*Proceedings of the 2016 Winter Simulation Conference*
*T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.*

# A CP APPROACH FOR PLANNING LOT RELEASE DATES IN A SUPPLY CHAIN OF SEMICONDUCTOR MANUFACTURING

Dirk Doleschal
Gottfried Nieke
Gerald Weigert

Electronics Packaging Laboratory
Technische Universität Dresden
D-01062 Dresden, GERMANY

Andreas Klemmt

Infineon Technologies Dresden GmbH
Königsbrücker Str. 180
D-01099 Dresden, GERMANY

## ABSTRACT

In this paper a constraint programming (CP) approach for calculating release dates for lots within a supply chain environment is investigated. The lot start times are verified by a simulation model using different dispatching rules focusing on tardiness. To test the presented CP approach a simple fab model is constructed. The fab model consists of parallel batch machines as well as work centers and single machines. The investigated objectives are tardiness, earliness and cycle time. Due to the high complexity decomposition methods for the CP approach are tested. The results from the CP method are lot start dates which are verified by a downstream simulation run. The results show that the presented CP approach could outperform the simulation for all objectives. The content of this paper could be used as a first investigation for new scheduling methods within a supply chain management.

## 1 INTRODUCTION

The semiconductor industry is an area with high competition and increasing demands to their products, e.g. building smaller and more powerful chips and having more and more transistors on one chip. In order to handle these problems and to maximize their profit, the semiconductor manufacturers have to reduce the costs of producing their products. This can be done by optimizing their production lines. On the one hand bigger wafers with more chips are used in the production line, but shifting the machines of the production line to the size of the bigger wafers is often very expensive. On the other hand the manufacturers can optimize the product flow of their production lines and supply chains to reduce the costs of the manufacturing process. Especially this is very important for the supply chains. Lee et al. 2006 already mentioned that they are dispersed all over the world, and because of the complexity of the production process and the increasing dispersion of the manufacturing facilities, warehouses, distribution centers etc. it becomes more and more important to manage them. To reduce the complexity, bottleneck detection and management methods can be used.

In literature many articles and books about supply chain management (SCM) can be found, but most of them deal with it from an economic point of view. In contrast to the economic point of view, in this investigation a scheduling perspective on supply chains should be used. Furthermore, literature that deals with SCM in semiconductor industry is hard to find. For example, Ehm et al. 2011 present a reference model for supply chain simulation in semiconductor industry. Lee et al. 2006 describes a mathematical optimization problem to optimize a semiconductor supply chain. He divides the supply chain into two parts. Here the Die Bank is a decoupling point, because he observed that in the beginning of the production process the goals of the manufacturer and in the end the goals of the consumer are more

978-1-5090-4486-3/16/$31.00 ©2016 IEEE 

important. Due to the high complexity of such scheduling problems, a bottleneck detection should be included. The main problem of bottleneck detection and management is that there is no unified definition of a bottleneck. Kromer 2005 mentions that there are

- active periods,
- capacity oriented,
- queue oriented,
- and workload oriented

approaches to detect a bottleneck. Depending on which approach was chosen, different machines can be obtained to be the bottleneck. In this work the workload oriented approach will be used. Methods for managing bottlenecks are divided into simulation-based and analytical methods. Many papers describe approaches to manage a bottleneck, e.g. Wang et al. 2008 introduce the "Algorithm for exponential dynamic bottleneck detection" and Yang et al. 2014 present a simulation based approach. Zhang et al. 2009 notes that the bottleneck in a production line can change depending on the workload and production mix. In summary no literature could be found which deals with scheduling methods for supply chain management in a semiconductor environment. In contrast to the literature found in this investigation, a mathematical scheduling method is used to calculate lot release dates. Currently this is tested for one part of a supply chain. In future works the described method should be enhanced for multiple coupled parts of a supply chain.

This paper is organized as followed. In chapter 2 the problem is described in detail. In chapter 3 the description of the mathematical optimization model is presented. Chapter 4 describes the generated test data and the implementation of the mathematical optimization approach for generating results. In chapter 5 the results are presented and finally in chapter 6 a conclusion and outlook is presented.

## 2 PROBLEM DESCRIPTION AND MOTIVATION

In this investigation we try to calculate lot release dates for each part of a supply chain. Basis for this is the due date for the lots at the end of the supply chain. One example for a supply chain of a semiconductor manufactory is presented in Figure 1. Thereby the fabs could be distributed all over the world. In this example the supply chain consists of three main parts – the Frontend, the Die Bank and the Backend. After the Backend the finished products have to be shipped to the customers. In the example of Figure 1 the Frontend is furthermore divided into three manufacturing sites and the Backend is divided into two sites. So in summary, the supply chain consists of six manufacturing sites (three Frontend sites, one Die Bank site and two Backend sites).

