Two different iterative sampling methods are proposed. In the first random sampling method the input list is initialized at each iteration as a random permutation. We call this algorithm Random Iterative Algorithm by Priority-rule Based Insertion (RIA-PBIA). In the second random sampling method, the list is changed by making few random modifications on the sorted list. It consists in dividing the list into intervals and, for each interval, in making a random modification following a uniform law. We call this algorithm Pseudo-Random Iterative Algorithm by Priority-rule Based Insertion (PRIA-PBIA).

5.4 Simulated Annealing

Simulated annealing (SA) belongs to the class of randomized local search algorithms and has been developed by [KGV83] to handle hard combinatorial problems. The use of simulated annealing supposes the definition of neighborhood. The neighborhood we consider is defined by the following moves:

1. "Batch move". A batch is moved randomly (both equipment and position can change).

2. "Operation move". An operation is moved randomly in an existing batch or a new batch is created (both equipment and position can change).

3. "Operation switch". Two batches with the same recipe are randomly selected. Then two randomly selected operations are switched.

In our implementation, 50% of the considered moves are batch moves, 25% are operation moves and 25% are operation switches. According to randomly generated moves, we proceed as follows. All random selections are made according to the uniform law. If a move is impossible due to a violation of a constraint, we restore the previous solution and try another move. Each neighbor solution is evaluated by means of the longest path computations presented in Section 5.1.

The algorithm starts with an initial solution $s0$ and then tries to find better solutions by searching in its neighborhood (obtained by the moves described above), and applying a stochastic acceptance criterion. When a neighbor (a solution s) of the current solution is selected we compare the difference $\Delta f = f(s) - f(s0)$. If the difference Δf is negative, the neighbor replaces the current solution. Otherwise, the neighbor is accepted with a probability based upon the Boltzmann distribution $P_{accept}(\Delta f) \approx exp(\frac{-\Delta f}{kT})$ where k is a constant and the temperature T is a control parameter. This temperature is gradually lowered following a geometrical law function $g(T) : g(T) = \alpha T$ with $\alpha < 1$.

6 Computational experiments

We coded the proposed algorithms in Java and we tested them on a 1 GigaOctet RAM and 3.4 gigahertz processor computer. We conducted two types of experiment tests: the first type of tests is conducted on actual fab data and compared to the constructive initial solution obtained by PBIA solutions, with pseudo and random lists and the Simulated Annealing (SA) solutions. In the second type of tests, we compared the proposed heuristic PBIA to an algorithm for scheduling jobs on parallel batch processors with incompatible job families and unequal ready times problem proposed by Mönch et al. [MBFP05] as it is close to the one tackled in this paper.

6.1 Experimental tests on actual fab data

These tests have been performed on actual instances issued for two months of production. There are 700 jobs yielding a total number of 1400 operations with about 50 different recipes to schedule on 70 furnaces and 12 cleaning machines. Each furnace has a capacity of 4 or 6 lots and each cleaning machine of 2 or 4 lots. The time needed to load and unload a batch varies between 10 and 30 minutes. The minimum and maximum time lags varie from 10 minutes to 4 hours. The target time horizon is 24 hours. The weights of the components of the objective function have been computed both to normalize the different indicators and to take account of the user preferences. For the experiments, we selected $\alpha = 601$, $\beta = 1500001$, $\gamma = 41$.

The number of replications is 5 for the simulated annealing and 20 for the pseudo and random lists. The computational time of the simulated annealing for one instance does not exceed 5 minutes and the constructive heuristic and the random and pseudo lists is limited to 1 minute. The percentage is calculated on the average relative improvement brought on the considered method over the reference solution obtained by the constructive heuristic (initial), i.e. $\frac{Method\,solution - Initial\,list}{Initial\,solution}$

From Table 2 and Table 3, we displayed the objective function solution values (note), the number of moves (f_{mov}), the batching coefficient (f_{batch}) and the X-factor (f_{X-fac}) obtained from the three methods: Simulated Annealing (SA), Pseudo (PRIA-PBIA) and Random (RIA-PBIA) algorithms on the basis of solutions obtained by PBIA (Initial). It is important to note that the Simulated Annealing starts from the best solution of (PRIA-PBIA) and (RIA-PBIA).

