Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

A COMPARISON OF AN CP AND MIP APPROACH FOR SCHEDULING JOBS IN PRODUCTION AREAS WITH TIME CONSTRAINTS AND UNCERTAINTIES

Christian Maleck
Gottfried Nieke
Karlheinz Bock

Institute of Electronic Packaging Technology
Technische Universität Dresden
Helmholtzstr. 10
Dresden, 01069, GERMANY

Detlef Pabst

Factory Solutions
GLOBALFOUNDRIES Inc.
Malta, New York 12020, U.S.A.

Marcel Stehli

Factory Solutions
GLOBALFOUNDRIES Inc.
Dresden, 01109, GERMANY

ABSTRACT

This research is motivated by the expensive cost of scraps because of timelink misses in a semiconductor manufacturing line due to tool downs. A timelink is a time constraint between defined process steps. This paper presents a mixed integer programming model (MIP) and a constraint programming model (CP) with downscaled time constraints. With the assistance of the survival analysis, a safety value will be computed and included as a constant in the MIP and as a dynamic expression in the CP, to downscale the allowed time between two specific operations. The MIP and CP models are tested on a realistic production area example with different problem sizes. The quality of the solution and the performance of these two approaches are compared with each other. The test results show that the CP model outperforms the MIP and quickly finds much earlier usable schedules for large problem sizes.

1 INTRODUCTION

The semiconductor manufacturing is one of the most complex and challenging production environments for scheduling. There are a wide variety of different process steps, multiple products and routes, tool inhibits, lots with varying priorities, different quantities of wafers, dependencies between products and processes, lot release dates, unscheduled and scheduled tool downs, production areas with time constraints between consecutive or concatenated process steps and also concatenated production areas which have time constraints. Lots which violate time constraints often need to be scrapped, or must be costly reworked. The ongoing shrinkage of chip size coupled with an increased wafer density has resulted in more and tighter time constraints, due to unwanted processes like oxidation or particles that have a higher yield impact due to smaller feature sizes. Therefore, production areas with time constraints have a much higher cost impact and, because of the intensified cost pressure in semiconductor manufacturing, a greater attention is being paid to these production areas. That's the reason why there is a high demand on scheduling applications that consider timelink areas.

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

There is a great number of related work which treats the topic of scheduling. The characteristics for different objective functions and also different kinds of deterministic scheduling formulations can be found in Brucker (2007), Garey et al. (1976), Graham et al. (1979) and Jaehn and Pesch (2014). There are also a few papers which addresses scheduling with time constraints between consecutive process steps in semiconductor manufacturing. The different constellations of time constraints in a wafer fab are analyzed by Klemmt and Mönch (2012). They present a MIP model for flow shop scheduling problems with time constraints. In Klemmt (2012) detailed formulations for numerous problem descriptions in wafer fabs are presented. A two-stage lot scheduling MIP approach, for small problems sizes with time constraints, presented by Yu et al. (2013). In Cho et al. (2014) two MIP formulations with the objective to determine the best gate-keeping decisions, for areas with time constraints are presented and compared with each other. A gate-keeping decision decides when a lot is allowed to enter a timelink area. The MIP formulations presented in Cho et al. (2014) were not subjected to strict time conditions. They are based on a reward and sanction system. The advantage of this approach is that these MIPs are always solvable but the needed computation time to find good solutions is enormous. The needed computation time to find serviceable solution is the major problem for solving large scheduling problems. That's why it is necessary to investigate other solving approaches, like constraint programming (CP). CP is designed to satisfy constraints so that feasible solutions can be found faster. This is necessary for the scheduling of production areas. Basic ideas and how solutions are obtained by CP is described in Vilím et al. (2015) and Baptiste et al. (2012). CP allows to formulate dependencies in production areas and their mathematical generalizations. It is natural and transparent and outperforms some state-of-the-art MIP solver (Rossi et al. (2006)). Wang et al. (2015) compare a CP and a MIP approach for scheduling operating theaters. It is a highly constrained problem which is tested on real life data. It turns out that the CP solution outperforms the MIP while minimizing the makespan. The presented minimization of a weighted sum shows that the MIP works generally a bit better, but if the problem gets to complex, the MIP, in contrast to CP, doesn't find a solution. Nowadays CP is applied for many different types of optimization problems and most of them are scheduling problems. A different approach can be found at Topaloglu and Ozkarahan (2011). They observe a medical resident scheduling problem and compare it with a MIP approach. It results that the MIP only solves problems witch small instances.

In semiconductor industry only a few papers address scheduling with CP. In Ham et al. (2017) a diffusion process area with batch processing and in Ham (2018) a Litho area with batch processing are investigated. They compares a CP with a hybrid model of CP and MIP and also a CP and heuristic approach. It turns out that for batch processes the CP computes good solutions and the hybrid model of CP and MIP underperformed the MIP, but the best solutions are obtained by the combination of CP and the heuristic approach.

The observed problem in Malapert et al. (2012) is a similar batch processing area like in Ham (2018), but less complex. Here also CP outperforms the observed MIP and branch and price algorithm. The challenges to integrate such scheduling methods in a semiconductor fab, is highlighted in Klemmt et al. (2017).

The related work also shows disadvantages of a CP approach. The optimality of a found solution can only be proved very hard and the solution quality depends on the used solver (Vilím et al. (2015)).

In this research the related work of Maleck and Eckert (2017) and Maleck et al. (2017) will be extended by a comparison of a MIP and CP approach. The approaches compare the solution quality of a complex production environment on the basis of three different objective functions, on three different problem sizes.

First the observed scheduling problem with time constraints is described. In subsection 3.1 the used objective functions and the reliability factors are presented. Then the subsections 3.2 and 3.3 introduces the observed MIP and CP model. The test environment and its results are presented in section 4. Finally a short conclusion and outlook is given in section 5.

2 PROBLEM DESCRIPTION

This section describes the observed general scheduling problem. It represents a simplified production area in a semiconductor fab with time constraints. This job shop problem has a set of lots, where a sequence of process steps has to be executed, on a set of single-processing tools. These process steps will also be referred as jobs in this paper. The time constraints are defined between consecutive or concatenated jobs, whereby each lot can have multiple time constraints, also called timelink areas.

The mathematical description is as followed, there is a set of lots $L := \{L_1, \ldots, L_n\}$, $n \in \mathbb{N}$ that has to be scheduled after a time $t_0 \in \mathbb{N}$, which represents the earliest possible scheduling time. Each lot $l \in L$ represents a set of $qty_l \in \mathbb{N}$ wafers, with a maximal quantity of 25. Lots have a priority weight $\omega_l \in \mathbb{N}$ and each lot l has its own sequence of jobs $J_l := \{O_1, ..., O_{o_l}\}$, $o_l \in \mathbb{N}$ that has to be executed. Each job $O_{l,o}$, $o \in J_l$ is associated with exactly one process $g(l,o) : J_l \to P$, where $P := \{P_1, \ldots, P_g\}$, $g \in \mathbb{N}$ is a set of processes.

