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

A MATHEMATICAL MODEL FOR THE EXTERNAL SCHEDULING OF A CLUSTER TOOL WORKSTATION

Gottfried Nieke
Christian Maleck
Karlheinz Bock

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

Marcel Stehli

Factory Solutions
GLOBALFOUNDRIES Inc.
Wilschdorfer Landstrasse 101
Dresden, 01109, GERMANY

Steffen Kalisch

Manufacturing Systems and Solutions
GLOBALFOUNDRIES Inc.
Wilschdorfer Landstrasse 101
Dresden, 01109, GERMANY

ABSTRACT

Cluster tools are closed mini production environments that are especially used in the semiconductor industry. They consist of multiple processing chambers. In contrast to single processing tools they can handle multiple lots with different process characteristics at the same time. The major challenge for scheduling cluster tools is the hard to predict process times of even comparable lots as they strongly depend on a changing product mix over the tool. In this paper a mathematical model for the external scheduling of a cluster tool workstation is presented. The goal is to minimize the weighted cycle time by accounting for the changing cycle times of each lot. A MIP model assigns lots to tools of a workstation and determines for each lot the chambers, which are used during the processing steps inside the cluster tool. Finally the developed method is compared with a simulation that uses dispatching strategies.

1 INTRODUCTION

Semiconductor manufacturing is a challenging business nowadays. Leading edge process technologies, product life-cycles that are getting shorter and shorter, a highly cost-sensitive production due to worldwide competition, and very demanding customer requirements, in terms of quality and delivery compliance, require a strong focus on Fab productivity and a cost-efficient production environment. Over the years so called cluster tools became an important part of the constant effort to increase productivity. Cluster tools combine different process steps into one tool and form a mini production environment on their own with a high potential for optimization. A cluster tool usually consists of three main parts: processing modules that will be referred as chambers in this paper, load ports and a handling robot. Figure 1 displays a simplified model of a cluster tool.

The handling robot is responsible for the transport of wafers inside the cluster tool and can have one or two handling units. At dual-armed robots the position of the arms is fixed in opposite directions. Therefore the robot can only perform one loading or unloading tasks at once. In the chambers the wafers are processed. They are typically arranged radially around the handling robot, but configurations with

chambers in a row are possible. The radial configuration assures that the transport times between two chambers are similar, but it also leads to a maximum of six chambers in one tool due to space limitations. The load ports serve as the entry and exit ports of a cluster tool. The number of load ports defines the maximum number of lots that can be processed in parallel.

Now the process flow of a cluster tool will be described briefly. Cluster tools process lots with up to 25 wafers. Lots are loaded onto a load port of the cluster tool. The robot handles them into the tool and performs a sequence of process steps by moving the wafers to one or more chambers, which then execute the required processes. After all process steps for a wafer are finished, it is returned back to the load port. Once all wafers of a lot finished their sequence of process steps, the lot is unloaded from the tool and another one is assigned and placed to the now empty load port.

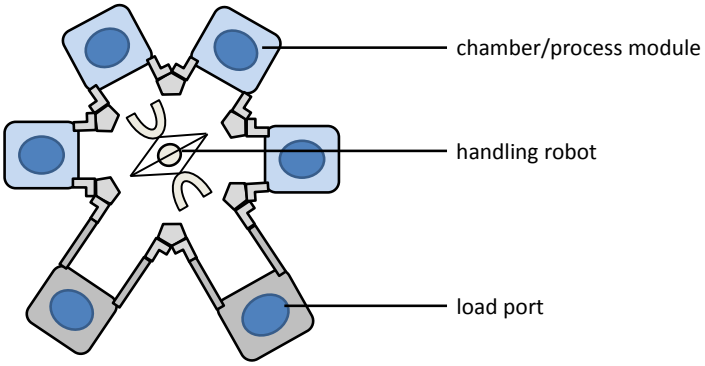

Figure 1: A simplified model of a cluster tool with a dual-armed robot.

Advantages of cluster tools over single processing machines are a reduced cycle time, a better utilization of space and lower capital costs (Wu et al. 2013), due to their capability of processing multiple wafers in parallel. Furthermore they can be easily adapted to other processes and the execution of different production processes in one configuration is also possible (Lee 2008). All these advantages lead to a broad usage of cluster tools in different production areas in semiconductor manufacturing, like etch or chemical vapor deposition.

In this paper a set of single-wafer cluster tools, without any internal buffer, will be investigated. Tools without internal buffers require the processing of a single wafer to be completed before it is moved back to the load port. Sophisticated handler strategies ensure the optimal usage of internal tool capacity. The process time for the chambers is nearly constant, however, changing the product mix at a cluster tool with possible chamber dedications, leads to different process times, even for lots with the same process type and quantity. This effect is intensified by unscheduled and scheduled downs of the certain chambers and strong chamber dedications for certain lots and processes. This leads to a reduced throughput of a cluster tool, if such effects are not handled well by the scheduling or dispatching strategy.