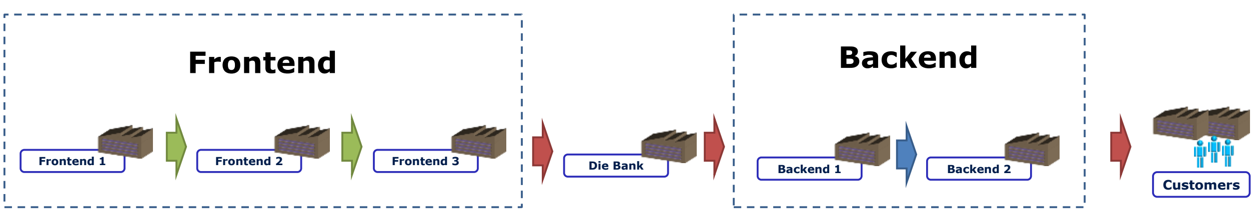


Figure 1: Example for a supply chain of a semiconductor industry.

Now the goal of this investigation is to calculate release dates for each manufactory site and each lot which have to be processed. Thereby the first goal is to meet the due date made for the customer. Furthermore the cycle time has to be reduced to reduce the WIP and therefore the fixed capital. The last goal is to reduce the earliness. With this goal the storage costs have to be reduced. The knowledge about the calculated release dates (and therefore the due date of the downstream fabrication site) is essential for efficient scheduling within each site.

In this approach the individual parts (fabrications) of the supply chain are represented in as much detail as possible. This leads to a very high complexity. For testing different methods, first a simulation model is built up. This is done in a dynamic way, which means that the simulation model is build up automatically based on the input data. So the input data defines the whole problem. One problem (one part of the supply chain) could be described by:

- The number of different stations
- Number of machines for each station
- Type and properties of the machines
- Number of different products
- Routes and properties for each products
- Lots and their properties

Based on this problem description, the goal is to calculate lot release dates for each lot under the subject to meet the due dates as exact as possible and minimize the cycle time as well. To reach this goal a constraint programming (CP) approach is used and compared to the result gained from a simulation run. The calculated release dates are tested within the simulation model using tardiness orientated dispatching rules. For better understanding the lots (typical used within semiconductor fabs) are called jobs in the further progress. This is done because the further investigation is useable for each production manufactory, not only the semiconductor. Typically for scheduling the smallest moveable units are called jobs. So the jobs have the same properties as the lots mentioned before.

### 2.1 Problem Formulation

In this investigation a scheduling problem with *m* machines and *n* jobs is observed. The underlying model consists of different process stages which contain either one (single stage) or multiple machines (parallel machine work centers). Also two machine types are considered: single machines, where only one job can be processed at once and batch machines, where a batch of several equal jobs can be processed simultaneously. Each stage can have set up times. Three different objective values are examined: the tardiness *T*, earliness *E* and cycle time *C*. Tardiness and earliness are defined as followes:

$$T = \max_{i \in J} \{0, z_i - d_i\} \quad (1)$$

$$E = \max_{i \in J} \{0, d_i - z_i\} \quad (2)$$

with

| | | |
|---|---|---|
| $J = \{1,..,n\}$ | : | set of all jobs |
| $d_i$ | : | Due date of job *i* |
| $z_i$ | : | Finishing date of job *i*. |

Additionally, two different types of objective functions are examined within the following constraint programming approach: a weighted sum and a multi criteria optimization. For the weighted sum the objective values where normalized with the help of the formula (3).

$$\frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (3)$$

$X_i \in \{C,T,E\}$, $i \in J$ and $X_{\min}$ and $X_{\max}$ are the lower respectively upper bound of $X_i$. In the test cases $X_{\min}$ and $X_{\max}$ are obtained by a simulation run, which was processed before the CP optimization starts.

## 3 CP MODEL DESCRIPTION

As mentioned in section 2 a constraint programming approach was tested to calculate the job release date for each part of the supply chain. To reach this goal the special product flow within the fabrication has to be considered. In this section the CP model is described in a mathematical way. So, first the input parameters and variables have to be described.

The scheduling model must meet different requirements, e. g. jobs cannot be processed before there release date and each job has a production order, i.e. a determined order how the stages of the system must be visited, that must be satisfied. For the description of the mathematical model some variables have to be defined. Also it is assumed that the index *i* represents jobs and that the index *k* represents machines. The notation of the following variables and equations is oriented towards Klemmt 2012. All in all the mathematical formulation of the model can be described with the help of the following variables:

| | | |
|---|---|---|
| $M = \{1,...,m\}$ | : | set of all machines |
| $n_i$ | : | Number of operations of Job *i* |
| $O_i = \{O_{i1},...,O_{in_i}\}$ | : | all operations of Job *i* |
| $M_B \subset M$ | : | set of all batch machines |
| $M_{io} \subset M$ | : | set of all machines that can process operation $O_{io}$, $o \in \{1,...,n_i\}$ |
| $b_k$ | : | Batch size of machine *k* |
| $f_i$ | : | Product type of job *i* |