We can notice that on average we improve solutions with the simulated annealing algorithm on the two considered months. An improvement of 47,05% on the number of moves f_{mov} is noticed (resp. 6,6% for the second month), the batching coefficient f_{batch} of 33,04% (resp. 4,15% for the second month), the f_{X-fac} increases by 41% (resp. is improved of 4,07% for the second month) and the objective value (note) is improved of 11,13% for the first month (resp. 12,61% for the second month).
We improve on average initial solutions with PRIA-PBIA (Pseudo-Random) of 38,32% on the number of

moves (resp. slightly deteriorated of 0,95% on second month), the batching coefficient of 28,23% (resp. 0,70% on the second month) and X-factor increases of 47,07% (resp. improved of 0,26% on the second month) and the objective function slightly decreases of 1,71% (resp. 1,42% on the second month).
For the Random solutions (RIA-PBIA), the number of moves is improved of 37,26% (resp. decreased of 2,05% on the second month), the batching coefficient is improved of 29,96% (resp. 1,17% on the second month), the X-factor is deteriorated of 48,93% (resp. 1,16% on the second month) and the note slightly decreased of 3,86% (resp. 6,30% on the second month).
In the Pseudo and random cases, we can notice that some indicators are sometimes slightly deteriorated (more in the case of Random lists) due to the fact we have a larger search space when using random lists than pseudo-random lists and then there is a capability of reaching better solutions but also worse see Table 4.