In the literature many researchers focus on cluster tools and their scheduling requirements. Perkinson et al. (1994) analyzed the relationship between process and transport times and the maximal throughput in cluster tools. They developed deterministic models to predict the throughput of a single-arm *n*-chamber cluster tool. Mönch et al. (2011) present challenges for the planning of processes in the semiconductor industry. They also pay attention to cluster tools and divide their scheduling methods into internal and external scheduling. Internal scheduling concentrates on the scheduling of the robot tasks. Lee (2008) presents an overview of different internal scheduling strategies. The majority of the approaches use petri nets to model the cluster tool and develop a MIP to find an optimal robot-task sequence, like Jung and Lee (2008; 2012), and Paek and Lee (2008). There is a distinction between two different types of internal scheduling. Many papers focus on cyclic scheduling, where one period of a robot task is determined to be repeated during the execution of the schedule (Wou and Zhou 2010; Zhu et al. 2015). If many lots of the same product type are processed consecutively, this method is used. Due to smaller orders there is an

increasing focus on non-cyclic scheduling. Here, each robot-task is planned individually (Kim et al. 2013; Kim et al. 2015). Overall, many different cluster tool models for the internal scheduling are observed. Wu and Zhou (2010), Rostami et al. (2001) and Qiao et al. (2012) focus on cluster tools with wafer residency constraints. Furthermore Wu et al. (2013), and Lee and Lee (2006) observe cluster tools where wafers can visit some chambers multiple times.

In contrast to the internal scheduling, only a few papers deal with external scheduling problems. Here, the lots are scheduled to the tools without determining the robot task sequence. The main challenge for the external scheduling is the varying process time of the lots on a cluster tool because of certain chamber restrictions, chamber outage, or other internal limitations. Dümmler (1999) tries to solve this problem by determining the cycle time through modelling and simulating different sequences of lots. Then a genetic algorithm is presented to solve the scheduling problem. Niedermayer and Rose (2004) point out that only with simulation the cycle time can be determined correctly, which is very expensive. That's why they present different fast cycle time approximations. They introduce a slow-down factor that describes how the cycle time increases, if different lots are processed in parallel with shared resources.

In this paper we concentrate on the external scheduling use-case and apply it to a workstation of cluster tools. As we are not considering any internal handling strategies, we have to control the order of the assigned lots, their processing needs, in term of required chambers, and process setup to achieve favorable tool utilizations while keeping cycle times low. To solve the scheduling problem of a cluster tool workstation, the challenging part is the approximation of processing times for the lots and their application in an optimization model. The approximation is done by using auxiliary variables that compute all possible lot combinations with the possible chamber combinations for each process step. We choose to develop a MIP model and use the minimization of the weighted cycle time of all lots as the objective function. Therefore, the MIP model computes an optimal assignment of the lots to the load ports of the cluster tools from the workstation. The processing order is given by calculated release dates for each lot. To control the varying product mix on a cluster tool, an optimal chamber combination for each process step is computed by the MIP model. The developed method is tested on a reduced cluster tool workstation sample. The results are compared with a simulation that uses a dispatching strategy.

The paper is organized as follows: Section 2 provides a problem statement, while section 3 describes the process time approximation and the developed optimization model. Section 4 explains the simulation model used for verification and presents the test sample and obtained results. The paper finishes with a conclusion of the results and an outlook on future work.

2 PROBLEM DESCRIPTION

The observed workstation consists of a set of cluster tools $\mathrm{M} = \{1,...,m\}, m \in \mathbb{N}$. Each cluster tool $k \in M$ has a set of load ports $L_k = \{1,...,l_k\}$, $l_k \in \mathbb{N}$ and a set of chambers $C_k = \{1,...,o_k\}$, $o_k \in \mathbb{N}$ on which a set of process setups, also called recipes, $P = \{1,...,r\}, r \in \mathbb{N}$ can be processed. On this workstation a set of jobs $J = \{1,...,n\}$, $n \in \mathbb{N}$ should be executed. Each lot $j \in J$ gets a release date $r_j \in \mathbb{R}$, a due date $d_j \in \mathbb{R}$, a number of wafers $m_j \in \{1,...,25\}$ and a weight $\omega_j \in \mathbb{R}_+$. Furthermore, it is assigned to a recipe $f_j \in P$ and a set of cluster tools $M_j \subseteq M$, on which it can be executed. Each recipe $p \in P$ has a number of process steps, also called stages, $\mathrm{S}_p = (\mathrm{s}_{p1},...,\mathrm{s}_{pn_p}), n_p \in \mathbb{N}$ where stage $s_{pi}, i \in \{1,...,n_p\}$ can be executed on a set of chambers $C_{pi} \subseteq C_k$. The time needed to execute a wafer on this stage is defined as $p_{kpi} \in \mathbb{R}$. The execution of the stages of a lot is overlapping because as soon as a wafer is finished and unloaded from the used chamber a new wafer of the lot, if present, is loaded to the freed chamber to be processed next. Each stage s_{pi} can be executed on several chambers because of quality requirements, capacity or cost reduction. Therefore, I_{kpi} describes the set of chamber combinations for this stage. The assigned chambers of the chamber combination $g \in I_{kpi}$ are pooled in the set A_g. During the processing of the lots all possible parallel chambers are used evenly. If for instance a lot j with 10 wafers uses only one chamber for a stage, all lots have to be executed on this chamber. However, if a chamber combination with two chambers is used at this stage, each chamber processes 5 wafers of the lot.

All lots that should be scheduled on this workstation wait in a queue in front of the cluster tool which has an infinite capacity. If the execution of a lot is finished, results have to be saved and the lot must be unloaded from the load port until a new lot can be loaded onto the load port. The time needed for the execution of these tasks is called $t_L \in$. Inside the cluster tool, the time needed for the handling robot to move a wafer from on chamber to another, is fixed and defined as $t_H \in$. Furthermore, it has to be ensured that the load ports and at least one of the needed chambers for each stage is available. The availability times are given by $\rho_l \in$ for L_k and $k \in M_j$ respectively $\eta_c \in$ for $c \in C_k$ and $k \in M_j$.