All these variables can be obtained by the description of the model. For the formulation of the mathematical model these two decision variables are needed:

| | | |
|---|---|---|
| $W_{iok} \in \{0,1\}$ | : | 1, if $O_{io}$ is processed by machine *k*, else 0 |
| $Y_{iokf} \in \{0,1\}$ | : | 1, if $O_{io}$ is processed by machine $k \in M_B$ with the setup for products of the type $f_i$, else 0 |

Additionally, each operation $O_{io}$ gets a start time $s_{io}$, a finishing time $e_{io}$, an optional setup time $t_{io}$ and a processing time $p_{io}$. Furthermore each job $i \in J$ gets a release date $r_i$, a due date $d_i$ and a product type $f_i$. Within the CP model two different objective functions are tested. The objective functions of the scheduling problem can be described as either

$$\omega_T \sum_{i=1}^{n} T_i + \omega_E \sum_{i=1}^{n} E_i + \omega_C \sum_{i=1}^{n} C_i \to \min \tag{4}$$

or

$$\sum_{i=1}^{n} T_i \to \min \tag{5}$$

$$\omega_E \sum_{i=1}^{n} E_i + \omega_C \sum_{i=1}^{n} C_i \to \min \tag{6}$$

Thereby $\omega_E$, $\omega_C$ and $\omega_T$ are weighting parameters of the corresponding objective value. In (4) a weighted sum is used to optimize the whole problem. Here the importance of each objective could be set with the weighting parameters. In the second version ((5) and (6)) a multi-criteria policy is used, where the different objectives are ordered and a lexicographic optimization is performed. In this multi-criteria optimization the tardiness is optimized with highest priority and the weighted sum of earliness and cycle time is optimized secondarily.

The constraints for the CP optimization problem could be described in the following way:

$$r_i \le s_{i,1} \qquad ;\forall i \in J \tag{7}$$

$$s_{io} + p_{io} + t_{i,o+1} \le s_{i,o+1} \qquad ;\forall i \in J, 1 \le o < n_i - 1 \tag{8}$$

$$\sum_{k \in M_{io}} W_{iok} = 1 \qquad ;\forall i \in J, k \notin M_B, 1 \le o < n_i \tag{9}$$

$$\sum_{\substack{k \in M_{io} \\ k \notin M_B}} Y_{iokf} = 1 \qquad ;\forall i \in J, f \in \{f_i\}_{i \in J}, 1 \le o < n_i \tag{10}$$

$$\sum_{\substack{i \in J \\ f = f_i}} Y_{iokf} \le b_k \qquad ;k \in M_B, f \in \{f_i\}_{i \in J}, 1 \le o < n_i \tag{11}$$

With (7) the release date for each job is complied. Here the first operation of each job could not start before the release date. Equation (8) ensures the sequence of the operations for each job . Furthermore this equation ensures that the operations could not overlap and an eventual setup time is considered. In equation (9) it is ensured that each operation is assigned exactly to one machine out of the set of the allowed machines. Equations (10) and (11) ensures that the capacity of the machines is not exceeded.

First results show that the presented CP model has a high complexity and therefore needs a long computation time to gain applicable results. Due to this an additional CP model based on bottleneck decomposition is tested. In general a bottleneck formulation describes the bottleneck and its surrounding in detail and simplifies the rest of the model, e.g. by modeling stages with delays and not in detail. For this an upstream simulation run is used to detect the bottlenecks within the model and this information is afterwards used in the constraint programming. All non-bottleneck stations are considered in the CP model as a delay. Also the delay time is calculated product-fine within the simulation model. The described CP model has to be modified only in (7) and (8) to consider the delay times. We assume that the delay time in front of an operation $o$ for a job $i$ is described with $d_{i,o}$. Than the following equations could be used:

$$r_i + d_{i,1} \le s_{i,1} \qquad ;\forall i \in J \tag{12}$$

$$s_{io} + p_{io} + t_{i,o+1} + d_{i,o+1} \le s_{i,o+1} \qquad ;\forall i \in J, 1 \le o < n_i - 1 \tag{13}$$

Here only the bottleneck stations are defined as an operation. Furthermore the due date $d_i$ of a job $i$ has to be reduced by the delay time between the last bottleneck station and the actual due date.

## 4 TEST DATA AND IMPLEMENTATION

To test the described method, first test data have to build up. For a first test and to validate the CP model, a simple model was used.

### 4.1 Test Model

The observed model consists of five stages and ten different product types and represents a simplified view of a backend test. Figure 2 shows a schematic view of the model. Thereby this model only comprises one station out of the supply chain. In future, the stations should be linked and the presented CP approach should be executed for each station of the supply chain.