	Simulated Annealing				Pseudo-Random (PRIA-PBIA)				Random (RIA-PBIA)				Initial			
Inst.	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{batch}	f_{X-fac}	Note
M1D1	18812	0,7565	2,4841	0,8621	14248	0,798	2,566	0,7926	14102	0,753	2,495	0,8000	13976	0,7228	2,5802	0,78
M1D2	13910	0,6981	2,1334	0,8555	13571	0,659	2,260	0,7668	13227	0,680	2,303	0,7495	13671	0,6555	2,2507	0,78
M1D3	14659	0,7290	2,5914	0,8086	14233	0,727	2,716	0,7488	14622	0,730	2,729	0,7634	14086	0,7057	2,7470	0,72
M1D4	18068	0,7922	2,4985	1,0653	17509	0,773	2,586	0,9988	16930	0,769	2,645	0,9218	17585	0,7678	2,5416	1,01
M1D5	16941	0,7791	2,7896	0,9287	16291	0,747	2,995	0,8061	16463	0,752	2,984	0,8102	16166	0,7450	2,9775	0,81
M1D6	17861	0,7841	2,5404	1,0418	17091	0,759	2,729	0,9253	17352	0,762	2,642	0,9626	17376	0,7504	2,6853	0,95
M1D7	18812	0,7937	2,4174	1,1245	17466	0,772	2,462	1,0227	17318	0,774	2,617	0,9517	17534	0,7565	2,4240	1,03
M1D8	17919	0,7843	2,8593	0,9586	17177	0,771	2,946	0,8840	16643	0,790	3,064	0,8563	17201	0,7639	2,8522	0,91
M1D9	18759	0,8073	3,0423	0,9825	18334	0,798	3,220	0,8988	17534	0,788	3,271	0,8146	18035	0,7921	3,2042	0,89
M1D10	16296	0,7891	2,8229	0,8967	15670	0,777	2,969	0,8162	15191	0,780	3,039	0,7490	15305	0,7751	3,0099	0,79
M1D11	16016	0,7623	2,7888	0,8659	14986	0,733	2,869	0,7534	15085	0,761	2,905	0,7555	14950	0,7047	2,8685	0,74
M1D12	16554	0,7975	3,0114	0,8708	15853	0,767	3,059	0,7786	15652	0,777	3,067	0,7616	15929	0,7576	3,0715	0,79
M1D13	16899	0,8275	2,6692	0,9940	15933	0,786	2,723	0,8694	15779	0,785	2,745	0,8679	15489	0,7892	2,8445	0,85
M1D14	16540	0,7585	2,4582	0,9707	15891	0,732	2,540	0,8753	15276	0,728	2,546	0,8296	16088	0,7199	2,6123	0,87
M1D15	16839	0,8154	2,7518	0,9636	15886	0,776	2,894	0,8404	15693	0,780	2,888	0,8215	15936	0,7688	2,8145	0,86
M1D16	14132	0,6944	2,3187	0,8200	13399	0,679	2,445	0,7375	13253	0,683	2,550	0,6928	13424	0,6752	2,4395	0,74
M1D17	15994	0,7681	2,3630	0,9734	15047	0,741	2,423	0,8782	15103	0,744	2,482	0,8679	15272	0,7216	2,3820	0,89
M1D18	15138	0,7735	2,5109	0,9026	13991	0,749	2,723	0,7646	14347	0,760	2,723	0,7787	14435	0,7411	2,6477	0,80
M1D19	15846	0,7911	2,6721	0,9145	14766	0,747	2,815	0,7805	14735	0,793	2,835	0,7752	14742	0,7300	2,7726	0,77
M1D20	17563	0,8150	2,8739	0,9636	16223	0,782	3,005	0,8088	15890	0,770	3,152	0,7590	16641	0,7665	2,9525	0,86
M1D21	17264	0,8041	2,4374	1,0553	16389	0,762	2,510	0,9443	16019	0,759	2,547	0,8682	16546	0,7602	2,4584	0,97
M1D22	17248	0,7609	2,4500	1,0028	16402	0,733	2,532	0,9069	16859	0,747	2,555	0,9274	16339	0,7230	2,4873	0,92
M1D23	17083	0,8010	2,4431	1,0400	16287	0,757	2,523	0,9310	16024	0,782	2,623	0,9079	16437	0,7586	2,4739	0,96
M1D24	14153	0,7740	2,5696	0,8328	13506	0,744	2,781	0,7120	13257	0,767	2,773	0,7255	13406	0,7361	2,8305	0,70
M1D25	14998	0,7541	2,4755	0,8851	14381	0,701	2,602	0,7673	14262	0,723	2,577	0,7756	14555	0,6996	2,5619	0,79
M1D26	15972	0,7891	3,0567	0,8161	14485	0,748	3,053	0,6986	14685	0,779	3,190	0,6477	15298	0,7286	3,0200	0,74
M1D27	17326	0,8299	2,5666	1,0467	16203	0,778	2,767	0,8805	16576	0,804	2,761	0,8940	16180	0,7664	2,6066	0,92
M1D28	18231	0,8165	2,8988	0,9969	17556	0,783	2,949	0,9031	16658	0,797	3,002	0,8441	17630	0,7812	2,9770	0,92
M1D29	17627	0,8155	3,3911	0,8381	15770	0,760	3,528	0,6565	15779	0,797	3,642	0,6590	16640	0,7584	3,4622	0,72
M1D30	17187	0,7956	3,1661	0,8633	16574	0,766	3,362	0,7456	16577	0,791	3,341	0,7850	16823	0,7603	3,3138	0,77
M1D31	17395	0,8036	3,5026	0,7977	16183	0,778	3,602	0,6814	16648	0,795	3,560	0,7027	16028	0,7701	3,6320	0,66
Average	16711	0,7826	2,6953	0,9335	15719	0,754	2,811	0,8250	15598	0,765	2,847	0,8073	15798	0,7436	2,7904	0,84

