Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

YIELD INTEGRATED SCHEDULING USING MACHINE CONDITION PARAMETER

Dirk Doleschal
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

Currently machines in a parallel work center in semiconductor manufactory are assumed uniform in terms of impact on yield for most logic to dispatch schedule this machine set. But in reality machines are different even though they are allowed for the same products. In some layer forming areas machines can get a so called health parameter which describes the current condition of the machine. A high health value means, that defects produced with a machine are less probable. Also the products, which are processed at this work center, differ in their complexity and wafer area used for one chip. The goal is to schedule products with a high complexity and a larger chip size to those machines with the best health value. Doing so will minimize defect wafer area. For this, different dispatching rules and a mixed integer programming approach are compared within a simulation model for practical test data.

1 INTRODUCTION

The improvement of yield is an important goal in a semiconductor manufactory. Since the production of today's chips in the semiconductor manufacturing partially takes up to three months and several hundred process steps (Potoradi et al. 2002; Yurtsever et al. 2009) until the completion of each layer, there is a large potential for optimization here. To optimize yield by scheduling, different approaches already exist in literature. For example Wein (1992) investigated the correlation between yield and cycle time. Other approaches for example try to optimize time dependencies between product steps (Klemmt and Mönch 2012). Furthermore, preventive maintenance is scheduled (i.e. Lange et al. 2014) to hold the machines within a stable health. But even when the machines get regular maintenance the machines perform different. This machine performance could be described by a health factor which also could be used in scheduling. Currently no literature was found in which the influence of machine health parameter to the scheduling was investigated.

In this paper we try to establish a yield integration into scheduling, where machine dependent parameter are used. These machine parameter give information about the actual health condition of a machine. In our case this is typically useful for layer forming processes within the semiconductor manufactory. This could be for example PVD (physical vapor deposition), CVD (chemical vapor deposition) or lithography step. Within these layer forming processes it may happen that a local defect on a wafer occurs. The result from this local defect is that all affected chips on this wafer are defective. Now our approach is that important products should be processed on machines with a high health value. The definition of important products thereby could be products with a high chip size per die on the one hand and products with a high number of complex layers on the other hand. The reason why the important products should be processed on a well performing machine is shown in Figure 1.

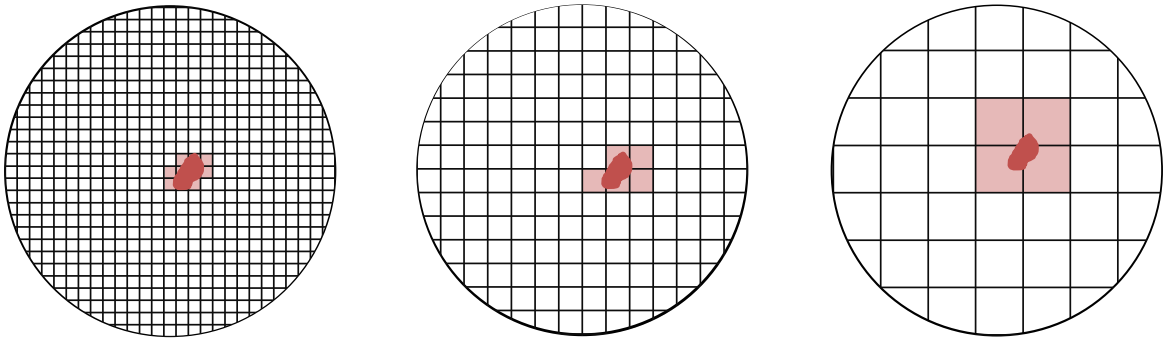

Figure 1: Affected wafer surface for a local defect.

In this figure a local defect is shown on three different wafers with different chip size per die. Here, the affected wafer area is much higher for products with a high chip size compared to products with a small chip size. Due to this fact, our goal is to compare different types of dispatching rules and a mixed integer programming approach which take care of the machine health parameter. In this work we do not calculate the health value. Also no direct connection between the health value and the yield is used. Furthermore, no direct real data is used. Instead of this, practical test data was generated.

The paper is structured as follows. In section 2 the problem is described. Section 3 is used to present the dispatching rules and define the mixed integer programming models. In section 4 the experimental setup is presented and afterwards in section 5 the results are shown. An outlook and conclusion is done in section 6.