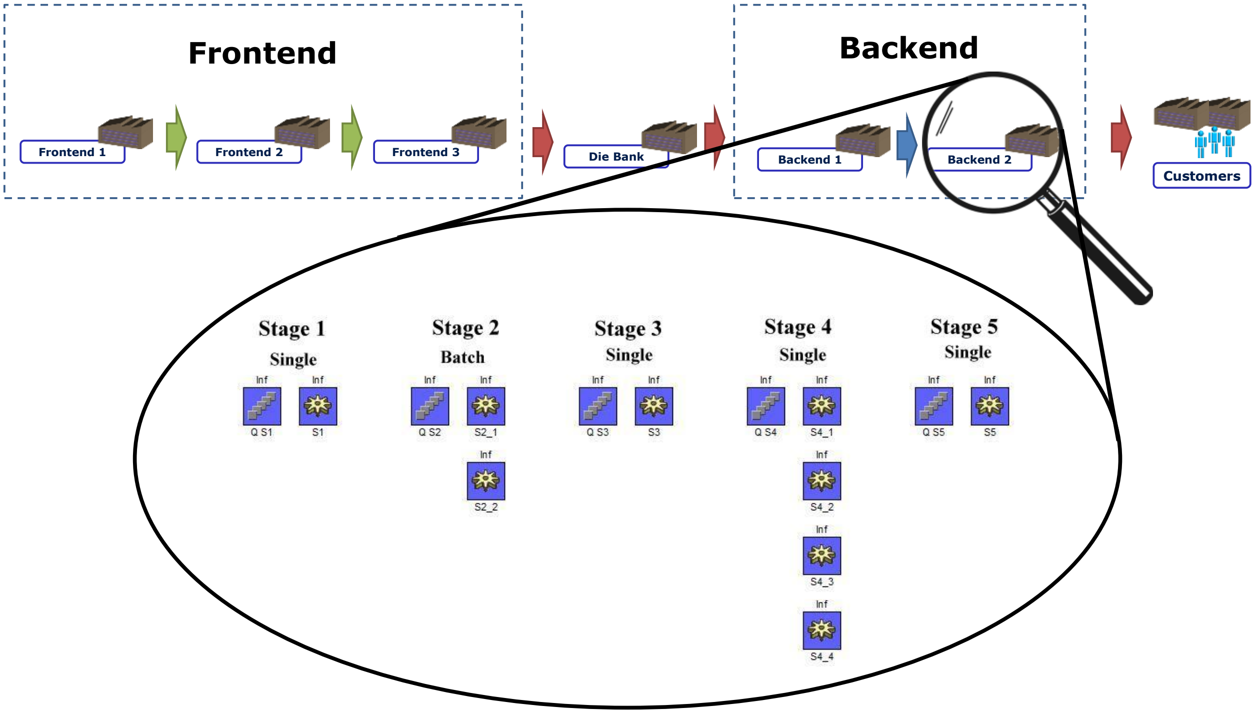


Figure 2: Schematic view of the investigated simulation model.

The test model consists of a time horizon of twelve weeks. There exist setup times for stages two (60min) and four (10min) which occur if the product type changes on a machine. Each stage has a queue with unlimited capacity in front of the machines, so no machine can be blocked. Stage 2 consists of two parallel batch machines with a capacity of six jobs per machine. Stage 4 is a work center with four unrelated parallel single machines. The ten different products pass the stages in the order that is shown in Figure 2, but some products skip a stage, respectively stage two, three or four. Also the processing times differ between the products and also station 2 uses the product flag of a job for building up parallel batches.

For generating the test data it was considered that the workload of each stage and the model is between 60% and 90% but as high as possible. There is also a correlation between the generated release and due dates as it is assumed that the release has to be in a certain period of time before the due date. Two different test sets have been build up. In the first test set, the release dates are distributed between 5 and 7 days before the due date. In the second test set, this time span is between 8 and 20 days before the due date. The granularity of the due dates is only on daily basis. This means a due date can only be scheduled at the end of a day. The generated data are uniformly distributed and are sorted in orders. Orders contain a certain number of jobs with the same product type, release and due date. Each order consists of 4 up to 20 jobs. All in all 145 orders were generated with 1600 up to 2100 jobs for the time horizon of twelve weeks. 100 different test instances with different orders were generated per parameter combination.

### 4.2 Implementation

For the simulation model the dispatching rules Earliest Due Date (EDD) and Batch Apparent Tardiness Cost (BATC, see Mason et al. 2002) were used. Both dispatching rules tend to minimize tardiness. As a solver for the CP model, the IBM ILOG CPLEX optimization studio was used. Here two different objective functions are tested. These functions are already described in equations (4), respectively (5) and (6). The first equation describes the approach for a weighted sum. This is described in the results as "sum". The second approach (equations (5) and (6)) describes the lexicographic optimization, where first the tardiness is optimized and with minor prioritization the weighted sum of earliness and cycle time. This approach is described in the results as "staticLex".