Table 2: Results on instances issued of the first month of the fab

Inst.	Simulated Annealing				Pseudo-Random (PRIA-PBIA)				Random (RIA-PBIA)				Initial			
	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{batch}	f_{X-fac}	Note	f_{mov}	f_{Batch}	f_{X-fac}	Note
M2D1	16059	0,795	2,578	0,945	14878	0,793	2,636	0,858	14205	0,781	2,685	0,784	15414	0,7793	2,6009	0,90
M2D2	16650	0,794	2,317	1,05	15534	0,765	2,490	0,901	15189	0,760	2,428	0,879	16176	0,7576	2,4129	0,96
M2D3	16081	0,789	2,367	0,98	15165	0,760	2,427	0,893	14954	0,780	2,501	0,840	15015	0,7386	2,4331	0,88
M2D4	15407	0,811	2,611	0,92	14283	0,783	2,633	0,833	13676	0,793	2,728	0,752	14583	0,7779	2,5839	0,86
M2D5	15432	0,788	2,211	1,00	14286	0,754	2,322	0,861	14607	0,771	2,261	0,925	14360	0,7391	2,3170	0,88
M2D6	12754	0,750	2,577	0,74	11241	0,714	2,759	0,576	11601	0,716	2,661	0,589	10992	0,7312	2,6738	0,61
M2D7	16313	0,804	2,609	0,96	15189	0,774	2,785	0,834	14660	0,792	2,789	0,809	15582	0,7692	2,7674	0,86
M2D8	16659	0,838	2,740	0,99	15402	0,821	2,851	0,866	14294	0,797	2,896	0,732	15787	0,8090	2,8106	0,90
M2D9	16453	0,791	2,466	0,99	14809	0,759	2,530	0,861	13795	0,760	2,532	0,766	15428	0,7567	2,5228	0,90
M2D10	15893	0,789	2,802	0,88	14907	0,763	2,887	0,762	13777	0,768	3,007	0,680	15251	0,7658	2,9120	0,80
M2D11	13123	0,749	2,626	0,74	12039	0,742	2,703	0,630	12512	0,732	2,654	0,623	12225	0,7231	2,7133	0,66
M2D12	13142	0,776	2,664	0,75	11253	0,759	2,651	0,655	11573	0,759	2,633	0,626	12451	0,7591	2,8560	0,67
M2D13	14496	0,829	3,067	0,77	13683	0,796	3,154	0,682	12913	0,793	2,847	0,663	13810	0,7910	3,1124	0,70
M2D14	14430	0,820	2,809	0,84	13149	0,796	2,783	0,742	13238	0,824	3,041	0,677	13586	0,7883	2,8831	0,75
M2D15	12341	0,756	2,618	0,71	11349	0,717	2,847	0,570	11774	0,733	2,781	0,597	11504	0,7111	2,8052	0,59
M2D16	14213	0,763	2,531	0,83	12949	0,718	2,613	0,709	12827	0,749	2,636	0,696	13269	0,7205	2,6122	0,73
M2D17	13400	0,727	2,546	0,76	12403	0,707	2,666	0,660	12371	0,691	2,831	0,577	12600	0,7064	2,6723	0,67
M2D18	13566	0,754	2,353	0,84	12231	0,705	2,445	0,703	12725	0,729	2,541	0,689	12255	0,7036	2,4386	0,71
M2D19	12153	0,764	2,803	0,66	11553	0,744	2,987	0,570	11672	0,743	2,922	0,582	11578	0,7379	3,0130	0,56
M2D20	11663	0,759	2,694	0,66	10940	0,739	2,829	0,578	11165	0,752	2,958	0,532	10939	0,7388	2,8204	0,58
M2D21	13457	0,743	2,413	0,81	12872	0,731	2,474	0,743	12650	0,716	2,654	0,661	12672	0,7356	2,5137	0,74
M2D22	13477	0,694	2,613	0,71	13392	0,671	2,832	0,632	12819	0,686	2,875	0,602	12734	0,6490	2,8218	0,58
M2D23	13647	0,692	2,804	0,67	13427	0,664	2,759	0,644	13844	0,679	2,800	0,646	13341	0,6554	2,7985	0,62
M2D24	13761	0,692	2,444	0,76	13234	0,680	2,465	0,709	13328	0,675	2,466	0,710	13211	0,6742	2,4502	0,72
M2D25	12721	0,683	2,339	0,73	12480	0,667	2,601	0,641	12384	0,675	2,595	0,631	12379	0,6672	2,6394	0,63
M2D26	13428	0,690	2,208	0,81	12232	0,654	2,345	0,676	12600	0,651	2,367	0,683	12382	0,6564	2,3503	0,69
M2D27	12270	0,678	2,450	0,68	11846	0,672	2,691	0,592	11379	0,665	2,817	0,516	11449	0,6507	2,7231	0,54
M2D28	14153	0,735	2,721	0,75	13277	0,714	2,841	0,661	13474	0,701	2,892	0,634	13252	0,7128	2,8649	0,66
M2D29	12795	0,719	2,748	0,67	12495	0,683	2,743	0,622	12289	0,691	2,905	0,550	12712	0,6887	2,7403	0,64
M2D30	13182	0,681	2,450	0,72	12314	0,664	2,662	0,603	12136	0,660	2,697	0,576	12311	0,6629	2,6743	0,61
M2D31	14244	0,741	2,577	0,80	13649	0,707	2,514	0,722	13488	0,702	2,704	0,673	13126	0,7008	2,5991	0,71
Avg.	14108	0,755	2,573	0,810	13176	0,730	2,675	0,709	13030	0,733	2,713	0,674	13302	0,7244	2,6818	0,72