Overall, a cluster tool can execute as much lots in parallel as it has load ports. If several lots are assigned to a cluster tool, they are processed by their arrival order. Hence, there could occur long waiting times for some lots because the needed chambers are occupied by other wafers from lots that entered the cluster tool earlier. The wafers of the waiting lot can either be processed between small gaps that occur in the processing of the lots with the earlier enter date, or after all wafers of the previously assigned lots were executed. This makes the calculation of the process time of the lots so difficult. The last case occurs especially for the first stage of the recipe because all wafers are loaded to the used chambers from the load port and therefore there is no gap between the processing of these wafers. Furthermore, it is possible that for instance small lots can overtake big lots. To reduce the waiting time it can be stated, if the chambers that were assigned to a lot for a certain stage are used exclusively by the lot ($\kappa_g = 0$) or if they are shared with other lots ($\kappa_g = 1$). The case $\kappa_g = 0$ simplifies the approximation of the process times of the lots inside a cluster tool because otherwise it is hard to predict when there are gaps between the execution of two stages of a lot where other lots can be processed. Due to the set P of recipes that can be processed on the cluster tool and all possible chamber combinations for each stage, there are a lot possibilities how the processing of a lot is executed at a cluster tool. This makes the calculation of the process time of a lot very difficult.

The different flow patterns that are used to process a lot on all required stages form the product mix, which has a direct impact at the process time of the lots. For the developed model the following flow patterns can be observed: serial, multi-stages, parallel chambers per stage, revisiting of chambers, shared and exclusive chambers and a mix of all of these flow patterns. A further description of the flow patterns can be found in Jung and Lee (2012). In the production environment the mixed case occurs at most. Figure 2 shows an example of two possible schedules for 3 different lots on a stage with 3 shared chambers.

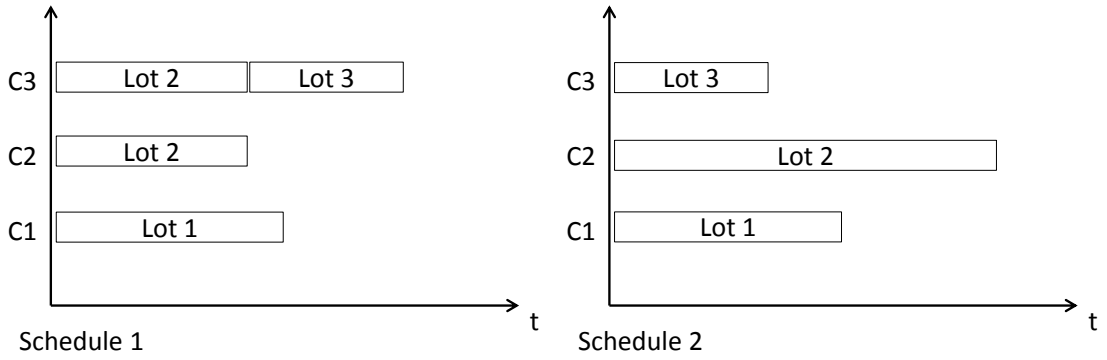

Figure 2: Possible schedules for 3 different lots on a stage with 3 chambers.

Lot 1 is dedicated to use chamber 1, while lot 3 can only be processed at chamber 3. Lot 2 is free to use the chambers 2 and 3. According to which lot (2 or 3) gets assigned first to a load port of a cluster tool, either schedule 1 (lot 2 before lot 3) or schedule 2 (lot 3 before lot 2) is executed. However, the process time of lot 2 is doubled in schedule 2 because only one chamber is used for the processing. This leads also to a higher makespan for this stage. The makespan is the total time that elapses between the

beginning and the end of the schedule. The decision, which lot is assigned first, depends on their size, priority and on the available chambers. For instance, if lot 3 has a higher priority than lot 2 schedule 2 will be preferred to reduce the cycle time of the lots, but if the lots have the same priority than schedule 1 will be preferred due to the longer process time of lot 2. All in all it is hard to predict, if lot 2 is executed on one or two chambers and thus a forecast of its cycle time is very difficult.

3 MATHEMATICAL MODEL

3.1 Computation of the Auxiliary Parameters

As it turned out in the example shown above the main problem for the scheduling of the lots is the hard to predict process time. To approximate this process time, some auxiliary parameters have to be introduced. For all $j \in J, k \in M$ and $i \in \{2,...,n_{f_j}\}$ the parameter $D_{jg_1g_2} = |A_{g_1}| \cdot p_{kf_ji} \in \mathbb{R}$ describes the time that lot j needs for the execution in stage s_{f_ji} with the chamber combination $g_2 \in I_{kf_ji}$, if the chamber combination $g_1 \in I_{kf_j,i-1}$ was used in the previous stage. For all $k \in M$ the parameter $E_{kp_1p_2g_1g_2} = |A_{g_1} \cap A_{g_2}|$ defines the number of shared chambers of the recipes $p_1, p_2 \in P$ for their chamber combinations $g_1 \in I_{kp_11}$ and $g_2 \in I_{kp_21}$. It is only implemented for the first stage because for all following stages it is directly tested in the constraints, if there are shared chambers for two lots. To get an approximation of the delay for these stages the parameter $B_{jg_1g_2}$ is defined for all $k \in M, j \in J, i \in \{2,...,n_{f_j}\}, g_1 \in I_{kf_j,i-1}$ and $g_2 \in I_{kf_ji}$. $B_{jg_1g_2}$ describes the needed capacity in stage s_{f_ji} for the chamber combination g_2, if in the previous stage the chamber combination g_1 was used. That means $B_{jg_1g_2}$ defines the number of slots that are needed in stage s_{f_ji} due to the used chamber combination g_1 in the previous state. To determine $B_{jg_1g_2}$ the relation $q = p_{kf_j,i-1} \cdot p_{kf_ji}^{-1}$ is observed by evaluating the processing of the last wafers of a lot at stage $s_{f_j,i-1}$ with the help of the following two cases.