The first attempt was to solve the optimization problem for the whole period of twelve weeks but it turns out that the problem is too big to solve it at once. So the period has to be portioned in certain overlapping time intervals of 25 days where the schedule of the first 5 days was taken and then the next 25 days were optimized. A schematic view of this method is shown in Figure 3.

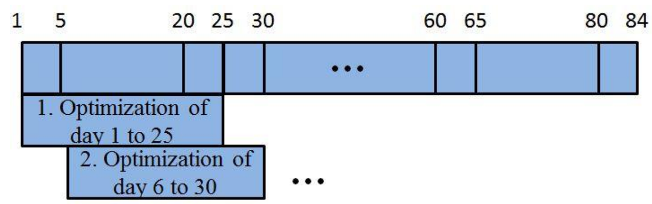


Figure 3: Schematic view of the decomposition scheme.

After testing and analyzing the data of the full model (FM) it turned out, that either stage three or stage 5 appear to be the bottleneck of the model. Because of these observations a bottleneck model (BM) was developed, where the first two stages are substituted with a delay. The delay was determined by a simulation that was done before the optimization process. This process made the model a lot easier, because during the optimization process no batches had to be calculated and less stages had to be planned, so all in all the number of process steps had been decreased.

All in all the following methods are used to calculate the resulting job start dates:

- Simulation model
- Constraint programming model
  - Bottleneck model (BM) and full model (FM)
  - Objective functions "staticLex" and "sum"

After the job start dates are computed by the constraint programming model, an additional simulation run was done to test these start dates in the simulation model. For this testing, an additional buffer of one day was implemented. This buffer schedules the calculated start date one day earlier (or as early as possible regarding the original input data). This is done because the schedules generated by the CP modelling approach are very tight. Due to this the simulation model cannot generate such an efficient schedule. Afterwards the simulation model is started using these job start dates and the results are compared.

## 5 RESULTS

The results for all methods described in the last section are shown here. As already described, several methods are compared. Here results for the objectives tardiness, earliness and cycle time are presented. Also in the tardiness result the number of tardy jobs are shown. Furthermore, the results are divided for the different job release dates (5 to 7 days and 8 to 20 days before job due date).

In all cases the result from the simulation can be used as reference. For all result figures the used methods are displayed on the x-axis. Thereby the lower line shows the method (i.e. "BM" – bottleneck model, "BM+Simulation" – bottleneck model with downstream simulation). The upper line displays the used objective function ("staticLex" or "sum") for the mathematical methods. Typically the results gained directly from the bottleneck model using CP are only approximate values, because the non-bottleneck stations are only considered with a delay time. Using the downstream simulation model, the whole model without delays is used again and the results from CP model are verified for the whole problem.

### 5.1 Release Dates 5 to 7 Days before Due Date

In this section the results for the test set where the jobs have a release date between 5 and 7 days before the due date are shown. Due to this short time between the release and due date here the tardiness is typically higher than in the other test set. In Figure 4 the average tardiness as well as the average number of tardy jobs are presented.

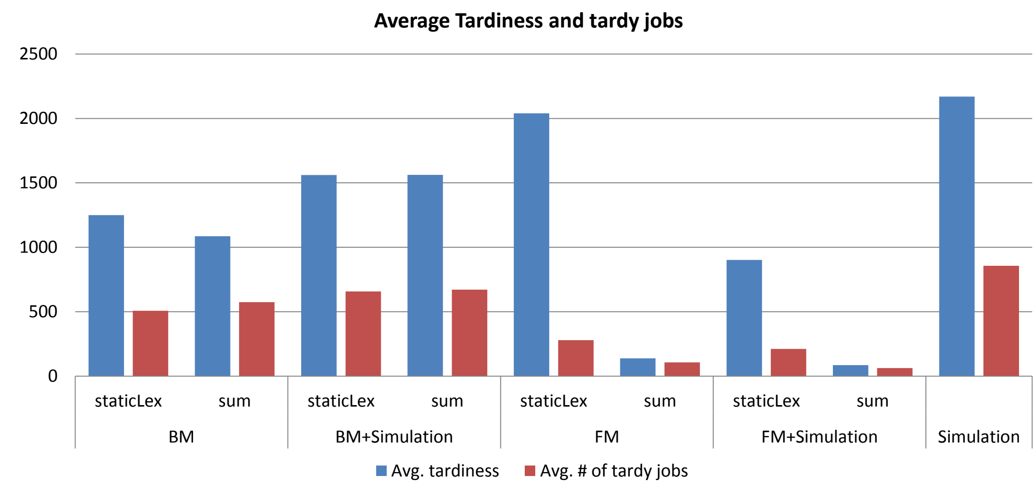


Figure 4: Average tardiness and number of tardy jobs for first test set.

The results show that for this combination the full model with objective function "sum" gets good results compared to the other methods. Thereby the tardiness gained from CP results without downstream simulation is a little bit higher than the results in downstream simulation. This is due to the included buffer of one day. The bottleneck model does not perform as good as the full model in this case, but both mathematical approaches outperform the result gained by the simulation model, although the dispatching rules in the simulation model are designed to minimize tardiness.