2 PROBLEM DESCRIPTION

The underlying work center problem was derived from a practical point of view. A work center with unrelated parallel machines was used as reference. Also an amount of jobs was created, whereby each job is assigned exactly to one product. Also each job has a release date and an operational due date, which defines the due date for the current operation. The defined products have a dedication matrix. This matrix contains the allowed machines for each product. Furthermore, the products are divided into 2 groups – important products (IP) and normal products (NP). This classification is used for the yield integrated scheduling and the resulting objective function. The processing times differ for each product, whereby the processing time is equal for one product and different allowed machines. Each machine has a health value, which defines the actual health state of this machine. In this investigation this value is defined between zero and one. The health value is equal for all products processed on this machine. The health value is also assumed to be constant during the whole time horizon. This is done because of the simplicity of the model.

The schedules, which are used to calculate the objectives, are generated with a discrete event simulation model (c.f. Figure 2). All properties, which are described within section 2, have been implemented in the simulation model.

Also several dispatching rules are included within this simulation model. These different dispatching rules are compared with a mixed integer programming approach. Furthermore, an interface was implemented where the results from a mixed integer programming (MIP) model could be used.

The observed objectives in this investigation are

- Flow factor – The ratio of the cycle time to the raw processing time
- Tardiness – The sum of delay for late jobs. Early jobs get a tardiness of zero
- Quality – This is defined as the sum of the machine health value for each important job processed on the corresponding machine

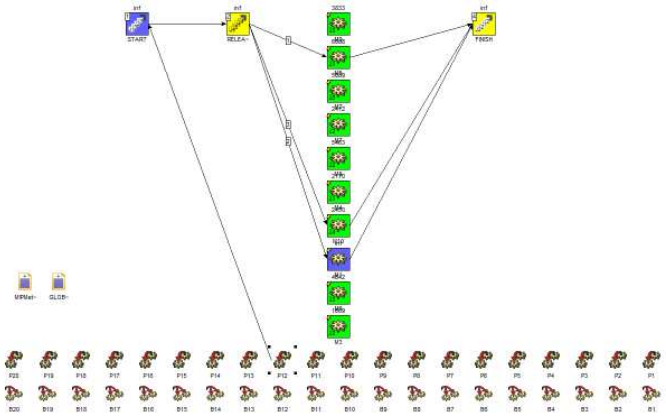

Figure 2: Exemplary simulation model (simcron MODELLER).

3 METHODS

3.1 Dispatching rules

As a reference, three different dispatching rules with different complexity are implemented:

- Operational due date - ODD
- Best Machine - BestM
- Practical rule - DePrio

The ODD rule is the simplest one. All jobs are ordered using their local operational due date. If a machine gets idle, the next allowed job regarding the due date is scheduled on this machine. Here no difference between important and non-important jobs is done. Also the health value of the machine is not observed. This rule is investigated by (Rose 2003) and is usually used to minimize tardiness.

The next dispatching rule is called “Best Machine”. Here for all important products only the machine with the highest health value is allowed. All other machines which are released for this product in the dedication matrix get locked. So these products could only be processed on the “best” machine out of the set of allowed machines. Furthermore, the sorting of the jobs is also done by the ODD rule. Additionally, all jobs of the important products get a priority state. This means, these jobs have a higher priority compared to the normal jobs. This is done because of the hardly reduced dedication matrix for the important products. Due to the nature of this dispatching rule, the result concerning the quality of the jobs is an upper bound for this objective.

The last used dispatching rule is a practical orientated rule called DePrio. Due to the confidentiality the rule is not described in detail here. It works with priority based dispatching lists for every machine. Here also the ODD rule is used as basis. Furthermore, the jobs get an additional priority in the case of important products. But the priority is reduced for important jobs depending on the actual load and machine health value. So in contrast to the basic ODD rule, where one global dispatching list exists, each machine has an own dispatching list now, where the jobs are sorted by their priority value.

Because mathematical methods for scheduling get more and more practicable, also two mixed integer programming scheduling approaches were investigated, which is described in the following.

3.2 Mixed integer programming

Two mixed integer programming approaches are tested. The background of both mathematical models is equal, but the implementation and the coupling with the simulation model differs between both models. Both mathematical models are based on a capacity planning. In the following the models are called MIPv1 and MIPv2 for version 1 and version 2.

3.2.1 Input parameter

For describing the implementation of the mathematical model the input parameter should be explained first. These input parameters are retrieved from the simulation model and could be divided into dynamic and static parameters. This means, a static parameter does not change within the time horizon – in contrast to a dynamic parameter.