Furthermore, there is a set of available tools $T := \{T_1, \ldots, T_m\}$, $m \in \mathbb{N}$. Each tool is a single-tool, which means that only one job can be performed at any given time on a given tool. In this problem, each process $k \in P$ has its own work center $W_k^{all} \subseteq T$, $W_k^{all} \neq \emptyset$, $\dot\bigcup_{k \in P} W_k^{all} = T$ with $t_k = |W_k^{all}|$ qualified tools for this process step. These tools must not be necessary identical. Each job $O_{l,o}$ must be performed on one tool t in the associated work center $t \in W_k^l$, where $W_k^l \subset W_k^{all}$ is a subset of allowed tools for a lot l for a process $k = g(l,o)$.

The processing time of one Wafer $p_{k,t}$ for each operation depends on the given process $k = g(l,o)$ and the working tool t. The entire process time of a job $O_{l,o}$ depends also on the quantity of wafers and is defined as

$$p_{k,t}^l := p_{k,t} \cdot qty_l. \tag{1}$$

Each lot $l \in L$ receives a release date r_l, which means that it can be scheduled at any point in the future, after the release date.

Furthermore, there exists a constant transport time $t_{transport} \in \mathbb{N}$ between two random tools. It is assumed that $t_{transport} \in \mathbb{N}$ is identical for all possible tool combinations

For some lots l there are time constraints $t_{(l,o,q)} > 0$ between defined consecutive or concatenated jobs $o, q \in J_l$, $o < q$, which can be formulated as

$$s_{l,q} \leq s_{l,o} + p_{g(l,o),t}^l + t_{(l,o,q)} \qquad l \in L, \tag{2}$$

where $s_{l,o}$ is the scheduled start time of job $O_{l,o}$ and $t \in W_{g(l,o)}^l$ is the working tool. An example of a Gantt diagram for a lot with a feasible solution, is shown in Figure 1.

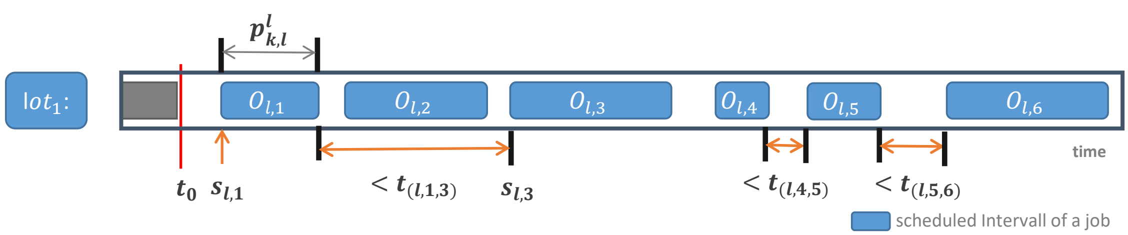

Figure 1: Simplified example of a Gantt diagram of a schedule with possible timelinks of a lot $l \in L$.

In relation to the SEMI E10 standard, which is described at Thomas Pomorski (2009), six basic equipment states were established. These six equipment states are assigned to basic up or down conditions for the survival analysis, similar to the approach in Maleck and Eckert (2017). The given set of tools are all up, but underlies there specific reliability.

3 MODEL FORMULATIONS

The following notations are used for the CP and MIP formulation:

n	$\in \mathbb{N}$	number of lots
m	$\in \mathbb{N}$	number of tools
g	$\in \mathbb{N}$	number of processes
T	$:= \{T_1, ..., T_m\}$	set of tools
L	$:= \{L_1, ..., L_n\}$	set of jobs
P	$:= \{P_1, ..., P_g\}$	set of processes
qty_l	$\in \{1, 2, .., 25\}$	quantity of wafers of lot $l \in L$
o_l	$\in \mathbb{N}$	number of jobs of lot $l \in L$
J_l	$:= \{O_1, ..., O_{o_l}\}$	set of jobs of lot $l \in L$
j_l	$\in J_l$	latest processed job of a lot $l \in L$
$O_{l,o}$	$\in J_l$	o-th job of lot $l \in L$
$g(l,o)$	$: J_l \to P$	process of job $O_{l,o}$ for $l \in L$
W_k^l	$\subseteq W_k \subseteq M$	set of allowed tools for lot l at process $k = g(l,o)$
$M_{k,t}$	$\in W_k$	t-th tool of W_k for $k = g(l,o)$
$p_{k,t}$	$\in \mathbb{N}$	process time per wafer of process $k = g(l,o)$ at tool $t \in T$
$p_{k,t}^l$	$\in \mathbb{N}$	process time of a job $O_{l,o}$ at tool $t \in W_k^t$ with $k = g(l,o)$
r_l	$\in \mathbb{N}$	release date of lot l
t_0	$\in \mathbb{N}$	earliest possible scheduling time
T_l	$:= \{(o,q) \mid 1 < o < q \leq o_l\}$	set of timelink areas of lot l with $o, q \in J_l$, $o < q$
T^l	$:= \{(o,q) \mid o < j_l \leq q \leq o_l\}$	entered timelink area of lot l with $o, q \in J_l$, $o < q$
$t_{(l,o)}$	$\in \mathbb{N}$, $(l,o) \in T^l$	enter time of lot l in a timelink area $(o,q) \in T^l$
$t_{(l,o,q)}$	$\in \mathbb{N}$, $(o,q) \in T_l$	timelink between job $O_{l,o}$ and job $O_{l,q}$, $l \in L$
ω_l	$\in \mathbb{N}$	weight of lot l
$t_{transport}$	$\in \mathbb{N}$	mean transport time between tools
K	$\in \mathbb{N}$	a large positive number

3.1 Objective Function

To investigate, how the performance and solution quality of the MIP and CP model depend on different optimization targets, three different objective functions were observed. The first objective function is the minimization of the makespan which is defined as the maximal cycle time of all lots $l \in L$ of a schedule. Whereby the cycle time is the completion time $C_l \in \mathbb{N}$ of a lot minus its release date r_l. It is formulated as

$$z_1 = \max_{l \in L} \{C_l - r_l\} \ \rightarrow \ min \, . \tag{3}$$

As second objective function, the normalized weighted cycle time sum, which is abbreviated as *NWS*, was used

$$z_2 = \sum_{l \in L} \left(\frac{\omega_l}{qty_l} \cdot (C_l - r_l) \right) \ \rightarrow \ min \, . \tag{4}$$