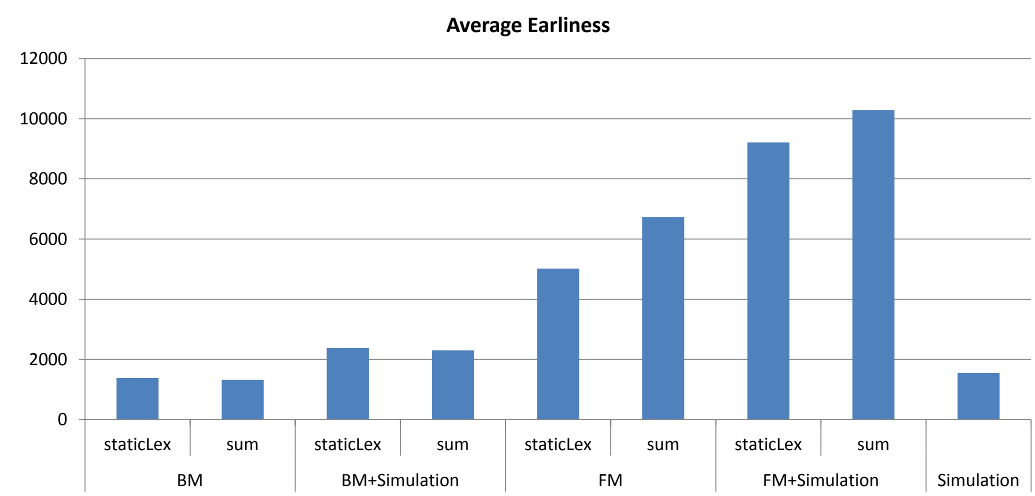


Figure 5: Average earliness for first test set.

Figure 5 shows the results for the average earliness. Here the simulation of the bottleneck model produce less earliness than the full model.

The result regarding cycle time is presented in Figure 6. Here the bottleneck model gets slightly better results than the simulation and the full model.

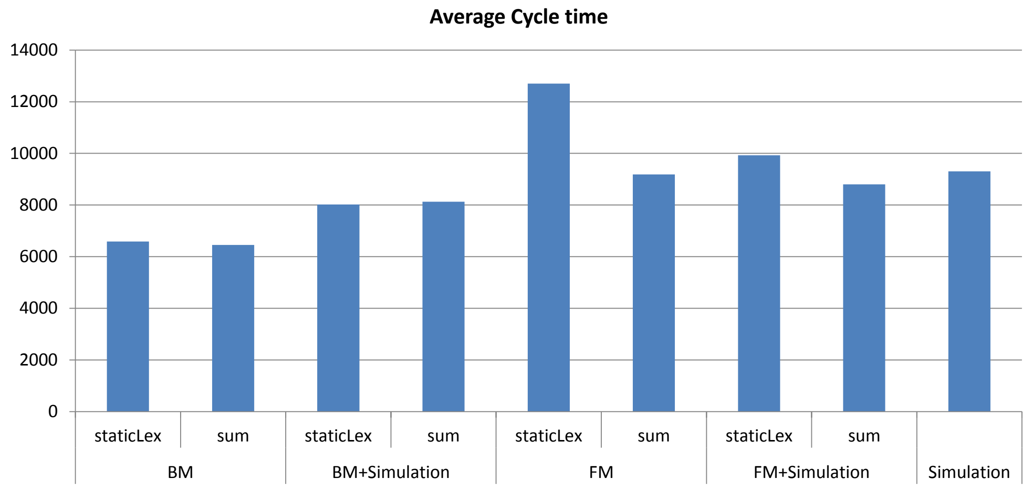


Figure 6: Average cycle time for first test set.

In summary, for this test set the mathematical methods outperforms the simulation. Overall the full model with summed objective function gains best results if the tardiness is high prioritized.

## 5.2 Release Dates 8 to 20 Days before Due Date

In this section the results are shown in the same manner as in the section before. This time the test set with release days between 8 and 20 days before due date is tested. The first result in Figure 7 again shows the average tardiness and number of tardy jobs for each tested method. In contrast to the first test set here the full model with summed objective function gains much worse results. The results from "staticLex" objective function are much better. But for this test set it shows that the bottleneck model has slightly better results than the full model. This is due to the reduced complexity for this test set.

