

SCHEDULING JOBS WITH TIME CONSTRAINTS BETWEEN CONSECUTIVE PROCESS STEPS IN SEMICONDUCTOR MANUFACTURING

Andreas Klemmt

Infinion Technologies Dresden GmbH

Königsbrücker Straße 180
Dresden, 01099, GERMANY

Lars Mönch

University of Hagen
Dep. of Mathematics and Computer Science
Universitätsstraße 1
Hagen, 58097, GERMANY

ABSTRACT

In this paper, we consider flow shop scheduling problems for jobs with time constraints between consecutive process steps. We start by analyzing different types of time constraints that arise in semiconductor wafer fabrication facilities. A simple heuristic that sequentially schedules the jobs in a list scheduling manner is proposed. Moreover, a decomposition approach based on mixed integer programming is developed. The two approaches are compared by means of randomly generated problem instances.

1 INTRODUCTION

Semiconductor wafer fabrication facilities (wafer fabs) can be modeled as complex job shops that have a large number of machines, different products with routes including several hundred operations, sequence-dependent setup times, reentrant process flows, and a mix of different processes including single wafer processing and batch processing. A batch is defined as a collection of jobs that have to be processed at the same time on one machine (Mönch et al. 2011). Frequent disturbances caused by machine breakdowns are also typical for wafer fabs.

Time constraints between consecutive process steps are another important class of constraints. For example, there is often a time constraint between operations in the etch work area and oxidation/deposition/diffusion work area (Scholl and Domaschke 2000). Time windows are installed by process engineering department to respect these time constraints. This is important to prevent native oxidation and contamination effects on the wafer surface. More than two consecutive process steps might be involved, and the time constraints might be nested. Jobs with a violation of the recommended time windows have to be scrapped. Rework is often not allowed.

Scheduling is a desirable technique in an environment as found in wafer fabs. However, dispatching is still the dominating production control technique in most wafer fabs. The problem of scheduling jobs with time constraints for flow shops and job shops with regular objectives is recently tackled by some researchers in the academic scheduling community. However, little is known for scheduling problems found in wafer fabs with reentrant flows and complex nested time constraints (Mönch et al. 2011). In this paper, we make a first attempt on modeling scheduling problems with time constraints found in wafer fabs. Furthermore, we propose two heuristics to solve a simplified version of the general scheduling problem with time constraints. Lots in semiconductor manufacturing are denoted as jobs throughout the rest of the paper to go conform with the scheduling literature.

The paper is organized as follows. The problem setting is described in Section 2 and different classes of time constraints are identified. A mixed integer program (MIP) is proposed for the researched problem. In addition, related literature is discussed in this section. The proposed solution approaches including a

simple list scheduling heuristic and a MIP-based decomposition heuristic are discussed in Section 3. The results of computational experiments are presented in Section 4.

2 PROBLEM FORMULATION

2.1 Time Constrains and Scheduling Problem

We start by describing the different classes of time constraints found in wafer fabs. A time constraint t_{jrs} holds for two process steps O_{jr} and O_{js} with $r < s$ of the same job J_i from product j if we have for the start times of the process steps $s_{is} \leq s_{ir} + t_{jrs}$, i.e., a time constraint is a maximum delay between the start times that is allowed to avoid oxidation and contamination effects of the wafers. Note that our notion of time constraints corresponds for $s = r + 1$ to positive time lags in the scheduling literature. Negative time lags are used in the scheduling literature to model a minimal delay between the start of two consecutive operations, e.g. for transportation times. In contrast to the scheduling literature, the time constraints found in wafer fabs are not necessary between immediately consecutive, i.e. adjacent, process steps. Five different classes can be distinguished for time constraints for a given product.