1. $q \leq 1$ is called the bottleneck case. Stage s_{f_ji} either cannot finish all wafers of the previous stage in time or all wafers of stage s_{f_ji-1} can be processed while stage s_{f_ji} has at least as much chambers as the previous stage. Therefore, there is a possible overlap that has to be executed during the next period. This overlap is added to the number of lots that were processed in the previous stage. It follows that $B_{jg_1g_2} = |A_{g_1}| + (1-q)$.
2. For the case $q > 1$ only the processing of the last wafers of a lot is considered. If they are allocated uniformly to the available chambers in stage $s_{f_j,i-1}$ for the chamber combination g_1, i.e. $m_j \bmod |A_{g_1}| = 0$ there are no unused chambers and thus $B_{jg_1g_2} = |A_{g_1}|$. If there are unused chambers during the processing of the last wafers of a lot, it holds $B_{jg_1g_2} = m_j \bmod |A_{g_1}|$. Hence, only as many slots are needed as chambers were used in stage $s_{f_j,i-1}$.

3.2 MIP Model

To describe the developed MIP model the following decision variables are needed.

$s_j \in \mathbb{R}$	:	process start time of lot $j \in J$
$e_j \in \mathbb{R}$	:	process end time of lot $j \in J$
$w_{jg} \in \mathbb{R}$	:	delay of the process time of lot $j \in J$ for $k \in M$, $i \in \{1,...,n_{f_j}\}$ and $g \in I_{kf_ji}$
$X_{jl} : J \times L \to \{0,1\}$	:	1, if lot $j \in J$ is assigned to load port $l \in L$
$Y_{hj} : J \times J \to \{0,1\}$	:	1, if the process start of lot $h \in J$ appears before the process start of $j \in J$ where $h \neq j$, 0 otherwise
$Z_{hj} : J \times J \to \{0,1\}$	:	1, if the process end of lot $h \in J$ appears before the process end of $j \in J$ where $h \neq j$, 0 otherwise

$W_{hj} : J \times J \to \{0,1\}$: 1, if the process start of lot $h \in J$ appears before the process end of $j \in J$ where $h \neq j$, 0 otherwise

$U_{jg} : J \times I_{kpi} \to \{0,1\}$: 1, if for lot $j \in J$ the chamber combination $g \in I_{kf_j i}$ is chosen in stage $s_{f_j i}$ with $k \in M, i \in \{1,...,n_{f_j}\}$, 0 otherwise

$T_{jg_1g_2} : J \times I_{kf_j i-1} \times I_{kf_j i} \to \{0,1\}$: 1, if for lot $j \in J$ the chamber combination $g_1 \in I_{kf_j,i-1}$ is chosen in stage $s_{f_j,i-1}$ and $g_2 \in I_{kf_j i}$ for stage $s_{f_j i}$ with $k \in M, i \in \{2,...,n_{f_j}\}$, 0 otherwise

The proposed MIP model is based on the MIP model of Maleck et al. (2017). Their model was simplified by removing the risk parameter and time-link constraints. Furthermore, the constraints with the constant process time had to be adapted to the varying process time, which occurs on cluster tools for instance due to the varying product mix. At last additional constraints were added to describe the production environment of the cluster tools. For sake of readability the following notations

$$\begin{aligned} \beta_{hjk} &= Y_{hj} \cdot W_{jh} \cdot \sum_{l \in L_k} X_{jl} \cdot X_{hl} \in \{0,1\} \\ \gamma_{hjg_2} &= \beta_{hjk} \cdot U_{hg_2} \in \{0,1\} \\ \delta_{hjg_1g_2} &= \beta_{hjk} \cdot T_{hg_1g_2} \in \{0,1\} \end{aligned}$$

are shortened for all $h, j \in J$ with $h \neq j$, $k \in M_j \cap M_h$, $g_1 \in I_{kf_h,v-1}$, $g_2 \in I_{kf_h v}$, $i \in \{2,...,n_{f_j}\}$ and $v \in \{1,...,n_{f_h}\}$ with $v \leq i$. They were linearized with the help of the *max*-condition from the ILOG CPLEX Optimization Studio.

The objective function minimizes the weighted cycle times of each lot

$$CT = \sum_{j \in J} \omega_j e_j \to \min$$

subject to

$$r_j \leq s_j \quad ; \quad \forall j \in J \tag{1}$$

$$\sum_{k \in M_j} \sum_{l \in L_k} X_{jl} = 1 \quad ; \quad \forall j \in J \tag{2}$$

$$X_{jl} \cdot \rho_l \leq s_j \quad ; \quad \forall j \in J, \forall k \in M_j, \forall l \in L_k \tag{3}$$

$$\sum_{g \in I_{kf_j i}} U_{jg} = \sum_{l \in L_k} X_{jl} \quad ; \quad \forall j \in J, \forall k \in M_j, \forall i \in \{1,...,n_{f_j}\} \tag{4}$$

$$s_j \geq \sum_{g \in I_{kf_j 1}} U_{jg} \cdot \max_{c \in A_g} \eta_c \quad ; \quad \forall j \in J, \forall k \in M_j, \forall g \in I_{kf_j 1} : \kappa_g = 0 \wedge \sum_{l \in L_k} X_{jl} = 1 \tag{5}$$