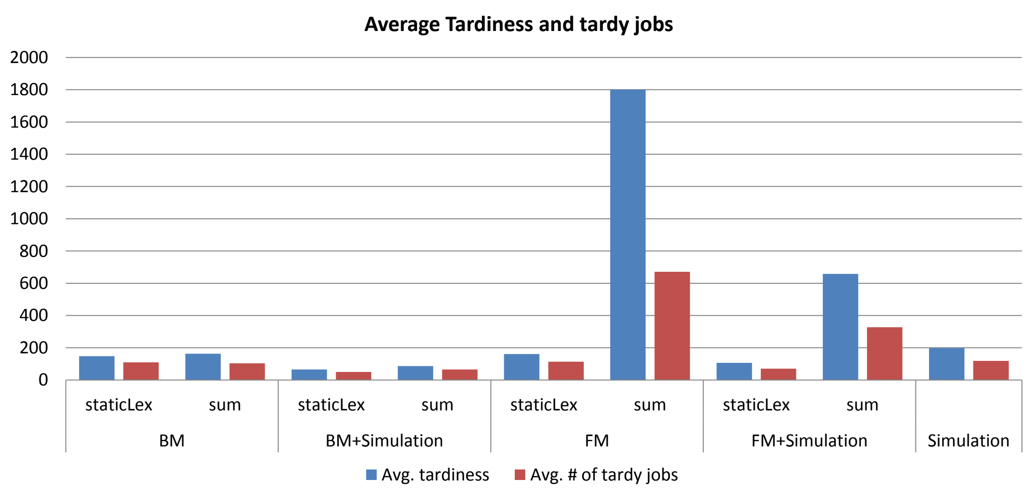


Figure 7: Average tardiness and number of tardy jobs for second test set.

In Figure 8 the results for the earliness are presented. Here the bottleneck model without downstream simulation gets the best results. Also the effect of the additional buffer of one day for the downstream simulation can be seen very good. This additional buffer results in more worth values for the earliness.

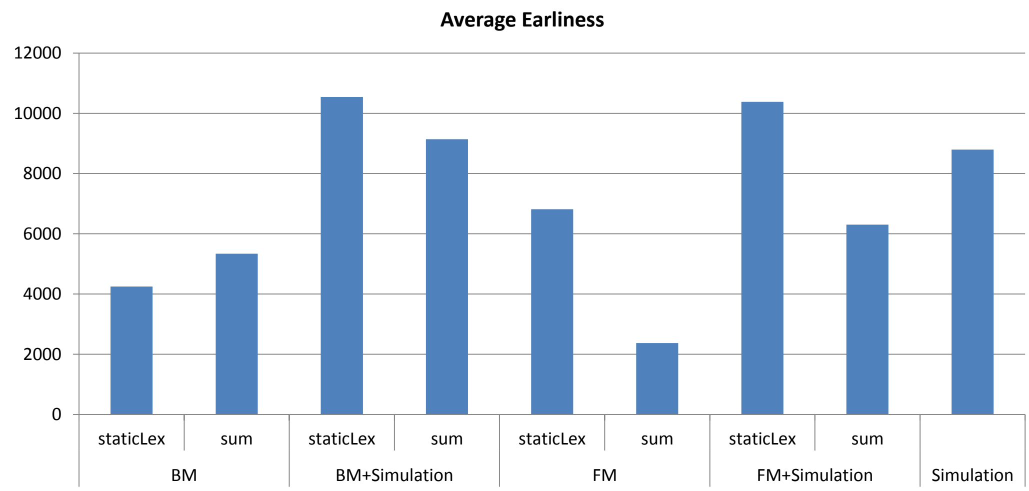


Figure 8: Average earliness for second test set.

The last result is shown in Figure 9. Here the result of average cycle time shows a very good performance of the mathematical methods. This is due to a much better balanced work flow. Here the benefit of a scheduling method compared to simulation method is shown very good. The scheduling method also comprises time and therefore has a better overview of the work and could balance this work much better over time.

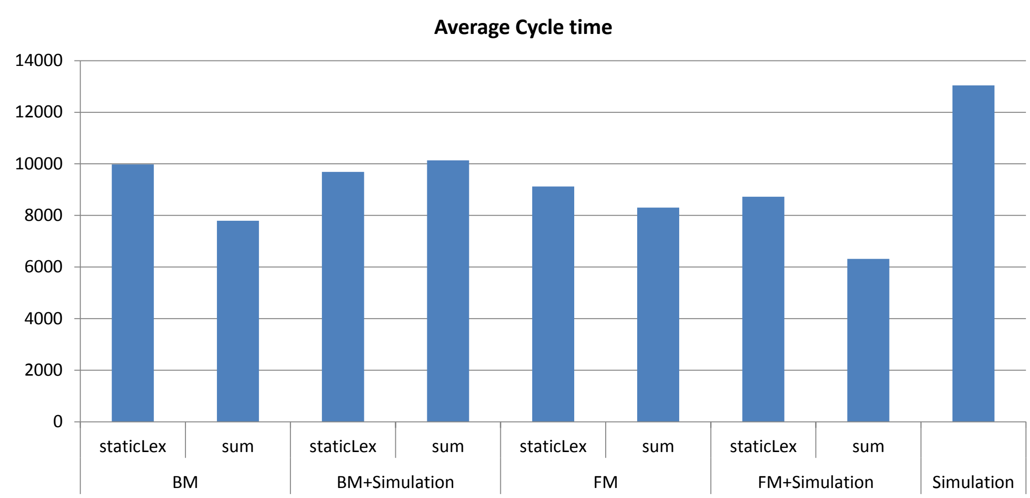


Figure 9: Average cycle time for second test set.