Finally, to get a more practicable and a balanced schedule for semiconductor manufacturing, a weighted and normalized objective function is used and abbreviated as *BNWS*. It contains the sum of the normalized cycle time sum dependent on the number of its jobs and the sum over the time in timelink areas of each lot, which weighted by the lot priority and normalized by its given quantity of wafers. These observed objective is defined as

$$z_3 = \sum_{l \in L} \left(\frac{\omega_l}{qty_l} \cdot \left(\frac{(C_l - r_l)}{|J_l|} + \sum_{(o,q) \in T_l : o > j_l} \left(s_{l,q} - e_{l,o} \right) + \sum_{(o,q) \in T^l} \left(s_{l,q} - t_{l,o} \right) \right) \right) \rightarrow \; min \tag{5}$$

whereby $e_{l,o}$ represents the end time of job $O_{l,o}$, $s_{l,q}$ represent the start time of job $O_{l,q}$ and $o < q$.

3.1.1 Integration of Reliability Factors

To get more robust schedules, as in Maleck et al. (2017), reliability factors are integrated in the CP and MIP model. Similar to the definitions in Maleck and Eckert (2017) and Maleck et al. (2017), the probability that a tool will stay up within the interval $[t_0, t_0 + \Delta t]$, $\Delta t > 0$ can be formulated with the help of the exponential distribution. So, if in a production area a tool is up at current time t_0, then the assumed availability of a tool for time period Δt is

$$\nu_t(t_0, t_0 + \Delta t) := \mathbb{P}[T > t_0 + \Delta t | T > t_0] = \frac{e^{-\lambda_t (t_0 + \Delta t)}}{e^{-\lambda_t t_0}} = e^{-\lambda_t \Delta t}. \tag{6}$$

where T is a single random variable that is continuous and non-negative. It represents the lifetime of a tool and the expectation value $1/\lambda_t$ is defined as the specific *(Mean Time Before Failures) MTBF* for tool $t \in T$. For basic details, see Wienke (2010) or Liu (2012). Since $\nu_l(t_0, t_0 + \Delta t)$ does not depend on t_0 for the exponential distribution, it is notated as $\nu_t(\Delta t)$.

3.2 MIP Formulation

The developed MIP has four types of decision variables:

C_l	$\in \mathbb{N}$	cycle time of lot $l \in L$
$w_{l,o,t}$	$\in \{0,1\}$	assignment from job $O_{l,o}$ to tool $t \in W_k^l$, $k = g(l,o)$
$s_{l,o}$	$\in \{\max\{r_l, t_0\}, ..., K\}$	starting time of job $O_{l,o}$
$x_{O_{l,o}, O_{h,q}}$	$\in \{0,1\}$	1 if job $O_{l,o}$ is scheduled before $O_{h,q}$, otherwise 0 and $(h \neq l)$

To integrate the reliability of the tools, a mean availability factor κ_k of W_k^l for process $k = g(l,o)$, $l \in L$, $o \in J_l$ is defined as

$$\kappa_k := \frac{1}{|W_k^l|} \cdot \sum_{t \in W_k^l} \nu_t(\Delta t) \,, \tag{7}$$

whereby ν_t is the with (6) calculated availability of a tool $t \in T$ for the interval $[t_0, t_0 + \Delta t]$. The range $\Delta t \geq 0$ is constant and given. The downscaled time constraint $t^{\kappa}_{(l,o,q)}$ with $(l,o,q) \in T$ and $k = g(l,o)$ for all jobs in front of a timelink area is

$$t^{\kappa}_{(l,o,q)} := t_{(l,o,q)} \cdot \prod_{i \in o+1,..,q} \kappa_{g(l,i)} \qquad \forall l \in L, \, \forall (o,q) \in T_l, \; j_l < o \,, \tag{8}$$

where $\prod_{i \in o+1,..,q} \kappa_{g(l,i)}$ is the mean survival probability of consecutive or concatenated process steps.

The objective functions (3), (4) and (5) with $e_{l,o} := s_{l,o} + \sum_{t \in W_k^l} \left(w_{l,o,t} \cdot p_{k,t}^l \right)$ are minimized subject to

$$s_{l,o_l} + w_{l,o_l,t} \cdot p_{k,t}^l \leq C_l \qquad \forall l \in L,\ \forall t \in W_k^l,\ k = g(l,o_l) \tag{9}$$

$$s_{l,o} + w_{l,o,t} \cdot p_{k,t}^l + t_{transport} \leq s_{l,o+1} \qquad \forall l \in L,\ \forall o \in J_l \backslash \{O_{l,o_l}\},\ j_l < o,\ k = g(l,o) \tag{10}$$

$$\sum_{t \in W_k} w_{l,o,t} = 1 \qquad \forall l \in L,\ \forall o \in J_l,\ j_l < o\ k = g(l,o) \tag{11}$$

$$s_{l,q} - e_{l,o} \leq t_{(l,o,q)}^{\kappa} \qquad \forall l \in L,\ (o,q) \in T_l,\ j_l < o \tag{12}$$

$$s_{l,q} - t_{l,o} \leq t_{(l,o,q)} \qquad \forall l \in L,\ (o,q) \in T^l \tag{13}$$

$$K(w_{l,o,t} - x_{O_{l,o},O_{h,q}} - 1) + s_{h,q} + p_{g(h,q),t}^h \cdot w_{h,q,t} \leq s_{h,q} \qquad \begin{array}{l} \forall l,h \in L,\ l < h,\ \forall t \in W_{g(l,o)}^l \cap W_{g(h,q)}^h, \\ \forall o \in J_l,\ o > j_l,\ \forall q \in J_h,\ q > j_h \end{array} \tag{14}$$

$$K(w_{h,q,t} + x_{O_{l,o},O_{h,q}} - 2) + s_{l,o} + p_{g(l,o),t}^l \cdot w_{l,o,t} \leq s_{l,o} \qquad \begin{array}{l} \forall l,h \in L,\ l < h,\ \forall t \in W_{g(l,o)}^l \cap W_{g(h,q)}^h, \\ \forall o \in J_l,\ o > j_l,\ \forall q \in J_h,\ q > j_h \end{array} \tag{15}$$

Due to the range of $s_{l,o} \in \{\max\{r_l, t_0\}, ..., K\}$ it is ensured, that each job is scheduled after t_0 and the lot release date r_l. The Constraints (9) and (10) restricts the objective function and consider the sequence of the process steps of a lot. Equation (11) ensures that each job is executed exactly once by a given lot. Inequalities (14) and (15) assure that only one lot can be scheduled on a given tool at a given time. Constraint (12) and (13) represent the time constraints between consecutive or concatenated process steps.