Static parameters

- n different products P_i (i=1,…,n)
- m unrelated parallel machines M_k (k=1,…,m)
- Dedication matrix $D \in \{0,1\}^{n \times m}$. Also $D_k := \{ i \mid D_{i,k} = 1 \}$ is the set of products permitted for processing on machine M_k. In the same manner, $D_i := \{k \mid D_{i,k} = 1\}$ is the set of machines permitted for processing products P_i
- $pt_{i,k} > 0$ is the processing time for a job of product P_i on machine M_k if $D_{i,k} = 1$,
- Machine health value $mh_k \in (0,1]$ $(k = 1,...,m)$ for each machine
- Set of important products $IP \subset \{1,...,n\}$

Dynamic parameters

- Remaining machine processing time $rpt_k \in \mathbb{N} (k = 1,\ldots,m)$
- Job volume $v_i \in \mathbb{N} (i = 1,\ldots,n)$

Additionally weighting parameters ω_1 and ω_2 exist, which are used to weight the MIP objective function.

3.2.2 Variables

The used decision variables in the mixed integer models are the following:

$X_{i,k} \in \mathbb{R}^+/\mathbb{N}$	amount of jobs from product P_i assigned to machine M_k (k=1,…m; $i \in D_k$)
$Y_{i,k} \in \{0,1\}$	Boolean matrix, which defines whether a product i is scheduled on machine j or not.
$C_{\max} \in \mathbb{N}$	maximum makespan for all machines
$G_k \in \{0,1\}$	Boolean vector, which defines whether a machine is used by an important product or not
$H_k \in \{0,1\}$	Boolean vector, which defines whether a machine is used by a normal product or not

Thereby X is a positive real variable in the first MIP model and a positive natural number in the second model. With the defined input parameters and variables the created mixed integer models could be described.

3.2.3 MIPv1

In this first approach, the underlying mixed integer programing model was used to reduce the dedication matrix in the simulation model. To describe the objective function in this mathematical model, the following substitutions are used:

Quality: $$Q = \sum_{\substack{i \in IP \\ k \in D_i}} X_{i,k} \cdot mh_k \tag{1}$$

Number of allowed machines for normal products: $$M_{NP} = \sum_{\substack{i \notin IP \\ k \in D_i}} Y_{i,k} \tag{2}$$

$$\text{Product overlap: } O = \sum_{k=1}^{m} G_k \cdot H_k \tag{3}$$

Now the mathematical model could be described. In this version the variable X is a real variable greater than 0:

$$\omega_1 \cdot C_{\max} + \omega_2 \cdot (-Q + O - M_{NP}) \rightarrow \min \qquad \text{subject to} \tag{4}$$

$$\sum_{k \in D_i} X_{i,k} = v_i \qquad i \in \{1,...,n\} \tag{5}$$

$$K \cdot Y_{i,k} \geq X_{i,k} \qquad i \in \{1,...,n\}; k \in D_i \tag{6}$$

$$X_{i,k} \geq Y_{i,k} \qquad i \in IP; k \in D_i \tag{7}$$

$$rpt_k + \sum_{i \in D_k} X_{i,k} \cdot pt_{i,k} \leq C_{\max} \qquad k \in \{1,...,m\} \tag{8}$$

$$K \cdot G_k \geq \sum_{\substack{i \in D_k \\ i \in IP}} Y_{i,k} \qquad k \in \{1,...,m\} \tag{9}$$

$$\sum_{\substack{i \in D_k \\ i \in IP}} Y_{i,k} \geq G_k \qquad k \in \{1,...,m\} \tag{10}$$

$$K \cdot H_k \geq \sum_{\substack{i \in D_k \\ i \notin IP}} Y_{i,k} \qquad k \in \{1,...,m\} \tag{11}$$

$$\sum_{\substack{i \in D_k \\ i \notin IP}} Y_{i,k} \geq H_k \qquad k \in \{1,...,m\} \tag{12}$$

This model has the task to minimize the maximal makespan and to optimize the quality parameter. Thereby it is triggered more than once within the simulation model (c.f. section 3.2.5). As seen above, the quality is simply described as the sum of the machine health values for each important job. So the second part of the objective function (4) has the goal to optimize this quality objective and additionally to minimize the product overlap for the machines as well as to maximize the number of allowed machines for the normal products. This is done because the so reduced dedication matrix Y is directly used within the simulation model. The parameters ω_1 and ω_2 are used to weight the two main goals of the objective function. Such an implementation is also described in Klemmt (2012). Equation (5) is used to assign all jobs to a machine. (6) ensures that $Y_{i,k}$ is 1 if $X_{i,k} > 0$. Therefore, K is a big number. For important products also the other direction is done with equation (7). With (8) the maximum workload over all machines is calculated. Equations (9) and (10) are used to define whether an important product is scheduled on a machine or not. In the same way, this is done for normal products with the help of (11) and (12) .