Table 3: Results on instances issued of the second month of the fab

One objective of this study was to develop and propose an efficient scheduler algorithm to support fabrication operators and improve the principal indicators in the diffusion area. Our proposed algorithm

	Simulated Annealing				Pseudo-Random (PRIA-PBIA)				Random (RIA-PBIA)			
	f_{mov}	f_{batch}	f_{x-fac}	Note	f_{mov}	f_{batch}	f_{x-fac}	Note	f_{mov}	f_{batch}	f_{x-fac}	Note
Avg. std. on 1^{st} Month	0,51%	0,62%	0,62%	1,01%	1,76%	0,58%	1,05%	2,52%	3,93%	1,60%	2,26%	5,76%
Avg. std on 2^{nd} Month	0,93%	0,50%	1,09%	1,44%	1,73%	0,60%	0,99%	2,21%	4,21%	1,74%	2,57%	6,33%

Table 4: Standart deviation values on the first and second month

(PBIA) has been implemented in this area and is actually running. We summarize some results obtained after the algorithm has been implemented. The comparison is done before and after using our algorithm. For confidential reasons, we are not allowed to describe the real values of the fab. This is why, with the agreement of the staff, we give results on equipment Type 1 and Type 2. They are respectively clean equipment and furnace equipment. Two indicators are presented: the X-factor and the batching coefficient.

For the first month, the X-factor on the Type 1 equipment (resp. the Type 2 equipment) is improved of 26% (resp. 17% for the Type 2 equipment). The batching coefficient is also improved by 20% for the

Oxidation and 29% for the Deposition equipment. For the second month, the X-factor is improved by 36% for the Type 1 and 23% for the Type 2 equipment. On the other hand, the batching coefficient is deteriorated by 6% for the Oxidation equipment and an improvement of 2% is noticed on the Deposition equipment. Substantial productivity improvements have been achieved in the diffusion area of the plant thanks to Priority-based Insertion algorithm (PBIA).

6.2 Experimental tests on data issued from Mönch et al. [MBFP05]

We want to compare our algorithm (PBIA) to an algorithm proposed in the literature for a close problem. We decided to compare it to a scheduling of jobs on parallel machines with incompatible job families and unequal ready times Mönch et al. [MBFP05].

We show through experimental tests that the constructive heuristic PBIA we designed for the specific problem of batching and scheduling in a diffusion area can be easily adapted to solve the problem described by Mönch et al. [MBFP05]. Furthermore, our proposed heuristic outperforms the heuristic described in their article on some instances for the problem of minimizing Total Weighted Tardiness (TWT).