3.3 CP Formulation

An efficient CP model needs to exploit the features of the used CP Optimizer. The IBM ILOG CP Optimizer provides specialized variables and constraints, which are used to describe the CP model. For further details of these specialized features please refer to IBM (2017) or the papers of Laborie and Rogerie (2008) and Laborie (2009). We formulate a parallel singe-processing tool problem with timelink areas into a CP as followed. First, this CP has three types of decision variables.

interval	$job_{l,o} \in [\max\{r_l, t_0\}, \infty)$	$\forall l \in L,\ \forall o \in J_l$	interval of a job
interval	$job_{l,o}^t$ *optional* $\in [\max\{r_l, t_0\}, \infty)$ size $p_{k,t}^l$	$\forall l \in L,\ \forall o \in J_l,\ \forall t \in W_k^l$	job to tool interval
variable	$\nu(job_{l,o}) \in (0.5, 1]$	$\forall l \in L,\ \forall o \in J_l$	survival probability of a job

Because of the range definition of the interval $job_{l,o}$ it is ensured, that each job is scheduled after t_0 and the lot release date r_l.

The reliability factors for tools in the CP model, are dynamical and the range Δt depends on the the end time $endOf(job_{l,o}^t)$ of a job-tool interval $job_{l,o}^t$ and is defined as

$$\nu_t(job_{l,o}^t) = e^{\left(-\lambda_t \cdot (endOf(job_{l,o}^t) - t_0)\right)}, \tag{16}$$

where $\lambda_t = \frac{1}{MTBF_t}$, $l \in L$, $o \in J_l$ and $t \in W_k^l$. The downscaled time constraint $t_{(l,o,q)}^{\kappa}$ with $(l,o,q) \in T$ and $k = g(l,o)$ for all jobs in front of timelink is

$$t_{(l,o,q)}^{\kappa} = t_{(l,o,q)} \cdot \prod_{i \in o+1,..,q} \nu(job_{l,o}), \tag{17}$$

where $\prod_{i \in o+1,..,q} \nu(job_{l,o})$ is the survival probability of consecutive or concatenated process steps.

The objective functions (3), (4) and (5) with $s_{l,o} := startOf(job_{l,o})$ and $e_{l,o} := endOf(job_{l,o})$ are minimized subject to the constraints

$$endBeforeStart\left(job_{l,o}, job_{l,o+1}, t_{transport}\right) \qquad \forall l \in L,\ \forall o \in J_l : j_l < o < last(J_l) \tag{18}$$

$$alternative\left(job_{l,o}, \{job^t_{l,o}\}_{\forall t \in W^l_k}\right) \qquad \forall l \in L,\ \forall o \in J_l : j_l < o,\ k = g(l,o) \tag{19}$$

$$noOverlap\left(\{job^t_{l,o}\}_{(\forall l \in L, \forall o \in J_l : t \in W^l_k)}\right) \qquad \forall t \in T,\ k = g(l,o) \tag{20}$$

$$lengthOf\left(job^t_{l,o}\right) > 0 \Rightarrow \nu(job_{l,o}) == \nu(job^t_{l,o}) \qquad \forall l \in L,\ \forall o \in J_l : j_l < o,\ \forall t \in W^l_k,\ k = g(l,o) \tag{21}$$

$$startOf(job_{l,q}) - endOf(job_{l,o}) \leq t^{\kappa}_{(l,o,q)} \qquad \forall l \in L,\ (o,q) \in T_l,\ j_l < o \tag{22}$$

$$startOf(job_{l,q}) - t_{l,o} \leq t_{(l,o,q)} \qquad \forall l \in L,\ (o,q) \in T^l \tag{23}$$

With the help of the predefined function *endBeforeStart()* constraint (18) assures that the process steps are executed in the right order, with respected to the transport time. Constraint (19) associates the job interval with the job-tool interval and together with (20) they ensure that each job is executed exactly once by a given tool and that only one job can be scheduled on one tool at a given time. If a job is scheduled on a tool the constraint (21) determines the survival probability of a job. Constraints (22) and (23) represent the time constraints between consecutive or concatenated process steps.

4 TEST AND RESULTS

In this section, we describe the test environment and compare the performance and solution quality of the two models. The MIP and CP models are implemented with IBM OPL and solved with the IBM ILOG CP and CPLEX 12.8.0.0 solver. The results were computed on exact one thread on a cluster computer with an AMD Ryzen Threadripper 1950X @ 3,4 Ghz and 32Gb RAM.

4.1 Test Environment

The tested production area example is motivated by a semiconductor fab. The CP and MIP approaches with the three different objective functions, are compared with the scheduling results of this production area, which is tested for three different problem sizes. The problem sizes depend on the quantities of the lots. This includes the number of jobs that has to be scheduled per lot and how many tools can operate a job. The smallest problem size is defined by a 5 lot problem with a sum of 21 jobs with altogether 50 job-tool assignments. The other scheduling problems contain 15 lots with 62 jobs and a total of 139 job-tool assignments. At last a 30 lot, 118 jobs problem with 266 job-tool assignments is tested. Additionally each lot can have different prioritys, release dates and quantities of wafers. The detailed lot and job definitions can be found in Appendix A at table 1. Also each lot can have one or more time constraint definitions between consecutive or concatenated jobs which is shown in Appendix A at Table 3. The tested examples have 16 available tools for 5 process steps, whereby the process time for a wafer on each tool, which can execute this process, not necessarily have to be identical. The detailed tool and process definitions are shown in the Appendix A at Table 2. Every test environment is solved by the MIP and CP model up to 16 times with the following different time limits: 1s, 3s, 5s, 10s, 20s, 30s, 60s, 90s, 2min, 5min, 10min, 30min, 1h, 2h, 4h and 6h.