$$U_{jg_j} + U_{hg_h} \leq 1 \quad ; \quad \forall j,h \in J, \forall k \in M_j \cap M_h, \forall g_j \in I_{kf_j 1}, \forall g_h \in I_{kf_h 1} : E_{kf_h f_j g_h g_j} > 0 \wedge \beta_{hjk} = 1 \wedge (\kappa_{g_h} = 0 \vee \kappa_{g_j} = 0) \wedge j \neq h \tag{6}$$

$$e_h + t_L \leq s_j \quad ; \quad \forall j,h \in J, \forall k \in M_j \cap M_h, \forall l \in L_k : X_{jl} = 1 \wedge X_{hl} = 1 \wedge e_h \leq s_j \wedge h \neq j \tag{7}$$

$$K \cdot (X_{hl} - Y_{hj} - 1) + X_{jl} e_j \leq s_h \quad \forall j,h \in J, \forall k \in M_j \cap M_h, \forall l \in L_k, \tag{8}$$

$$K \cdot (X_{jl} + Y_{hj} - 2) + X_{jl} e_j \le s_j \qquad \begin{matrix} K > 0 : j \ne h \\ \forall j,h \in J, \forall k \in M_j \cap M_h, \forall l \in L_k, \\ K > 0 : j \ne h \end{matrix} \tag{9}$$

The first constraint (1) assures that a lot cannot be started before its release date. (2) and (3) ensure that each lot is only assigned to one load port and that this load port is available. Constraint (4) guarantees that for each stage of the recipe from a lot exactly one chamber combination is assigned, if and only if the lot is assigned to one of the load ports of the corresponding cluster tool. (5) assures that all chambers of the chosen chamber combination for stage 1 have to be available, if they are used exclusively. With the help of (6) it is forbidden for this case that other lots are assigned to these chambers. $\beta_{hjk} = 1$ assures that the two observed lots are processed on the same cluster tool simultaneously. With (7) the delay time between two lots on the same load port is specified, meaning the time that is needed between operation and process start respectively process end. Constraints (8) and (9) ensure that for each load port only one lot can be assigned at the same time. During the implementation of these two constraints, the term $X_{jl} e_j$ was linearized with the help of the implication-constraint of the IBM ILOG Optimization studio so that the whole model can be implemented as a MILP.

To calculate the approximated process time of a lot its delay time has to be described. This was realized with the following three constraints, which will be described briefly. For all $j \in J, k \in M_j$ and $g_j \in I_{kf_j 1}$ with $\kappa_{g_j} = 1$

$$w_{jg_j} \ge \min_{c \in A_{g_j}} \sum_{h \in J} \sum_{\substack{g_h \in I_{kf_h 1}: \\ c \in A_{gh}}} \gamma_{hjg_h} \cdot \left\lceil \frac{m_h}{|A_{g_h}|} \right\rceil \cdot p_{kf_h 1}$$

calculates the minimal time that one of the assigned chambers of lot j is available for stage 1 because this is the time this lot has to wait before its processing can be started, due to the requirement that the lots inside the cluster tool are scheduled by their arrival order. It is only implemented for the case that the chambers are shared ($\kappa_{g_j} = 1$) because the exclusive case is already covered by (6). γ_{hjk1g_h} assures that the processing of the lots j and h overlap each other on the same tool. The calculation of the waiting time of stage 1 is described separately because there are no gaps between the execution of the wafers of a lot due to the fact that they are all unloaded from the load ports to the assigned chambers.

For all stages that follow after the first stage, the computation of the delay time is divided into exclusive and shared chambers because, in contrast to the first stage, both cases can occur. For all $j \in J$, $k \in M_j, i \in \{2,...,n_{f_j}\}, g_j \in I_{kfji}$ and $p_{\max,i-1} = \max\{p_{kp,i-1} \mid j \in J, k \in M_j\}$ with $\kappa_{g_j} = 1$ and $\sum_{\substack{h \in J: \\ h \ne j}} \beta_{hjk} \ge 1$

$$w_{jg} \ge \max\left\{ \underbrace{0; \sum_{\substack{g_1 \in I_{kf_j,i-1}, \\ g_2 \in I_{kfji}}} \left(T_{jg_1g_2} \cdot D_{jg_1g_2}\right)}_{=(I)} \underbrace{- \min_{j_1 \in J} \left(p_{kf_{j_1},i-1} + (1 - \beta_{j_1jk}) \cdot p_{\max,i.1}\right)}_{=(II)} \underbrace{+ \sum_{\substack{h \in J: \\ h \ne j}} \sum_{g_1 \in I_{kf_h,i-1}} \left(\delta_{hjg_1g_2} \cdot D_{hg_1g_2}\right)}_{=(III)} \right\}$$

describes the case with shared chambers. (I) computes, how much time the enquiring lot j needs on the current stage by multiplying the needed process time for the actual stage D_{jg1g2} of lot j with the used chamber combinations for the previous and the current stage. They are described by T_{jg1g2}. For two consecutive stages only one entry in T_{jg1g2} equals one for each lot j. (II) determines the time that is left, until a new wafer from the previous stage is transported to the actual stage s_{fji}. Therefor the minimal

process time for the previous stage $s_{f_j,i-1}$ of all lots is determined. If the two observed lots are not processed on the same cluster tool, a large value is added so that the sum of (I) to (III) becomes less than zero and hence the delay time w_{jg} is set to 0 because of the *max*-condition. At last (III) sums up the time, the processing of the wafers of the earlier assigned lots need, which use at least one of the chambers from chamber combination g. $\delta_{hjg_1g_2}$ ensures that the lots are processed simultaneously on the same cluster tool, as the enquiring lot j. Furthermore it specifies, which chamber combinations were used for the lot $h \in J, h \neq j$. The time needed for lot h is given by $D_{hg_1g_2}$. Overall, the delay time is computed by summing up (I) to (III). If there is enough time left to process enquiring lot j, and all the already assigned lots before a new wafer arrives, than the sum is smaller than 0 meaning that the delay time becomes 0 because of the *max*-condition.