1. The first class consists of time constraints between two immediately consecutive process steps, i.e., we have $s = r + 1$ for the indices of the two process steps.
2. The second class allows time constraints between two process steps that are consecutive, but they are not adjacent in the product route. Hence, in contrast to the first class, we have $r + 1 < s$.
3. The third class is given by time constraints that belong to the first and the second class. Note that for two different t_{jrs} and t_{jkl} from this class with $r < k$ we have $r + 1 \leq s < k + 1 \leq l$, i.e., an overlapping of time constraints does not occur.
4. The latter condition from the third class is not fulfilled in the fourth class. Here, an overlapping explicitly takes place, i.e., we have $r \leq k < \min(l, s)$ for any two time constraints t_{jrs} and t_{jkl} with $r \leq k$.
5. Finally, the fifth class contains all the time constraints from the third and the fourth class.

Note that only the first class among the different classes is treated in the machine scheduling literature (see Subsection 2.2). The five classes are depicted in Figure 1. In this paper, we consider only the third and fourth class for flexible flow shops for the sake of simplicity.

The following notation will be used though the rest of the paper to formulate the scheduling problems and the solution schemes.

1. The jobs belongs to different products. Each product is represented by its route.
2. There exist f products. R_j represents the route of product $j, j = 1, \dots, f$.
3. Each route consists of m operations ($m > 1$). O_{jo} is the o -th operation of route $j, j = 1, \dots, f, o = 1, \dots, m$. Therefore, totally we have m operations for each product.
4. There exist m different stages in the flow shop. Each stage has a work center $W_o, o = 1, \dots, m$ that consists of m_o identical parallel machines.
5. M_{ol} is the l -th machine of $W_o, o = 1, \dots, m, l = 1, \dots, m_o$.
6. Each operation O_{jo} has to be performed on one of the machines of work center W_o .
7. There are n jobs to be scheduled. J_i represents the i -th job.
8. The product of J_i is represented by $f(i)$.
9. The processing time p_{jo} of each job of product j is assumed to be the same on all machines of work center W_o , i.e., the machines of the work center are identical.
10. The constraint t_{jop} is the maximum amount of time between the start of operation O_{jo} and the start of O_{jq} for $q > o$. If we do not have any time constraint between two operations O_{jo} and O_{jp} , we set $t_{jop} := \infty$. The set of all existing time constraints for all the products is denoted by $T := \{(j, o, p) | 1 \leq o < p \leq m\}$. In some situations, there might be initial time constraints for jobs as a

result of operations started before the begin of the scheduling horizon. Such a time constraint for operation O_{jp} of job i is denoted by t_{ip} where $j = f(i)$. The corresponding set of all initial time constraints is given by $T^{(i)} := \{(i, p) | 1 \leq p \leq m\}$.

11. The release date of job J_i is r_i .
12. The due date of job J_i is denoted as d_i .
13. The completion time of J_i is denoted by C_i .
14. The tardiness of J_i is given by $T_i = (C_i - d_i)^+$, where $x^+ := \max(x, 0)$.
15. Each machine is a single resource. Machine preemptions are not allowed.