So, with this implementation we try to help the simulation model by assigning the jobs to the machines. Due to the reduced dedication matrix the important products should be scheduled on good machines with as less overlap with other products as possible.

3.2.4 MIPv2

In contrast to the first version of the mathematical model, here the dedication matrix is not reduced. In this approach the capacity planning model is more integrated into scheduling. The result from the mathematical model is an assignment of a number of jobs to machines. Due to the fact that the mathematical model is still a capacity planning model, this is done without knowledge of due dates or release dates. In the simulation model, this result is used to assign the number of jobs to the machines.

The order in which the jobs are processed is still defined by the simulation model using the ODD rule. Here the mathematical model could be written much easier. So, the only substitution which is made equal to version 1 is the quality objective (1).

In this second version the variable $X_{i,k}$ is defined as positive integer. This is done because the result from X is directly used in the simulation model and the jobs could only be assigned in the whole to the machines and cannot be divided into smaller jobs.

Now, the optimization model can be formulated as following:

$$\omega_1 \cdot C_{\max} - \omega_2 \cdot Q \rightarrow \min \qquad \text{subject to} \tag{13}$$

$$\sum_{k \in D_i} X_{i,k} = v_i \qquad i \in \{1, ..., n\} \tag{14}$$

$$rpt_k + \sum_{i \in D_k} X_{i,k} \cdot pt_{i,k} \leq C_{\max} \qquad k \in \{1, ..., m\} \tag{15}$$

(13) is the objective function. The parameters ω_1 and ω_2 are also used to weight both objectives. (14) is used to ensure that all jobs are assigned to machines. With (15) the makespan $C_{\max}$ is calculated for the whole machine pool. For this the assignment matrix X and the remaining processing times *rpt* are used.

3.2.5 Result from mathematical model and implementation

The results from these mathematical models are used within the defined simulation model. For this a user specific script code has been implemented within the simulation model, which is needed to trigger the mathematical optimization run and to transpose the result to the simulation model as shown in Figure 3.

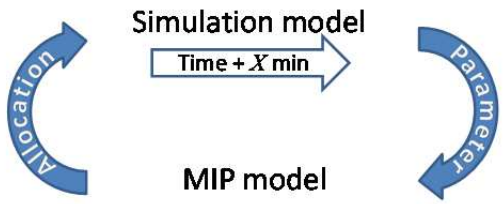

Figure 3: Dynamic coupling between MIP model and simulation model.

The MIP model is calculated every 10 (simulation) minutes. The MIP model also gets a forecast of 10 minutes for incoming jobs. This means, all jobs with a release date within the next 10 minutes are considered. In the first MIP version the implementation of the reduced dedication matrix is done directly in the simulation model. The consideration of the results of the second MIP version within the simulation model is done via product volume lists for each machine. So, each machine gets the information how many jobs of a product are allowed to be processed. The sequence of the jobs is done by the simulation model using the described ODD rule.

4 EXPERIMENTAL SETUP

To test the presented methods, a set of test instances is generated. Table 1 gives an overview for the used parameters.

Table 1: Experimental setup (UD - uniform distribution).

Factor	Values used	Total values
Number of products n	20	1
Number of jobs per product n_i	UD ~ [100; 500]	1
Number of machines m	10	1
Minimum machine health	0.1; 0.4; 0.7	3
Number of important products	1; 2; 4	3
Product dependent processing times rpt_i	UD ~ [1h; 3h]	1
Release dates rdd	UD ~ $[0, \frac{1.03}{m} \cdot \sum_{i=1,..,n} rpt_i \cdot n_i]$	1
Operational due date odd	$odd = rdd$ + UD ~$[2 \cdot \overline{rpt}; 10 \cdot \overline{rpt}]$	1
	Number of independent instances	50
	Total number of problems	450

The unrelated parallel machine work center consists of 10 machines, processing 20 different products. The number of important products differs between 1 and 4, where the corresponding products are chosen randomly. The maximum machine health is always 1. The minimum health varies between 0.1 and 0.7. The processing times for each product are chosen randomly between one and three hours. The release dates for all jobs are distributed between 0 and 1.03 * minimum capacitive makespan. The value "1.03" is calculated experimentally to gain an average flow factor of about three using the ODD rule. The operational due dates are distributed using the average processing times.