For the case that the chambers of g are used exclusive the delay time for the enquiring lot j is determined for all $j \in J, k \in M_j, i \in \{2,...,n_{f_j}\}$ and $g \in I_{kf_ji}$ with $\kappa_{g_j} = 0$ and $\sum_{\substack{h \in J: \\ h \neq j}} \beta_{hjk} \geq 1$ by

$$w_{jg} \geq \sum_{\substack{h \in J: \, i_1 = 1 \\ h \neq j}} \sum^{n_{f_h}} \sum_{\substack{g_h \in I_{kf_hi_1}: \\ E_{kf_hf_jg_hg} > 0}} \gamma_{hjg_h} \cdot \left\lceil \frac{m_h}{|A_{g_h}|} \right\rceil \cdot p_{kpi_1} \, .$$

Again, γ_{hjg_h} assures that the lots j and h are processed on the same cluster tool at the same time. Than the needed process time of all stages $s_{f_hi_1}$ of lot h, with the chamber combination g_h that share chambers with the chamber combination g of j, is computed. The delay time is the sum of these determined process times.

Now, the approximation of the process time can be expressed for all $j \in J$ and $k \in M_j$ with the help of the constraint

$$s_j + \sum_{g \in I_{kf_j1}} \left(w_{jg} + p_{kp1} \cdot U_{jg} \cdot \left\lceil \frac{m_j}{|A_g|} \right\rceil \right) + \left(n_{f_j} + 1 \right) \cdot t_H \cdot m_j + \sum_{i=2}^{n_{f_j}} \left(\sum_{\substack{g_1 \in I_{kf_j,i-1}, \\ g_2 \in I_{kf_ji}}} w_{jg} + p_{kf_ji} \cdot T_{jg_1g_2} \cdot \left\lceil \frac{B_{jg_1g_2}}{|A_{g_2}|} \right\rceil \right) \leq e_j \, .$$

It defines the time between the process start and end time of a lot j by summing up the raw production and delay times for each stage with the corresponding chamber combination and the transport time between the stages.

4 TESTS AND RESULTS

4.1 Test Environment

In order to test the above described model, a simplified example was generated. The observed workstation consists of two identical cluster tools. Each of these cluster tools has a dual-armed handling robot and four radial arranged chambers. The following dispatching strategy is used to assign a new lot out of the infinite queue in front of the workstation to a load port of a cluster tool. If a load port is freed, one lot out of this queue is selected to be processed next on the workstation. The selection is done by calculating an x-factor which describes the relationship of the overall process time including waiting times and the raw process time of a lot, while taking its priority into account. At least one of the needed chambers for the first stage has to be available so that the processing of the chosen lot can be started immediately. If no chamber is available for any lot, the lot with the shortest waiting time is chosen to be processed next. Inside the cluster tool the FIFO dispatching strategy is used.

Overall, seven lots with 5 to 25 wafers with three different recipes should be scheduled. Their properties are displayed in Table 1. The recipes of these lots have two stages. Recipe 1 uses for stage 1 chamber 1, while recipe 10 can use the chambers 2 and 3 and the execution from recipe 11 can be realized

on the chambers 1, 2 and 3. For the execution of stage 2 all recipes use chamber 4. This means that for recipe 10 there are three and for recipe 11 seven different possible chamber combinations, how the lots can be executed on the cluster tools. Furthermore, the chambers of stage 1 are used exclusive while chamber 4 for stage 2 can be shared for all recipes. Although stage 2 only uses one chamber it does not become a bottleneck of the cluster tool because the process time is a lot shorter than the process time for stage 1.

Table 1: Properties of the scheduled lots.

$j \in J$	f_j	r_j (s)	m_j	ω_j	M_j
1	1	0	5	1	1, 2
2	10	0	10	2	2
3	1	30	15	1	1, 2
4	11	0	20	1	2
5	11	20	25	5	1
6	1	0	5	1	2
7	10	0	10	2	1, 2

4.2 Results

In order to test the results of the optimization model, which was implemented using the ILOG CPLEX Optimization studio 12.8, a simulation model was implemented using the simcron MODELLER 3.3. A deadlock-free handling robot control was developed so that the cluster tool behaves similar as in reality. This allows the comparison of the generated schedule and the approximated process times of the lots with a simulation model using the above described dispatching rules (*sim_disp*). In the queue in front of the workstation the lots are sorted by their x-factor. The theoretical results of the MIP model will be called *ilog_opt*, and the simulation of the theoretical results is defined as *sim_opt*. The schedule of *sim_opt* is generated by applying the computed tool assignments and the release dates of the lots and simulating them with the developed simulation model. The transportation times inside the cluster tool are assumed to be identical ($t_H = 2.5s$) and the delay time between two lots on the same load port is 120 seconds ($t_L = 120s$). All load ports and chambers get an availability ρ_l respectively η_c which is displayed in the Gantt charts as grey rectangles. The calculations were executed on one core of an Intel i76600U. The computation time for the MIP model was 30 seconds.