The aim of the problem tackled in Mönch et al. [MBFP05] consists in scheduling of parallel batch processing machines with incompatible job families and unequal ready times of jobs to minimize TWT include:

- Jobs of the same family have the same processing time.
- All the batch processing machines are identical in nature.
- Once a machine is started, it cannot be interrupted, i.e. no preemption is allowed.

To solve this NP-hard problem the authors proposed two different decomposition approaches. The first approach constructs fixed batches, then assigns these batches to the machines using a genetic algorithm (GA), and at last, sequences batches on each machine. The second approach first assigns jobs to machine using a GA, then constructs the batches on each machine for the jobs assigned to it and at last, sequences these batches. In conclusion, among the tested algorithms, the authors proved on their tests that the algorithm GA 2 BATC-II which is in the second type of approach provides in average better results.
As the authors used a genetic algorithm, the Table 5 summarized the different parameters used in their algorithm (see Mönch et al. [MBFP05]).

Parameter	Value
Population	200
Crossover probability	0.80
Mutation probability	0.01
Replacement probability	0.60
Diversity	0.03
Number of generations	1000

Table 5: Parameter settings GA for approach 2 Mönch et al. [MBFP05]

After taking the 162-instances (randomly generated) from the authors, we have conducted experiment tests on the same computer as one described in the subsection 6.1.
Let us recall that in this subsection the objective function is the TWT and the goal is to compare the TWT obtained from our proposed heuristic (PBIA) to algorithm GA 2 BATC-II proposed in Mönch et al. [MBFP05].
As it is known, the input of our proposed algorithm is a list. Then, to give a list to PBIA, we retain the best one list between two considered lists as follows. The first list is ordered by increasing order of due date and the second one is ordered by decreasing order of the priority of jobs. As the evaluation of a list does not require much time (less than 3 minutes), for each of the two lists we calculate the TWT and we retained the minimum one. With the solution obtained by the minimum one, we apply local search improvement (see Subsection 5.3). We have reported in Table 6 the assessment of this comparison.

From Table 6, we notice that on average as well as for the case of $m = 3$, $m = 4$ and $m = 5$, our heuristic (PBIA)is better in terms of objective function and computational times. In the considered article, the

Compare	GA 2 BATC II		PBIA		number of times we outperform their algorithm
	Objective function	Time	Objective function	Time	
Machines					
m=3	412,231	34210	411,149	218	29
m=4	300,137	28241	278,088	168	43
m=5	230,616	19198	206,133	149	46
Batch size					
B=4	389,084	17303	367,578	140	59
B=8	239,572	37129	229,336	216	59

Table 6: Comparison between our proposed algorithm and one of the best algorithm in Mönch et al. [MBFP05]

authors in one of the phase of their algorithm used genetic procedure. In general, this requires relative long computation times. The proposed constructive heuristic requires low computing time in comparison to GA 2 BATC-II. On 162 tested instances, PBIA outperforms GA 2 BATC-II on 118 instances in less than 200 seconds. Furthermore, with batch size of 4, PBIA outperforms 59 times on 81 and on batch size of 8, we outperformed 59 times on 81.

We have used a heuristic which was not originally designed for the problem of minimization of TWT on parallel batch machines with incompatible job families and unequal ready times of jobs. The results of the tests showed that our heuristic outperformed in the majority of the cases the GA 2 BATC-II: we had better objective function values and less computation times.