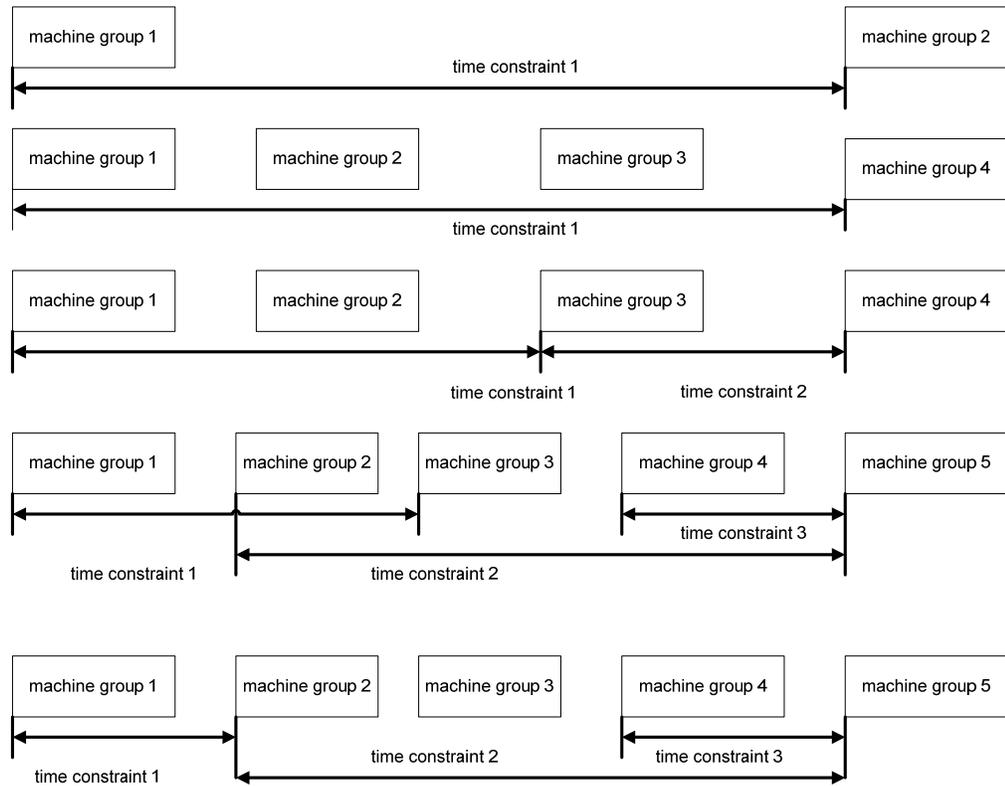


Figure 1: Classes of time constraints found in wafer fabs

Using the three-field notation from scheduling theory, the researched scheduling problem is given by

$$FF | r_i, t_{ip}, t_{jop} | TT, \tag{1}$$

where the notation FF is used for flexible flow shops and the performance measure total tardiness (TT) is given by

$$TT := \sum_{i=1}^n T_i. \tag{2}$$

Scheduling problem (1) is NP-hard because the special case when time constraints do not occur, i.e., when we have $t_{jop} = \infty$ for all products and all pairs of operations, is already NP-hard.

2.2 MIP Formulation

The proposed MIP is similar to the MIP formulations found in Klemmt (2012). The following decision variables are used within our formulation:

$s_{io} \in \mathbb{R}_+$: starting time of job J_i in operation o ($i = 1, \dots, n, o = 1, \dots, m$)

$x_{ig} \in \{0,1\}$: 1 if operation o of job J_i is scheduled before operation o of job J_g , 0 otherwise ($i, g = 1, \dots, n, o = 1, \dots, m, i < g$)

$w_{iol} \in \{0,1\}$: 1 if operation o of job J_i is scheduled on machine l ($i = 1, \dots, n, o = 1, \dots, m, l = 1, \dots, m_o$), 0 otherwise

$C_i \in \mathbb{R}_+$: completion time of job J_i ($i = 1, \dots, n$)

$T_i \in \mathbb{R}_+$: tardiness of job J_i ($i = 1, \dots, n$).

The parameter r_{ol} defines the availability time of machine M_{ol} . K is a very large positive number. The remaining parameters have been already introduced in Subsection 2.1. Now the MIP model for scheduling problem (1) can be formulated as follows:

$$\min \sum_{i=1}^n T_i \tag{3}$$

subject to

$$C_i - T_i \leq d_i, \quad i = 1, \dots, n \tag{4}$$

$$s_{io} + p_{jo} \leq C_i, \quad i = 1, \dots, n, o = 1, \dots, m, j = f(i) \tag{5}$$

$$w_{iol} r_{ol} \leq s_{io}, \quad i = 1, \dots, n, o = 1, \dots, m, l = 1, \dots, m_o \tag{6}$$

$$r_i \leq s_{i1}, \quad i = 1, \dots, n \tag{7}$$

$$\sum_{l=1}^{m_o} w_{iol} = 1, \quad i = 1, \dots, n, o = 1, \dots, m \tag{8}$$

$$s_{io} + p_{jo} \leq s_{i,o+1}, \quad i = 1, \dots, n, o = 1, \dots, m-1, j = f(i) \tag{9}$$