At first we compare *sim_opt* with *ilog_opt*. The variation of the approximated process time of the lots regarding to the simulated one is between -1.7% and 2.2%. The process time cannot be computed exactly during the optimization because the handling robot was not modeled. In the simulation model the handling robot control policy prefers some wafers over others to prevent deadlocks. There are two exceptions. For lot 7 the difference of the approximated and the simulated process time was 5.4% and for lot 5 -7.7%. One possible reason for the variation of lot 5 is that it has many wafers and therefore it is more likely that for example more lots are delayed because of the intern control policy of the handling robot. Overall, it turns out that the simulated process time tends to be smaller than the approximated one. Therefore, the objective function *CT* of *ilog_opt* is reduced by about 0.9% in *sim_opt* and the makespan is decreased too.

Comparing *sim_opt* with *sim_disp* it turns out that the overall makespan of *sim_disp* is about 7 minutes smaller than the makespan of *sim_opt*. However, the workload of the cluster tools of the workstation of *sim_disp* is about 4% higher compared to the workload of *sim_opt* because *sim_opt* takes longer waiting times, by assigning later release dates for some lots, into account to minimize the weighted cycle time. In contrast *sim_disp* starts the available lots as early as possible. This is the reason why the weighted cycle time of *sim_disp* is about 4.3 hours longer than the one of *sim_opt*. This means that *sim_opt* reduces the weighted cycle time of *sim_disp* up to 38%. The main difference between the two

executed schedules can be found in the order and assignment of the lots of cluster tool 1. The Gantt charts for the load port assignment are displayed in Figure 3.

Sim_disp chooses lot 3 over lot 5 because some of the needed chambers for the processing of stage 1 of lot 5 are not available. The same counts for lot 7. Due to constraint (5) the lots cannot be started at tool 1 because for stage 1 they use the chambers exclusively and therefore, all needed chambers have to be available. On the contrary *sim_opt* takes the higher priority of lot 5 over lot 3 into account, although lot 5 has to wait until all required chambers get available. It results in a higher makespan of *sim_opt* regarding to *sim_disp*, but the weighted cycle time of the scheduling problem is minimized. The Gantt charts of the chambers in Figure 3 display the overlapping processing of the two stages. Furthermore, for chamber 4 it is depicted how the parallel processing of lot 3 and 7 is executed.

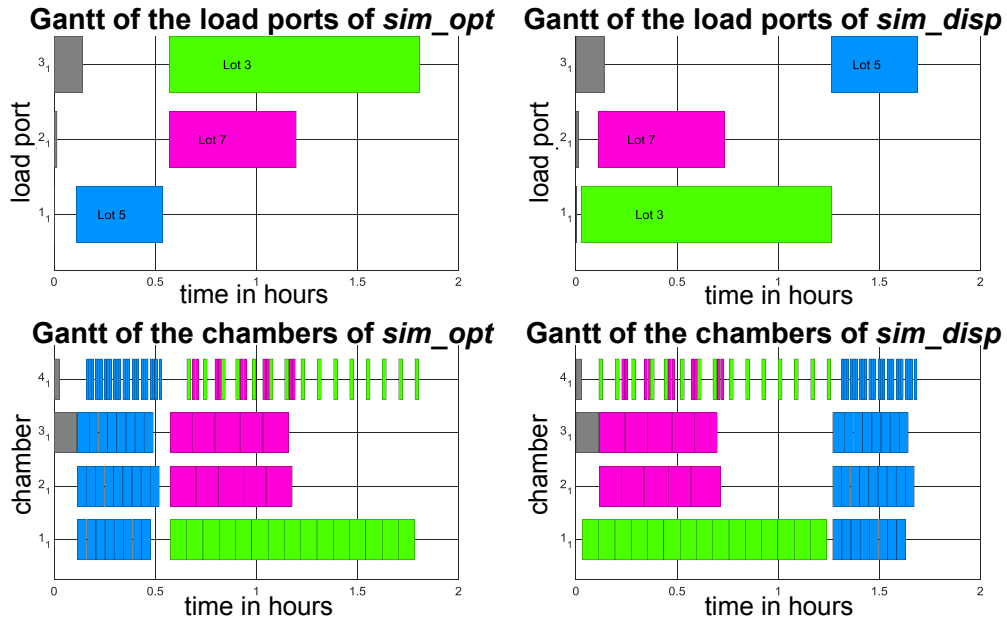

Figure 3: Gantt charts for the load ports and chambers of cluster tool 1 for *sim_opt* and *sim_disp.*

Overall, it turns out that for the minimization of the makespan *sim_disp* is the best choice because of the usage of dispatching rules, which generate non-delayed schedules. Furthermore, if additional properties are added to the lots, for instance weights, it could be beneficial to delay the processing of some lots to optimize the overall objective target. Therefore the developed optimization model can be used which computes optimized release dates and cluster tool assignments for each lot.

5 CONCLUSION AND OUTLOOK

In this paper a new approach for the external scheduling of a cluster tool workstation is presented, using a MIP model that includes the approximation of the process time of each lot directly into its constraints. The goal of this MIP model is to find an optimal assignment of lots to tools of a cluster tool workstation and to find for each lot a suitable chamber combination for each stage. The processing order is given by the computed release dates. The main problem of the external scheduling of a cluster tool is the varying process time of the lots due to the changing product mix. It is solved by approximating the process time of each lot with auxiliary variables, while taking all possible chamber combinations for each stage of a recipe into account. The model was tested on a small workstation with two cluster tools for seven lots. The solutions of the MIP model, the simulation of the MIP solution and a simulation that uses dispatching rules are compared with each other. It turns out that the simulation of the optimized schedule generates

the best results and that, even for this small test sample, the weighted cycle time of the lots can be optimized, while the makespan remains similar.