7 Concluding remarks

We proposed a model and a method based on a disjunctive graph for batching and scheduling problem in a semiconductor manufacturing factory while taking into account complex constraints and optimizing multiple measures. A constructive algorithm has been proposed to solve the problem, local search improvements based on the disjunctive graph representation have been defined and a simulated annealing algorithm has been developed. The computational tests made on real instances of the factory plant showed that we obtain quality solutions in fast calculation times. For the industry, very substantial productivity improvements have been achieved in the diffusion area and also on the overall fab thanks to the use of the proposed algorithm named Priority rule-Based constructive algorithm (PBIA).
With adjustments, the Priority-rule Based Insertion Algorithm (PBIA) has been embedded in a software used by the industry. The use of a disjunctive graph brings significant improvements for interactive scheduling at the fab level. A prototype software gives the graphical user interface of the software. It includes the off-line batching and scheduling phase and also an interactive module that allows the decision-makers to make modifications and to test options directly on the proposed plan.

Regarding, the simulated annealing algorithm, the procedures significantly improve the different criteria (number of moves, batching coefficient and X-factor) in comparison with a local search method. We also compared our algorithm to an algorithm designed for a problem of scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times Mönch et al. [MBFP05] as it is close to the one tackled in this article. The computational tests showed that for more than half of the instances our heuristic called Priority-rule Based Insertion (PBIA) outperforms the algorithm proposed by Mönch et al. [MBFP05].

This research can be extended in many ways. Other types of moves such as the simultaneous move of two jobs linked by a maximum time lag should be tested. The maximal time lags are currently hard constraints. However in practice some of them can be relaxed and treated as soft constraints or objectives. Another important issue can be in a multicriteria analysis study. Computing several Pareto optimal solutions are useful particularly when the situation frequently changes as it is usually the case of semiconductor manufacturing.

Acknowledgments

This work was part of the MEDEA+ European project HYMNE (High Yield driven MaNufacturing Excellence in sub 65 nm CMOS), partly funded by the "Ministère de l'Économie, de l'Industrie et de l'Emploi" (French Ministry of Economy, Industry and Employment).

References

[ABZ88] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34:391–401, 1988.

[BMFP04] H. Balasubramanian, L. Mönch, J.W. Fowler, and M.E. Pfund. Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. *International Journal of Production research*, 48(8):1621–1638, 2004.

[CPPZ02] R. Cigolini, M. Perona, A. Portioli, and T. Zambelli. A new dynamic look-ahead scheduling procedure for batching machines. *Journal of scheduling*, 5(2):185–204, 2002.

[DU00] E. Demirkol and R. Uzsoy. Decomposition methods for reentrant flow shops with sequence dependent setup-times. *Journal of Scheduling*, 3:155–177, 2000.

[GNST04] K. Gentner, K. Neumann, C. Schwindt, and N. Trautmann. Batch production scheduling in the process industries. In J.Y.T. Leung, editor, *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, pages 48.1–48.21. CRC Press, Boca Raton, 2004.

[GR88] C. Glassey and M. Resende. Closed-loop job release control for vlsi circuit manufacturing. *IEEE Transactions on Semiconductor manufacturing*, 1(1):36–46, 1988.

[Hun98] Y.-F. Hung. Scheduling of mask shop e-beam writers. *IEEE Transaction of Semiconductor Manufacturing*, 11(1):165–172, 1998.

[ICN+03] K. Ibrahim, M. A. Chik, W.S. Nizam, N. L. Fem, and N.F. Za'bah. Efficient lot batching system for furnace operation. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 322–324, 2003.

[KGV83] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, tome 220(4598):671–680, 1983.

[KJC10] Y.-D. Kim, B.-J. Joo, and S.-Y. Choi. Scheduling wafer lots on diffusion machines in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing*, 23(2):246–254, 2010.

[Kum94] P.R. Kumar. Scheduling semiconductor manufacturing plants. *IEEE Control Systems Magazine*, 14(6):30–40, 1994.

[Lit61] J. Little. A proof for the queuing formula $l = \lambda w$. *Operations Research*, 16:651–665, 1961.

[MBFP05] L. Mönch, H. Balasubramanian, J.W. Fowler, and M.E. Pfund. Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. *Computers & Operations Research*, 32:2731–2750, 2005.

[MFC02] S.J. Mason, J.W. Fowler, and W.M. Carlyle. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, 5(3):247–262, 2002.