4.2 Computational results

In this subsection the computed results will be presented and compared with each other. First the MIP approach found, for the smallest problem size of 5 lots, with all objective functions, straightway found the optimal results and stopped the solving process within one second. For the makespan (3) the optimal solution was 10630, for the normalized weighted sum *NWS* (4) 969.2. The objective value of the balanced normalized weighted sum (*BNWS*) (5) was 1205.37. The CP approach only found for the makespan immediately the optimal solution and stopped the solving process. For the *NWS*, CP also delivered the optimal solution, but was not able to prove it. The CP optimal gap was within one second 0.04% and within three seconds 0.007%. The CP underperformed with the *BNWS* (5) as objective function. The first solution within one second delivered a value of 1237.03 with a gap of 0.065% and within three seconds a value of 1213.72 with a gap of 0.047%. Even within 4 hours the CP approach doesn't found a better solution value as 1212.97 with a gap of 0.046%. The performance of the CP and MIP approaches with the makespan objective function (3) are shown for the 15 lot scheduling problem in Figure 2 and for the 30 lot scheduling problem in Figure 3. The dashed line depicts the progress of the objective gap (in percent) regarding to the MIP and CP solutions which were found. The thin line, without points, depicts the best possible found objective value, which is the lower bound of the optimization problem. The stronger lines describe the best found solution values within a given calculation time. The MIP approach found, for the 15 lot problem, the optimal solution within 30 minutes but underperformed in the first 30 seconds in comparison to the CP approach. Figure 2 provides also the disadvantages of the CP solver. That is that the CP approach found the optional solution, but was not able to prove it. For the biggest tested problem size, the CP outperforms the MIP. The MIP found the first solution after 30 seconds whereas the CP found the first solution within one second and a much better solution within 30 seconds.

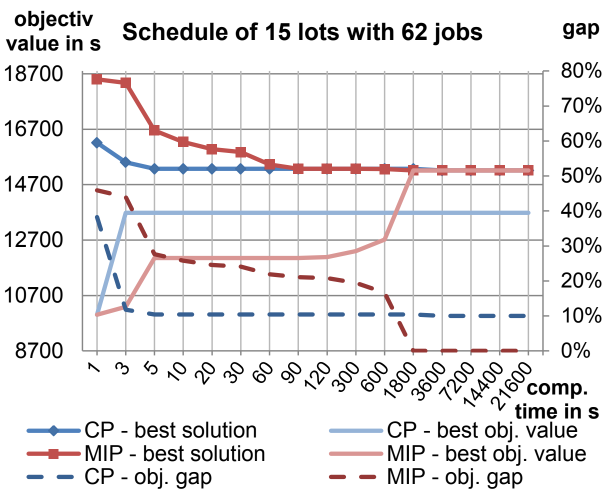

Figure 2: Computational results on the minimization of the makespan (3) of a 15 lot problem.

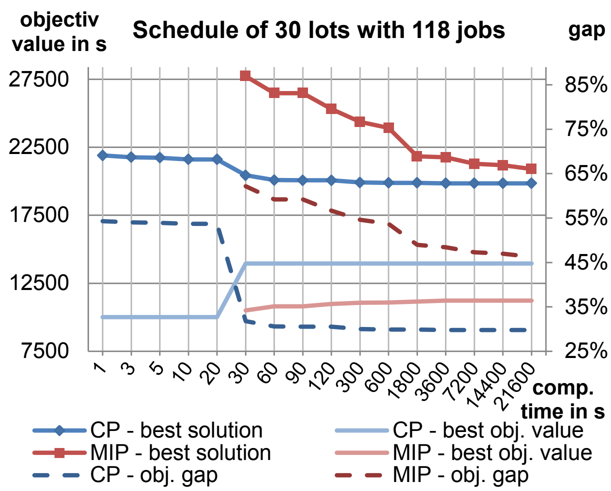

Figure 3: Computational results on the minimization of the makespan (3) of a 30 lot problem.

The computational results for the minimization of the *NWS* (4) are shown for the 15 and 30 lot scheduling problem in Figure 4 and 5. The 15 lot problem delivers for the CP and MIP approach very similar results. For the larger problem size the solution quality of the MIP is shifted to the right, in comparison to the CP solution. The results of the *BNWS* (5) are presented in Figure 6 and 7. It is shown that the CP found within the first seconds solutions. These solutions are better than the solutions found by the MIP approach. But within five seconds the MIP delivers for the 15 lot scheduling problem similar solutions as the CP. For the larger problem size, the MIP found its first solution only after 30 seconds, but this solution is a little bit better than the best solution found with the whole CP approach. Additionally within the next given computation times the MIP outperforms the CP. Based on this results, it is suggested to combine the CP and MIP approach, whereby in the first stage the CP should compute a start solution and in the second stage the MIP should use the objective *BNWS*. A combination of CP, with a computation time of 5 seconds, and MIP model, with a computation time at minimum 5 seconds, is also shown in Figure 7. It turns out that the combination of CP and MIP outperforms the single CP and MIP models. After 10 seconds it computes a better solution than the MIP alone only after 90 seconds.

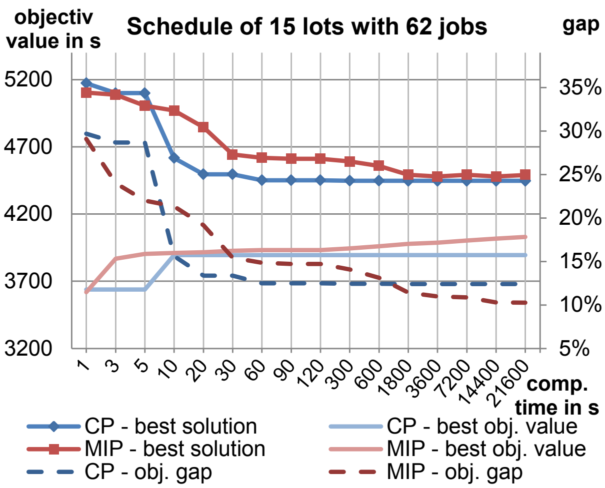

Figure 4: Computational results on the minimization of the *NWS* (4) of a 15 lot problem.

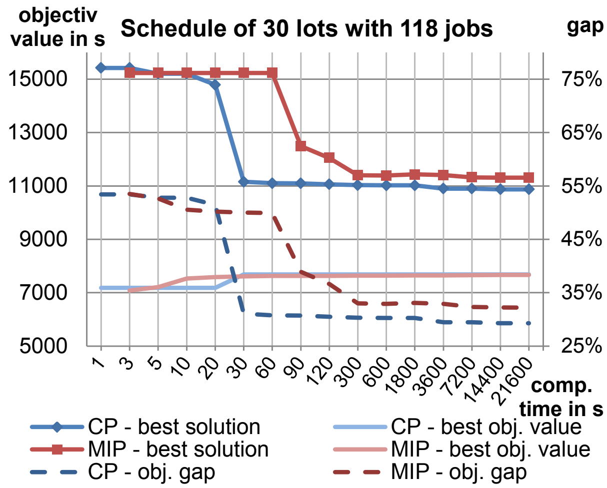

Figure 5: Computational results on the minimization of the *NWS* (4) of a 30 lot problem.