$$K(w_{iol} - x_{igo} - 1) + s_{go} + p_{jo} w_{gol} \leq s_{io}, \quad o = 1, \dots, m, l = 1, \dots, m_o, 1 \leq i < g \leq n, j = f(g) \tag{10}$$

$$K(w_{gol} + x_{igo} - 2) + s_{io} + p_{jo} w_{iol} \leq s_{go}, \quad o = 1, \dots, m, l = 1, \dots, m_o, 1 \leq i < g \leq n, j = f(i) \tag{11}$$

$$s_{iq} \leq s_{io} + \sum_{k=o}^{q-1} p_{jk} + t_{ioq}, \quad (j, o, q) \in T, i = 1, \dots, n, j = f(i) \tag{12}$$

$$s_{ip} \leq t_{ip}, \quad (i, p) \in T^{(i)}, i = 1, \dots, n. \tag{13}$$

The objective (3) is the TT performance measure. The constraint sets (4) and (5) restrict the objective function (3). The equation sets (6) and (7) ensure that each job can only be processed if the machine is available and the job is ready, respectively. Constraints (8) enforce that each job is processed in each stage on exactly one machine. The serial processing restrictions are given by equation set (9). The inequalities (10) and (11) are disjunctive constraints regarding the machine capacity (single resources). Constraint set (12) enforces the time constraints for the consecutive process steps. Finally, the constraints (13) make sure that the initial time constraints for starting times of process steps are fulfilled. It is interesting to see that the MIP formulation takes into account the nested time constraints in a rather elegant manner.

Because of the NP-hardness of problem (1), it has been computationally shown by some preliminary experimentation that even state of the art MIP solver can only solve problem instances up to 15 jobs and 15 machines to optimality in a reasonable amount of time. Consequently, we have to look for efficient heuristics to solve medium- and large-size problem instances.

2.3 Related Literature

There are only a few papers that address time constraints between consecutive process steps in wafer fabs. Robinson (1998) uses queueing theory to predict the reprocessing probability when lots exceed the time constraint. Time constraints between wet etch and furnace operations are discussed by Scholl and Domschke (2000). A Kanban-type mechanism is proposed to form and start batches on the wet etch bench. In addition, the situation of time constraints between two consecutive process steps including a batching operation is discussed by Choung et al. (2000) using constraint programming, by Mason et al. (2007) proposing an appropriate genetic algorithm, and by Yurtsever et al. (2009) using problem-specific iterative heuristics.

It turns out that time constraints, called time lags, are quite often considered in resource constraint project scheduling (see, for example, Dorndorf et al. 2000 and Schwindt and Trautmann 2000). General time constraints similar to our setting are discussed in these papers.

There are only a few publications in the machine scheduling literature that deal with time constraints between consecutive operations. Depper and Portmann (2006) present a dispatching rule-based construction algorithm coupled with a cluster decomposition method to come up with feasible schedules in a job shop environment with makespan objective. Fondrevelle et al. (2008) study permutation flow shop scheduling problems where maximum time lags between immediately consecutive operations occur. The total completion time criterion is considered. A heuristic based on dispatching rules and a branch-and-bound procedure are proposed.

A job shop scheduling problem with makespan objective and minimal and maximal time lags between immediately consecutive process steps is discussed by Caumond et al. (2008). A memetic algorithm that works on a disjunctive graph representation is proposed. Finally, constraint programming approaches for a somewhat similar problem are presented by Grimes and Hebrard (2010). Wang and Li (2009) propose several heuristics for a two-machine flow shop with limited waiting time constraints for a makespan objective.

To the best of our knowledge, nested time constraints have not been discussed in the machine scheduling literature so far, neither in a flow or a job shop setting with the rare exception of Depper and Portman (2006). Furthermore, it seems that most of the process conditions found in wafer fabs like batching or reentrant flows are not considered in these academically motivated papers.