[MFDP+09] L. Mönch, J.W. Fowler, S. Dauzère-Pérès, S. Mason, and O. Rose. Scheduling semiconductor manufacturing operations: Problems, solution techniques, and future challenges. In *4th Multidisciplinary International Conference on Scheduling: Theory & Applications, Dublin, Ireland*, 2009.

[MH03] L. Mönch and I. Habenicht. Simulation-based assessment of batching heuristics in semiconductor manufacturing. In S. Chick, P.J. Sanchez, D. Ferrin, and D. J. Morrice, editors, *Winter Simulation Conference*, pages 1338–1345. ACM, 2003.

[MO03] S.J. Mason and K. Oey. Scheduling complex job shops using disjunctive graphs: a cycle elimination procedure. *International Journal of Production Research*, 5(41):981–994, 2003.

[MS06a] M. Mathirajan and A. I. Sivakumar. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor manufacturing. *International Journal of Advanced Manufacturing Technology*, 29:990–1001, 2006.

[MS06b] M. Mathirajan and A. I. Sivakumar. Minimizing total weighted tardiness on heterogeneous batch processing machines with incompatible job families. *International Journal of Advanced Manufacturing Technology*, 28:1038–1047, 2006.

[MT05] J. Montoya-Torres. Manufacturing performance evaluation in wafer semiconductor factories. *International Journal of Productivity and Performance management*, 55(3/4):300–310, 2005.

[MU98] S. Mehta and R. Uzsoy. Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE Transactions*, 30:165–178, 1998.

[OFKK09] A. Oulamara, G. Finke, and A. Kamgain Kuiten. Flowshop scheduling problem with batching machine and task compatibilities. *Computers and Operations Research*, 36(2):391–401, 2009.

[OM01] K. Oey and S. Mason. Scheduling batch processing machines in complex job shops. In *Proceedings of the Winter Simulation Conference*, pages 1200–1207, Arlington, USA, December 2001.

[OU97] I. Ovacik and R. Uzsoy. *Decomposition methods for complex factory scheduling problems.* Kluwer Academic Publishers, 1997.

[PBF+08] M.E. Pfund, H. Balasubramanian, J.W. Fowler, S.J. Mason, and O. Rose. A multi-criteria approach for scheduling semiconductor wafer fabrication facilities. *Journal of Scheduling*, 11(1):29–47, 2008.

[PFC05] I.C. Perez, J.W. Fowler, and W.M. Carlyle. Minimizing total weighted tardiness on a single batch process machine with incompatible job families. *Computers and Operations Research*, 32(2):327–341, 2005.

[SK03] C.S. Sung and Y.H. Kim. Minimizing due date related performance measures on two-batch processing machines. *European Journal of Operational Research*, 147:644–656, 2003.

[SM01] C.S. Sung and J. Min. Scheduling in a two-machine flowshop with batch processing machines for earliness/tardiness measure under a common due date. *European Journal of Operational Research*, 131:95–106, 2001.

[Su03] L-H. Su. A hybrid two-stage flow shop with limited waiting time constraints. *Computers and Industrial Engineering*, 44:409–424, 2003.

[Sze01] S.M. Sze. *Semiconductor devices: Physics and technology.* John Wiley & Sons, Second Edition, 2001.

[UW00] R. Uzsoy and C.-S. Wang. Performance of decomposition procedures for job-shop scheduling problems with bottleneck machines. *International Journal of Production Research*, 38:1271–1286, 2000.

[Uzs95] R. Uzsoy. Scheduling batch processing machines with incompatible job families. *International Journal of Production research*, 33:2685–2708, 1995.

[VS06] A. Varadarajan and S.C. Sarin. A survey of dispatching rules for operational control in wafer fabrication. In A. Dolgui, G. Morel, and C. E. Pereira, editors, *12th IFAC Symposium on Information Control Problems in Manufacturing*, pages 715–726, 2006.

[Wei88] L.M. Wein. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 1(3):115–130, 1988.