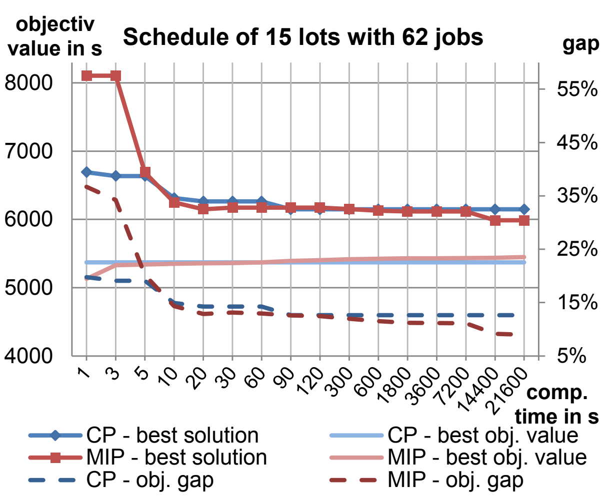

Figure 6: Computational results on the minimization of the *BNWS* (5) of a 15 lot problem.

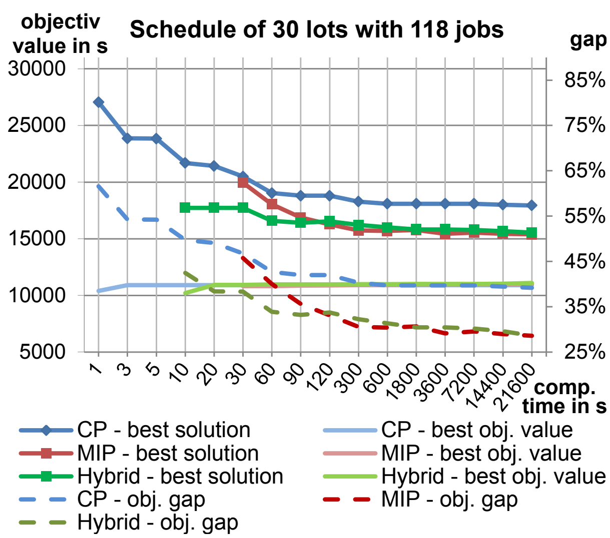

Figure 7: Computational results of the minimization of the *BNWS* (5) of a 30 lot problem.

5 CONCLUSION AND OUTLOOK

In this paper, a MIP and CP approach with time constraints was presented which was tested with three different objective functions for three different problem sizes. It turns out that the CP outperforms the MIP with the makespan objective function. For larger problem sizes the CP model finds immediately solutions in contrast to the MIP approach. The test results have been shown that the MIP can prove the optimality much faster than the CP. They also showed that the MIP computes better solutions for more complex objective functions which consist weighted sums, under the restriction that there is enough computation time available. It is advisable for large scheduling problems with makespan optimization problem to use a CP approach or for a weighted sum optimization problem to use the presented combination of CP and MIP. For further investigations the CP and MIP combination should be tested on a more realistic fab environment with over 300 lots containing more than 1000 jobs. Additionally the robustness of such schedules and a long-term simulations study, based on real production environments, should be investigated. For this case, the hybrid model, which is presented in Maleck et al. (2017), seems to be suitable.

6 ACKNOWLEDGMENTS

This work was part of the "'Responsive Fab"' project, co-funded by grants from the European Union (EFRE) and the Free State of Saxony (SAB). (project number 100259418).

A APPENDIX

Table 1: Set of jobs $o \in J_l$ of each lot $l \in L$.

lot	j_l	qty_j	r_l	ω_j	sequence id: $(k,\ W_k^l)$, $k = g(l,o)$				
1	0	22	0	1	{ 1: (1 {1 2 3}),	2: (3 {5 6}),	3: (5 {9}),	4: (6 {12 13 14 15 16}) }	
2	0	25	20	2	{ 1: (2 {3 4}),	2: (3 {5}),	3: (5 {9 10}),	4: (6 {12 13 14 15 16}) }	
3	0	25	0	2	{ 1: (1 {1 3}),	2: (3 {5 6}),	3: (4 {7}),	4: (5 {9 11}),	5: (7 {12 13}) }
4	0	14	30	4	{ 1: (1 {1 2 3 4}),	2: (4 {8}),	3: (5 {10 11}),	4: (6 {12 13 14 15 16}) }	
5	0	25	0	2	{ 1: (3 {6}),	2: (4 {7 8}),	3: (5 {10 11}),	4: (8 {14 15 16}) }	
6	0	3	85	5	{ 1: (2 {1 2 3}),	2: (3 {5 6}),	3: (5 {9}),	4: (8 {14 15 16}) }	
7	0	15	20	2	{ 1: (1 {3 4}),	2: (3 {5}),	3: (5 {9 10}),	4: (6 {12 13 14 15}) }	
8	0	25	0	4	{ 1: (2 {1 3}),	2: (3 {5 6}),	3: (4 {7}),	4: (5 {9 11}),	5: (8 {14 15}) }
9	0	25	10	2	{ 1: (1 {1 2 3 4}),	2: (4 {8}),	3: (5 {10 11}),	4: (6 {12 13 14 15 16}) }	
10	0	23	10	2	{ 1: (3 {6}),	2: (4 {7 8}),	3: (5 {10 11}),	4: (7 {12 13}) }	
11	0	25	0	2	{ 1: (1 {1 2 3}),	2: (3 {5 6}),	3: (5 {9}),	4: (7 {12 13}) }	
12	0	25	90	2	{ 1: (1 {3 4}),	2: (3 {5}),	3: (5 {9 10}),	4: (6 {12 13 14 15 16}) }	
13	0	23	0	2	{ 1: (2 {1 2 3}),	2: (3 {5 6}),	3: (5 {9}),	4: (9 {15 16}) }	
14	0	14	200	6	{ 1: (2 {3 4}),	2: (3 {5}),	3: (5 {9 10}),	4: (8 {15 16}) }	
15	0	25	0	2	{ 1: (3 {6}),	2: (4 {7 8}),	3: (5 {10 11}),	4: (6 {12 13 14 15 16}) }	
16	0	25	30	2	{ 1: (1 {1 3}),	2: (3 {5 6}),	3: (4 {7}),	4: (5 {9 11}),	5: (9 {14 15 16}) }
17	0	23	0	2	{ 1: (2 {1 3}),	2: (3 {5 6}),	3: (4 {7}),	4: (5 {9 11}),	5: (8 {14 15 16}) }
18	0	23	10	4	{ 1: (2 {3 4}),	2: (3 {5}),	3: (5 {9 10}),	4: (7 {12 13}) }	
19	0	16	15	4	{ 1: (1 {1 2 3 4}),	2: (4 {8}),	3: (5 {10 11}),	4: (6 {12 13 14 15 16}) }	
20	1	14	10	4	{ 1: (2 {1 3}),	2: (3 {5 6}),	3: (4 {7}),	4: (5 {9 11}),	5: (6 {12 13 14 15 16}) }
21	2	25	5	2	{ 1: (1 {1 2 3}),	2: (5 {9 10}),	3: (4 {8}),	4: (3 {5 6}),	5: (6 {12 13 14 15 16}) }
22	1	25	90	2	{ 1: (2 {1 3}),	2: (5 {9 10}),	3: (4 {8}),	4: (3 {5 6}),	5: (9 {14 15}) }
23	2	23	0	2	{ 1: (2 {1 2 3 4}),	2: (5 {9 11}),	3: (4 {7}),	4: (3 {5 6}),	5: (8 {14 15}) }
24	2	19	200	6	{ 1: (2 {1 2 3}),	2: (5 {9 11}),	3: (4 {7 8}),	4: (3 {5 6}),	5: (6 {13 14 15 16}) }
25	1	23	30	2	{ 1: (2 {1 2}),	2: (3 {5 6}),	3: (4 {7}),	4: (6 {12 13 14 15}) }	
26	2	15	30	2	{ 1: (2 {1 3}),	2: (3 {5 6}),	3: (4 {7 8}),	4: (7 {12 13}) }	
27	1	23	0	2	{ 1: (2 {1 3 4}),	2: (3 {5 6}),	3: (4 {7 8}),	4: (6 {12 13 14 15}) }	
28	1	23	10	4	{ 1: (2 {1 4}),	2: (3 {5 6}),	3: (4 {7 8}),	4: (5 {9 11}),	5: (8 {14 15 16}) }
29	1	19	15	4	{ 1: (2 {1 2 4}),	2: (3 {5 6}),	3: (4 {7 8}),	4: (5 {9 11}),	5: (8 {14 15 16}) }
30	2	14	10	4	{ 1: (2 {1 3}),	2: (3 {5 6}),	3: (4 {7 8}),	4: (5 {9 11}),	5: (6 {12 13 14 15 16}) }