3 SOLUTION APPROACHES

3.1 List Scheduling

This approach is motivated by dispatching rule-based approaches for resource-constrained project scheduling problems. A feasible schedule is constructed step by step by scheduling all the operations of a single job. When infeasibility occurs with respect to the time constraints then the affected operations are right-shifted. The scheme can be summarized as follows:

1. Initialization step: Sort the jobs in non-decreasing order with respect to their due dates, i.e. in earliest due date (EDD) order. Discretize the time axis by $t = 1, 2, \dots$
2. Schedule the first job in EDD order taking the time constraints into account.
3. Repeat for $i = 2, \dots, n$
 - a) Repeat for all stages
 - b) Determine the machine with the smallest availability time for the current stage. If the current process step can be started without time constraint violation then schedule it and go to a).
 - c) Otherwise, do a right shift of the starting time of the earlier process step that contributes to the time constraint by increasing it by one and try to schedule the process steps taking other time constraints into account until the current one that leads to the infeasibility.

This class of schedules is called canonical by Caumont et al. (2008). We denote the list scheduling by LS for abbreviation.

3.2 Decomposition Approach

This approach is an extension of the list scheduling procedure described in Subsection 3.1. Here, a feasible schedule is constructed in a step by step manner by scheduling all operations of a set of jobs in a single iteration. The feasibility of the partial schedule is ensured by solving the MIP model described in Subsection 2.2. In this model, the primary driver for complexity is the size of n and consequently the number of variables x_{igo} and the number of disjunctive constraints (10) and (11). The motivation of a decomposition approach is based on the reduction of n by iteratively solving subproblems with a maximal job size of \bar{n} . For the repeated solution of the MIP subproblems, objective function (3) is replaced by

$$K \sum_{i=1}^n T_i + \sum_{i=1}^n \sum_{o=1}^m s_{io}. \tag{14}$$

The advantage of objective function (14) is that jobs without tardiness or the same tardiness cost are additionally shifted to the left. This leads to smaller values of r_{ol} . The scheme can be summarized as follows:

1. Initialization step: Sort the jobs in non-decreasing order with respect to their due dates, i.e. in earliest due date (EDD) order. Set $r_{ol} = 0$ ($o = 1, \dots, m, l = 1, \dots, m_o$). Set $S := \emptyset$, where S is the set of already scheduled jobs and initialize the set L of unscheduled jobs, denoted by $L := \{1, \dots, n\}$.
2. Repeat for the job set I that contains the first $\min(|L|, \bar{n})$ unscheduled jobs in EDD order
 - a) Schedule job set I using the MIP model (4) to (13) for objective function (14).
 - b) Update r_{ol} as follows:

$$r_{ol} = \max(s_{io} + p_{jo} \mid i \in I, i \text{ is processed on machine } l), \quad o = 1, \dots, m, l = 1, \dots, m_o, \tag{15}$$

where s_{io} is obtained from the MIP.

- c) Update the set of scheduled jobs by $L := L - I$ and the set of already scheduled jobs by $S := S \cup I$, respectively.

Note that this decomposition approach allows for solving problem instances including more than 100 machines, more than 20 stages, nested time constraints, and a large number of jobs in near-to real time when $\bar{n} \leq 6$ is chosen. It can be used for all the classes of time constraints described in Subsection 2.1. We use the abbreviation MIP(\bar{n}) when \bar{n} jobs are used for a subproblem.

4 COMPUTATIONAL RESULTS

4.1 Design of Experiments

We are interested in assessing the performance of the two heuristics by solving randomly generated problem instances. It is expected that the performance of the heuristics depend on the number of jobs, the number of stages, the number of machines parallel in each stage, the ready time and due date setting, and finally the time constraint class and the tightness of the time constraints.

A time constraint flow factor T_F controls this tightness, i.e., we set the length of a time constraint according to

$$t_{jop} := T_F \sum_{k=0}^{p-1} p_{jk}. \tag{16}$$

The ready times of the jobs are chosen as follows. While 25% of the jobs are initially ready, the remaining jobs are uniformly distributed over the interval $[0, 0.75u]$ where

$$u = \max_o \left(\frac{\sum_{j=1}^f p_{jo} n}{m_o f} \right) \tag{17}$$

is the utilization of the bottleneck stage. The due dates are generated based on the ready dates in a standard manner using a flow factor to control their tightness.