For further investigations the presented approximation should be tested on other cluster tool types. Currently, the developed MIP model is tested on more complex examples with more lots. Later, the tests will be extended by more cluster tools per workstation, more lots and also for longer time intervals. The goal is to integrate the presented model into a rolling horizon environment. Furthermore the robustness of the obtained solutions should be tested and, if the complexity of the model gets too big, a constraint programming approach should be implemented.

ACKNOWLEDGMENTS

The work was performed in the project "Erforschung von Grundlagen und Konzepten zur Gestaltung einer automatisch auf sich ändernde Anforderungen, hinsichtlich Produktionsvolumen und Produktmix, reagierende Halbeiterfabrik" (Responsive Fab), co-funded by grants from the European Union (EFRE) and the Free State of Saxony (SAB). (project number 100259418).

REFERENCES

Dümmler, M. 1999. "Using Simulation and Genetic Algorithms to Improve Cluster Tool Performance". In *Proceedings of the 1999 Winter Simulation Conference,* edited by P. A. Farrington et al., 875-879. Piscataway, New Jersey: IEEE.

Jung, C. and T.-E. Lee. 2008. "Efficient Scheduling Method Based on an Assignment Model for Robotized Cluster Tools". In *Proceedings of the 2008 IEEE International Conference on Automation Science and Engineering*, 79-84. Washington, DC: IEEE.

Jung, C. and T.-E. Lee. 2012. "An Efficient Mixed Integer Programming Model Based on Timed Petri Nets for Diverse Complex Cluster Tool Scheduling Problems". *IEEE Transactions on Semiconductor Manufacturing* 25(2):186-199.

Kim, H.-J., J.-H. Lee and T.-E. Lee. 2013. "Scheduling Cluster Tools with Ready Time Constraints for Consecutive Small Lots". *IEEE Transactions on Automation Science and Engineering* 10(1):145-159.

Kim, H.-J., J.-H. Lee and T.-E. Lee. 2015. "Noncyclic Scheduling of Cluster Tools with a Branch and Bound Algorithm". *IEEE Transactions on Automation Science and Engineering* 12(2):690-700.

Kim, H.-J., T.-E. Lee, H.-Y. Lee and D.-B. Park. 2003. "Scheduling Analysis of Time-Constrained Dual-Armed Cluster Tools". *IEEE Transactions on Semiconductor Manufacturing* 16(3):521-534.

Lee, T.-E. 2008. "A Review of Scheduling Theory and Methods for Semiconductor Manufacturing Cluster Tools". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason et al., 2127-2135. Piscataway, New Jersey: IEEE.

Lee, H.-Y. and T.-E. Lee. 2006. "Scheduling Single-Armed Cluster Tools with Reentrant Wafer Flows". *IEEE Transactions on Semiconductor Manufacturing* 19(2):226-240.

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 *Proc. of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 3714-3725. NJ: IEEE

Mönch, L., J. W. Fowler, S. Dauzère-Pèrès, 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.

Niedermayer, H. and O. Rose. 2004. "Approximation of the Cycle Time of Cluster Tools in Semiconductor Manufacturing". In *Proceedings of the 2004 Industrial Engineering Research Conference*, edited by R. King et al., Houston, Texas.

Paek, J.-H. and T.-E. Lee. 2008. "Optimal Scheduling of Dual-Armed Cluster Tools without Swap Restriction". In *Proceedings of 2008 IEEE International Conference on Automation Science and Engineering* , 103-108. Washington, DC: IEEE.

Perkinson, T. L., P. K. McLarty, R. S. Gyurcsik and R. K. Cavin. 1994. "Single-Wafer Cluster Tool Performance: An Analysis of Throughput". *IEEE Transactions on Semiconductor Manufacturing* 7(3): 369-373.

Qiao, Y., N. Q. Wu and M. C. Zhou. 2012. "Real-Time Control Policy for Single-Arm Cluster Tools with Residency Time Constraints and Activity Time Variation by Using Petri Net". *IEEE International Conference on Networking, Sensing and Control*, April 11th-14th, Beijing, China.

Rostami, S., B. Hamidzadeh and D. Camporese. 2001. "An Optimal Periodic Scheduler of Dual-Arm Robots in Cluster Tools with Residency Constraints". *IEEE Transactions on Robotics and Automation* 18(5): 609-618.

Wu, N. Q. and M. C. Zhou. 2010. "A Closed-Form Solution for Schedulability and Optimal Scheduling of Dual-Arm Cluster Tools With Wafer Residency Time Constraint Based on Steady Schedule Analysis". *IEEE Transactions on Automation Science and Engineering* 7(2): 303-315.

Wu, N. Q., M. C. Zhou, F. C. Chu and C. Chu. 2013. "A Petri-Net-Based Scheduling Strategy for Dual-Arm Cluster Tools with Wafer Revisiting". *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43(5):1182-1194.

AUTHOR BIOGRAPHIES

GOTTFRIED NIEKE obtained his master's 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.

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.

STEFFEN KALISCH received a M.S. in Industrial Engineering from Dresden University of Technology, Germany. He is leading the WIP Flow domain at GLOBALFOUNDRIES Inc., Dresden, Germany. His email address is steffen.kalisch@globalfoundries.com.

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 on the field of scheduling, simulation & optimization of manufacturing processes, especially in semiconductor industry. His email address is christian.maleck@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.