Table 2: Tool $t \in T$ definitions and set of their process times.

t	$MTBF$	$\{(k:\ p_{k,t}\ \ in\ \ s),\ k \in P\}$		t	$MTBF$	$\{(k:\ p_{k,t}\ \ in\ \ s),\ k \in P\}$		
1	170842	{(1: 70),	(2: 90)}	9	129253	{(5: 90)}		
2	129253	{(1: 85),	(2: 55)}	10	140808	{(5: 110)}		
3	170842	{(1: 75),	(2: 62)}	11	154462	{(5: 100)}		
4	154462	{(1: 58),	(2: 95)}	12	215875	{(6: 110),	(7: 110)}	
5	154462	{(3: 50)}		13	154462	{(6: 100),	(7: 100)}	
6	190859	{(3: 80)}		14	350923	{(6: 94),	(8: 70),	(9: 89)}
7	350923	{(4: 40)}		15	215875	{(6: 100),	(8: 87),	(9: 110)}
8	248034	{(4: 50)}		16	154462	{(6: 80),	(8: 107),	(9: 100)}

Table 3: Set of timelink areas of each lot $l \in L$.

l	$\{(o \rightarrow q : t_{l,o,q}\ \ in\ \ s),\ \forall (o,q) \in T^l\}$				l	$\{(o \rightarrow q : t_{l,o,q}\ \ in\ \ s),\ \forall (o,q) \in T^l\}$			
1	{(1 → 3: 8600)}				16	{(1 → 3: 9600),	(3 → 4: 9600),	(4 → 5: 15600)}	
2	{(1 → 2: 6600),	(2 → 3: 6600)}			17	{(1 → 3: 9600),	(3 → 4: 8600),	(4 → 5: 15600)}	
3	{(1 → 2: 6600),	(2 → 3: 5600),	(3 → 4: 5600),	(4 → 5: 15600)}	18	{(1 → 3: 8600),	(3 → 4: 10600)}		
4	{(2 → 3: 16600),	(3 → 4: 15600)}			19	{(1 → 3: 9600),	(3 → 4: 86400),		
5	{(1 → 2: 5600),	(2 → 3: 4400)}			20	{(1 → 2: 5600),	(2 → 3: 4600),	(3 → 4: 6600),	(4 → 5: 15600)}
6	{(1 → 3: 8600),	(3 → 4: 16600)}			21	{(1 → 2: 6600),	(2 → 3: 8600),	(3 → 5: 86400)}	
7	{(1 → 2: 9600),	(2 → 3: 5600)}			22	{(1 → 2: 3600),	(2 → 3: 8600),	(3 → 5: 86400)}	
8	{(1 → 2: 6600),	(2 → 3: 8600),	(3 → 4: 6600),	(4 → 5: 15600)}	23	{(1 → 2: 5600),	(2 → 3: 8600),	(3 → 5: 16600)}	
9	{(2 → 3: 9600),	(3 → 4: 86400)}			24	{(1 → 2: 6600),	(2 → 3: 8600),	(3 → 5: 18600)}	
10	{(1 → 2: 8900),	(3 → 4: 86400)}			25	{(1 → 2: 3600),	(2 → 3: 8600),	(3 → 4: 8600)}	
11	{(1 → 2: 5600),	(3 → 4: 86400)}			26	{(1 → 2: 3600),	(2 → 3: 8600),	(3 → 4: 8600)}	
12	{(1 → 2: 4900),	(3 → 4: 86400)}			27	{(1 → 2: 3600),	(2 → 3: 9600),	(3 → 4: 8600)}	
13	{(1 → 2: 5600),	(2 → 3: 4600),	(3 → 4: 86400)}		28	{(1 → 2: 3600),	(2 → 3: 6600),	(3 → 5: 86400)}	
14	{(1 → 2: 6600),	(2 → 3: 7600),	(3 → 4: 86400)}		29	{(1 → 2: 3600),	(2 → 3: 8600),	(3 → 5: 86400)}	
15	{(1 → 2: 8600),	(2 → 3: 8600),	(3 → 4: 86400)}		30	{(1 → 2: 3600),	(2 → 3: 6600),	(3 → 5: 86400)}	

REFERENCES

Baptiste, P., C. Le Pape, and W. P. Nuijten. 2012. *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. International Series in Operations Research & Management Science. New York, US: Springer Science and Business Media.

Brucker, P. 2007. *Scheduling Algorithms*. 5. ed. Heidelberg: Springer.

Cho, L., H. M. Park, J. K. Ryan, T. C. Sharkey, C. Jung, and D. Pabst. 2014. "Production Scheduling with Queue-time Constraints: Alternative Formulations". In *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*, edited by Y. Guan and H. Liao, 282–291. Montral, QC, Canada: Institute of Industrial and Systems Engineers (IISE).