We consider two scenarios for the time constraints. The first scenario is related to the third class and contains subsequent blocks of time constraints. The number of time constraints n_t is selected according to $DU[5,15]$. In a next step, $2n_t$ process steps from the route R_j are arbitrarily chosen in such a way that selecting the same process step two times is possible. All selected process steps are sorted in ascending order with respect to o . Now the time constraints t_{joq} of two consecutive process steps O_{jo} and O_{jq} with $q > o$ are defined in a pairwise manner where time constraints of type t_{joo} are ignored.

The second scenario contains a subset of the time constraints from the fourth class. Here, we consider single-overlapping time constraints. A maximal number $n_t \sim DU[5,15]$ of time constraints is determined in a first step for each product. Then, a process step O_{jo} with $o < m - 5$ is arbitrarily chosen. A time constraint t_{joq} between two operations O_{jo} and O_{jq} for $q = o+l, l \sim DU[3,6]$ is defined. Now a process step O_{jo} is determined in such a way that it only lies in the time interval that corresponds to the last time constraint. This procedure is repeated until n_t is reached or $q > m$ holds. O_{jq} is then generated as for the first pair of process steps with the only difference that it is not located in the interval that corresponds to the last time constraint. The resultant design of experiments is summarized in Table 1.

Table 1: Design of Experiments

Factor	Level	Count
number of products	5, 10	2
number of work centers	10	1
number of machines per work centers	$m_o \sim DU[2,4]$	1
processing times	$p_{jo} \sim DU[10,25]$	1
time constraint scenario	1,2 (class 3,4)	2
time constraint flow factor	1.5, 3	2
number of jobs per route	10	1
release date	$r_i \sim U[0, 0.75u]$	1
due date	$d_i := r_i + 2 \sum_{o=1}^m p_{f(i)o}$	1
	number of independent problem instances per factor combination	5
	total number of problem instances	40

4.2 Results

We present computational results for the 40 problem instances in Table 2. Here, we use the decomposition approach for $\bar{n} := \{3,4,5,6\}$. The corresponding maximum computing times per subproblem is five seconds for $\bar{n} \leq 5$ and 30 seconds for $\bar{n} = 6$. We show the TT values for the corresponding heuristic relative to the best TT value among the five schemes for each problem instance. Note that LS requires less

than 0.5 seconds for a single problem instance. The MIP solver IBM ILOG CPLEX 12.4 is used on a Quad core (2.6GHz) computer with 4 GB RAM. The smallest TT value for each problem instance is marked as bold.