All in all the results show that the mathematical approach can outperform the simple simulation for nearly all objectives.

## 6 CONCLUSION AND OUTLOOK

In this paper a mathematical model for getting release dates by given due dates was introduced. For testing this mathematical approach a simplified production line of a backend test facility from semiconductor industry is used. Due to the complexity of the problem the mathematical model is also implemented with a bottleneck model. Additionally, two different objective functions – staticLex and sum – for the mathematical model were tested. To compare the results from the mathematical models also the calculated release dates were used in a downstream simulation.

Two different test sets were used. In the first test set the original release dates are between 5 and 7 days before due date. This short time between release date and due date automatically generates tardiness which should be minimized. In the second test set the original release date are 8 to 20 days before the due dates for each order. So here the tardiness is much smaller than in the first test set.

The results show that the mathematical approach could outperform the simulation model using dispatching rules for all objectives (tardiness, earliness and cycle time). Thereby the dispatching rules used in the simulation model tend to minimize tardiness. Due to the nature of discrete event simulation also cycle time should be minimized because only non-delayed schedules are generated.

The work done in this paper is only a first investigation for this field of supply chain scheduling. More work should be invested for bigger models and coupling of several fab models where each fab uses the release date of the following fab as due date. So it might be possible to calculate good release (and due) dates for each fab within a supply chain by knowing the customers due date. Also the bottleneck detection and planning within the mathematical approach should be enhanced. Furthermore a coupling between mathematical model and simulation models, for example for bottleneck calculation, could be used.

## ACKNOWLEDGEMENT

The work has been performed in the project eRamp (Grant Agreement N°621270), co-funded by grants from Austria, Germany, Slovakia and the ENIAC Joint Undertaking.

## LITERATURE

Ehm, H., H. Wenke, L. Mönch, T. Ponsignon, and L. Forstner. 2011 ."Towards a Manufacturing Supply Chain Simulation Reference Model for the Semiconductor Industry." In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu eds., 2124-2135. Pheonix, Arizona.

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

Kromer, M. 2005. *Leistungsbewertung und Engpassanalyse von Fertigungssystemen der Elektrotechnik*. master thesis, TU Dresden.

Lange, J., D. Doleschal, G. Weigert, and A. Klemmt. 2014. "Scheduling Preventive Maintenance Tasks with Synchronization Constraint for Human Resources by a CP Model Approach." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller eds., 2454-2465. Savannah, Georgia.

Lee, Y. H., S. Chung, B. Lee, and K.H. Kang. 2006. "Supply Chain Model for the Semiconductor Industry in Consideration of Manufacturing Characteristics." In *Journal of Production Planning & Control*, vol. 17, no. 5, 518-533.

Mason, S., J.W. Fowler, and W.M. Carlyle. 2002. "A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops." In *Journal of Scheduling 5*, 247-262.

Wang, Z., J. Chen, and Q. Wu. 2008. "A New Method of Dynamic Bottleneck Detection for Semiconductor Line." In *Proceedings of the 17th World congress The International Federation of automatic control*, vol. 17, no 1, 14840-14845.

Yang, K., Y. Chung, and S.C. Park. 2014. "Short-Term Bottleneck Detection for Process Planning in a FAB." In *International Journal of Computer and Communication Engineering*, 442.

Zang, H., Z. Jian, and C. Guo. 2009. "An Optimized Dynamic Bottleneck Dispatching Policy for Semiconductor Wafer Fabrication." In *International Journal of Production Research*, vol. 47, no. 12, 3333-3343.

## AUTHOR BIOGRAPHIES

**DIRK DOLESCHAL** studied mathematics at Dresden University of Technology, Germany. He obtained his degree in 2010 in the field of optimization. He has been a Research Assistant at Electronics Packaging Laboratory of the Dresden University of Technology since 2010 and works in the field of production control, simulation & optimization of manufacturing processes. His email is Dirk.Doleschal@tu-dresden.de

**GOTTFRIED NIEKE** studies mathematics at Dresden University of Technology, Germany. Currently he is working on his master thesis. This paper refers to this work. His email is g_nieke@avt.et.tu-dresden.de

**GERALD WEIGERT** is an Assistant Professor at Electronics Packaging Laboratory of the Dresden University of Technology. Dr. Weigert works in the field of production control, simulation & optimization of manufacturing processes, especially in electronics and semiconductor industry. He was involved in development of simulation systems as well as in its application in industrial projects for scheduling. His email is Gerald.Weigert@tu-dresden.de.

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