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. "The Complexity of Flowshop and Jobshop Scheduling". *Mathematics of Operations Research* 1(2):117–129.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. R. Kan. 1979. "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey". *Annals of Discrete Mathematics* 5:287–326.

Ham, A. 2018. "Scheduling of Dual Resource Constrained Lithography Production: Using CP and MIP/CP". *IEEE Transactions on Semiconductor Manufacturing* 31(1):52–61.

Ham, A., J. Fowler, and E. Cakici. 2017. "Constraint Programming Approach for Scheduling Jobs with Release Times, Non-identical Sizes, and Incompatible Families on Parallel Batching Machines". *IEEE Transactions on Semiconductor Manufacturing* 30(4):500–507.

IBM 2017. *IBM ILOG CPLEX Optimization Studio OPL Language Reference Manual*. Version 12 Release 8 ed. IBM.

Jaehn, F., and E. Pesch. 2014. *Ablaufplanung: Einführung in Scheduling*. Heidelberg, Germany: Springer.

Klemmt, A. 2012. *Ablaufplanung in der Halbleiter- und Elektronikproduktion: Hybride Optimierungsverfahren und Dekompositionstechniken*. Wiesbaden, Germany: Springer Vieweg.

Klemmt, A., J. Kutschke, and C. Schubert. 2017. "From dispatching to scheduling: Challenges in integrating a generic optimization platform into semiconductor shop floor execution". In *Proceedings of the 2017 Winter Simulation Conference (WSC)*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 3691–3702. Piscataway, NJ: IEEE.

Klemmt, A., and L. Mönch. 2012. "Scheduling Jobs with Time constraints Between Consecutive Process Steps in Semiconductor Manufacturing". In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. Uhrmacher, 2173–2182. Piscataway, NJ: IEEE.

Laborie, P. 2009. "IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems". In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, edited by W.-J. van Hoeve and J. N. Hooker, 148–162. Heidelberg, Germany: Springer.

Laborie, P., and J. Rogerie. 2008. "Reasoning with Conditional Time-Intervals". In *Proceedings of the Twenty-First International FLAIRS Conference*, 555–560: AAAI.

Liu, X. 2012. *Survival analysis: models and applications*. John Wiley & Sons.

Malapert, A., C. Guret, and L.-M. Rousseau. 2012. "A constraint programming approach for a batch processing problem with non-identical job sizes". *European Journal of Operational Research* 221(3):533–545.

Maleck, C., and T. Eckert. 2017. "A comparison of control methods for production areas with time constraints and tool interruptions in semiconductor manufacturing". In *40th Int. Spring Seminar on Electronics Technology (ISSE)*. Sofia, Bulgaria: IEEE.

Maleck, C., G. Weigert, D. Pabst, and M. Stehli. 2017. "Robustness analysis of an MIP for production areas with time constraints and tool interruptions in semiconductor manufacturing". In *Proceedings of the 2017 Winter Simulation Conference (WSC)*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 3714–3725. Piscataway, NJ: IEEE.

Thomas Pomorski 2009. "Major Revision Update for SEMI E10 Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)". http://www.semi.org/en/semi-e10-specification-equipment-reliability-availability-and-maintainability. called at 06/26/2018.

Rossi, F., P. Van Beek, and T. Walsh. 2006. *Handbook of constraint programming*. New York City, New York, US: Elsevier.

Topaloglu, S., and I. Ozkarahan. 2011. "A constraint programming-based solution approach for medical resident scheduling problems". *Computers & Operations Research* 38(1):246–255.

Vilím, P., P. Laborie, and P. Shaw. 2015. "Failure-Directed Search for Constraint-Based Scheduling". In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, edited by L. Perron and M. A. Trick, 437–453. Heidelberg: Springer.

Wang, T., N. Meskens, and D. Duvivier. 2015. "Scheduling operating theatres: Mixed integer programming vs. constraint programming". *European Journal of Operational Research* 247(2):401–413.

Wienke, A. 2010. *Frailty models in survival analysis*. CRC Press.

Yu, T.-S., H.-J. Kim, C. Jung, and T.-E. Lee. 2013. "Two-stage lot scheduling with waiting time constraints and due dates". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 3630–3641. Piscataway, NJ: IEEE.

AUTHOR BIOGRAPHIES

CHRISTIAN MALECK obtained his degree in Mathematics in 2016 at the Technische Universität Dresden, Germany. He has been a Research Assistant at the Institute of Electronic Packaging Technology of the Technische Universität Dresden since 2016 and works in the field of scheduling, simulation & optimization of manufacturing processes, especially in the semiconductor industry. His e-mail address is christian.maleck@tu-dresden.de.

GOTTFRIED NIEKE obtained his masters degree in mathematics in 2016 at the Technische Universität Dresden, Germany. He works since 2016 as a Research Assistant at the Institute of Electronic Packaging Technology of the Technische Universität Dresden on the field of scheduling, simulation & optimization of manufacturing processes, especially in semiconductor industry. His e-mail address is gottfried.nieke@tu-dresden.de.

KARLHEINZ BOCK, senior member IEEE, achieved the Dr.-Ing. degree in RF microelectronics from the University of Darmstadt, Germany. During his scientific life he has been with Tokoku University in Sendai Japan 1995-96, Imec vzw. In Leuven Belgium 1996-1999 and Fraunhofer IZM and EMFT in Munich Germany 1999-2014. Since March 2008 until September 2014 he also served as professor of Polytronic Microsystems at the University of Berlin (TU Berlin). He received in 2012 the Dr. h. c. from Polytechnical University of Bukarest in Romania. Since October 2014 he serves as professor of electronics packaging and director of the Institute for Electronics Packaging (IAVT) at the Dresden university of technology (TU Dresden), at present he serves also as vice dean of the faculty of electrical and computer engineering and as IEEE EPS region 8 representative in the board of governors. His email address is karlheinz.bock@tu-dresden.de.

DETLEF PABST received graduate degrees in Mathematics and Programming at the University of Wuerzburg, Wuerzburg, Germany, in 2001 and 2003, respectively. He is currently at GLOBALFOUNDRIES Inc., Malta, NY, USA. His research interests include optimization methods applied to scheduling and production control problems, as well as design of automated decision systems in a general manufacturing setting. His email address is detlef.pabst@globalfoundries.com.

MARCEL STEHLI received the Ph.D. degree in Computer Science from the University of Hagen, Hagen, Germany, in 2010. He is currently at GLOBALFOUNDRIES Inc., Dresden, Germany, as an Industrial Engineer. His research interests include production control of semiconductor wafer Fab, applied optimization, and artificial intelligence applications in manufacturing. His email address is marcel.stehli@globalfoundries.com.