Table 2: Computational Results

Instance	Scenario	n	T_F	LS	MIP(3)	MIP(4)	MIP(5)	MIP(6)
1	1	50	1.5	1.000	1.000	1.000	1.000	1.000
2	1	50	1.5	1.000	1.000	1.000	1.000	1.000
3	1	50	1.5	1.000	1.000	1.000	1.000	1.000
4	1	50	1.5	1.000	3.000	11.000	11.000	9.000
5	1	50	1.5	1.000	1.000	1.000	1.000	1.000
6	1	50	3.0	1.000	1.000	1.000	1.000	1.000
7	1	50	3.0	2.742	1.318	1.000	1.288	1.136
8	1	50	3.0	2.081	1.811	1.108	1.000	1.270
9	1	50	3.0	3.040	1.840	1.000	1.600	1.200
10	1	50	3.0	2.000	1.000	1.000	1.000	1.000
11	1	100	1.5	1.231	1.196	1.081	1.014	1.000
12	1	100	1.5	1.000	1.000	1.000	1.000	1.000
13	1	100	1.5	1.327	1.091	1.070	1.002	1.000
14	1	100	1.5	1.201	1.113	1.078	1.022	1.000
15	1	100	1.5	1.395	1.044	1.052	1.084	1.000
16	1	100	3.0	1.214	1.112	1.197	1.109	1.000
17	1	100	3.0	1.188	1.094	1.165	1.102	1.000
18	1	100	3.0	1.192	1.064	1.087	1.023	1.000
19	1	100	3.0	1.305	1.116	1.098	1.057	1.000
20	1	100	3.0	1.130	1.038	1.041	1.013	1.000
21	2	50	1.5	1.000	1.000	1.000	1.000	1.000
22	2	50	1.5	1.000	1.000	1.000	1.000	1.000
23	2	50	1.5	1.000	1.000	1.000	1.000	1.000
24	2	50	1.5	1.000	1.000	1.000	1.000	1.000
25	2	50	1.5	1.000	1.000	1.000	1.000	1.000
26	2	50	3.0	1.000	1.000	1.000	1.000	1.000
27	2	50	3.0	1.053	3.684	1.474	1.000	1.211
28	2	50	3.0	1.000	1.000	1.000	1.000	1.000
29	2	50	3.0	1.000	1.000	1.000	1.000	1.000
30	2	50	3.0	1.000	1.000	1.000	1.000	1.000
31	2	100	1.5	1.181	1.146	1.009	1.000	1.001
32	2	100	1.5	1.141	1.115	1.322	1.000	1.121
33	2	100	1.5	1.120	1.077	1.003	1.024	1.000
34	2	100	1.5	1.182	1.040	1.025	1.003	1.000
35	2	100	1.5	1.121	1.099	1.078	1.048	1.000
36	2	100	3.0	1.083	1.019	1.010	1.018	1.000
37	2	100	3.0	1.138	1.043	1.044	1.019	1.000
38	2	100	3.0	1.248	1.113	1.061	1.022	1.000
39	2	100	3.0	1.282	1.103	1.088	1.190	1.000
40	2	100	3.0	1.177	1.009	1.027	1.103	1.000

We can see from Table 2 that for the majority of problem instances LS has the worst performance. For 100 jobs, typically MIP(6) performs best. It seems that in this situation the larger value of \bar{n} and the larger maximum computing time of 8.5 minutes per problem instance is beneficial. But even MIP(5) with a maximum computing time of 100 seconds performs very well. It seems that scenario 2 is harder to solve as scenario 1 because in this case the difference in TT between LS and the best performing MIP approach is smaller as for scenario 1. For some instances with 50 jobs, MIP(6) is outperformed by MIP (5). Overall we conclude that the decomposition approach is an efficient heuristic for problem (1).

5 CONCLUSIONS AND FUTURE RESEARCH

In this paper, we discussed scheduling problems with time constraints as typically can be found in wafer fabs. We started by classifying the different types of time constraints found in wafer fabs. Then, we studied a simplified version of this problem, basically a flexible flow shop scheduling problem with time constraints. A list scheduling heuristic and a decomposition approach based on an MIP formulation is proposed. It turned that the MIP-based heuristic outperforms the simple heuristics.

There are different directions for future research. First of all, we are interested in proposing efficient heuristics for all the different classes of nested time constraints discussed in Subsection 2.1. Furthermore, several improvements of the proposed list-based scheduling heuristic seem to be possible. For example, instead of a job-based list scheduling approach as proposed in the present paper, we expect that it is possible to work with a stage-based list scheduling approach. We also think that constraint programming approaches similar to Grimes and Hebrard (2010) are useful for this class of scheduling problems. Finally, we believe that it is worth to extend the present heuristics to the case of flexible job shops with reentrant process flows.

ACKNOWLEDGMENTS

The second author would like to thank Volker Schmalfuß, X-FAB Semiconductor Foundries AG Erfurt, who helped him to identify and understand different problems related to time constraints in wafer fabs in course of the SimArt project.

REFERENCES

- Caumont, A., P. Lacomme, and N. Tchernev. 2008. "A Memetic Algorithm for the Job-shop with Time Lags." *Computers & Operations Research*, 35:2331-2356.
- Choung, Y.-I., K.-S. Jun, D.-S. Han, Y.-C. Jang, T.-E. Lee, and R. C. Leachman. 2001. "Design of a Scheduling System for Diffusion Processes." In *Proceedings International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM)*, 69-73.
- Deppner, F., and M.-C. Portmann. 2006. "A Hybrid Decomposition Approach Using Increasing Clusters for Solving Scheduling Problems with Minimal and Maximal Time Lags." In *Proceedings Tenth International Workshop on Project Management and Scheduling (PMS 2006)*.
- Dorndorf, U., E. Pesch, and T. Phan-Huy. 2000. "A Time-Oriented Branch-and-Algorithm for Resource Constraint Project Scheduling with Generalized Precedence Constraints." *Management Science*, 46(10):1365-1384.
- Fondrevelle, J., A. Oulamara, and M.-C. Portmann. 2008. "Permutation Flowshop Scheduling Problems with Time Lags to Minimize the Weighted Sum of Machine Completion Times." *International Journal of Production Economics*, 112:168-176.
- Grimes, D., and E. Hebrard. 2010. "Job shop Scheduling with Setup Times and Maximal Time-lags: A Simple Constraint Programming Approach." In *Proceedings 7th International Conference on Integration on Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2010)*, LNCS 6140, 147-161.

- Klemmt, A. 2012. *Ablaufplanung in der Halbleiter- und Elektronikproduktion: Hybride Optimierungsverfahren und Dekompositionstechniken*. Springer Vieweg.
- Mason, S. J., M. E. Kurz, L. M. Pohl, J. W. Fowler, and M. E. Pfund. 2007. "Random Keys Implementation of NSGA-II for Semiconductor Manufacturing Scheduling." *International Journal of Information Technology and Intelligent Computing*, 2(3).
- Mönch, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations." *Journal of Scheduling*, 14(6):583-599.
- Robinson, J. 1998. *Capacity Planning in a Semiconductor Wafer Fabrication Facility with Time Constraints Between Process Steps*. PhD Thesis, University of Massachusetts Amherst.
- Scholl, W., and J. Domaschke. 2000. "Implementation of Modeling and Simulation in Semiconductor Wafer Fabrication with Time Constraints Between Wet Etch and Furnace Operations." *IEEE Transactions on Semiconductor Manufacturing*, 13(3):273-2277.
- Schwindt, C., N. Trautmann. 2000. "Batch Scheduling in Process Industries: an Application of Resource-Constrained Project Scheduling." *OR Spektrum*, 22(4):501-524.
- Wang, B., and T. Li. 2010. "Heuristics for Two-machine Flowshop Scheduling with Limited Waiting Time Constraints." *Proceedings of the 20th International Conference on Production Research*.
- Yurtsever, T., E. Kutanoglu, and J. Johns. 2009. "Heuristic Based Scheduling System for Diffusion in Semiconductor Manufacturing." In *Proceedings of the 2009 Winter Simulation Conference*, 1677-1685.

AUTHOR BIOGRAPHIES

ANDREAS KLEMMT received his master's degree in mathematics in 2005 and Ph.D. in electrical engineering in 2011 from the Dresden University of Technology. He works as operations research and engineering expert at Infineon Dresden. His current research interests are capacity planning, production control, and simulation and optimization. His email is Andreas.Klemmt@infineon.com.

LARS MÖNCH is a Professor in the Department of Mathematics and Computer Science at the University of Hagen, Germany. He received a master's degree in applied mathematics in 1994, a Ph.D. in the same subject from the University of Göttingen, Germany in 1996, and a habilitation degree in information systems from the Technical University of Ilmenau in 2005. His current research interests are in simulation-based production control of semiconductor wafer fabrication facilities, production planning, and applied optimization and artificial intelligence applications in manufacturing, logistics, and service operations. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), SCS, INFORMS, and IIE. His email address is Lars.Moench@fernuni-hagen.de.