On the basis of these test instances, the presented methods are tested and results are generated.

5 RESULTS

In this section the results for the used methods and the observed objectives are presented. Thereby, the parameter ω_2 for the MIP model is chosen out of the set {1; 5; 25; 50}. The parameter ω_1 is 1.

In Figure 4 the results for all test instances and the quality objective are shown.

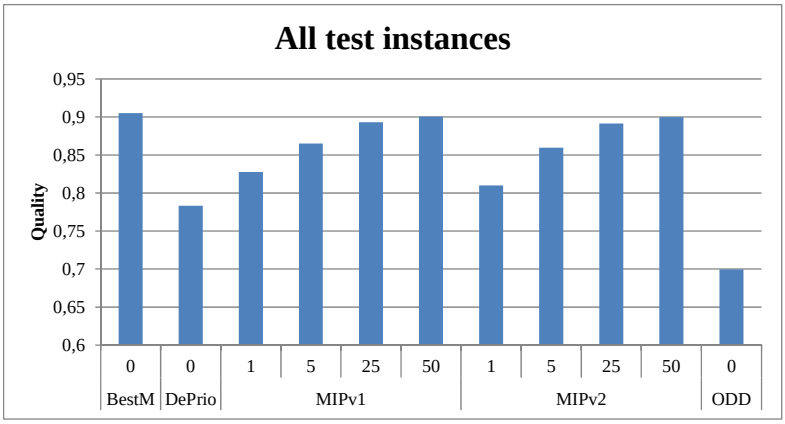

Figure 4: Results for all test instances regarding quality objective.

As expected, the Best Machine dispatching rule generates the highest quality value. The result of the MIP methods depends on the used parameter ω_2. Also, the ODD rule has the lowest quality result.

The results for the tardiness and the flow factor are divided into three bars for each method. The first bar shows the result for all products. In the second bar only the important products are considered and vice versa in the last bar only the result for the normal products are presented. The x-axis has two rows. The first row describes the parameter ω_2 and the second row the used method.

In the next figures the results for the tardiness objective are presented. In Figure 5 the results for all test instances are shown. Thereby, the y-axis is cut at 10, because partly results are much higher. As seen in this result, the Best Machine rule and the mixed integer approaches with a high parameter ω_2 generates poor results regarding tardiness. This is due to the reduced dedication matrix especially for these test instances with a high number of important products. The DePrio rule performs a little bit poorer than the ODD rule for important products. This rule could also be parameterized. However, this has not been investigated until now.

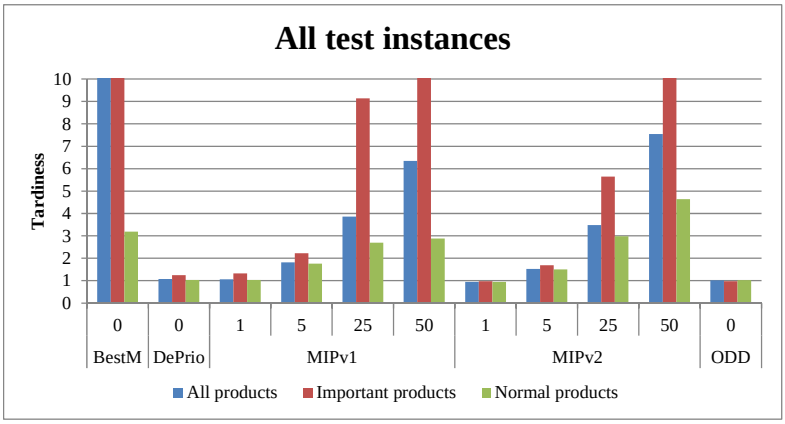

Figure 5: Result for all test instances regarding tardiness (y-axis cut off).

To show the influence of the number of important products, in Figure 6 the results for 1 and 4 important products are presented. Thereby, this time the y-axis is not cut off. This result shows that the choice of parameters is important.

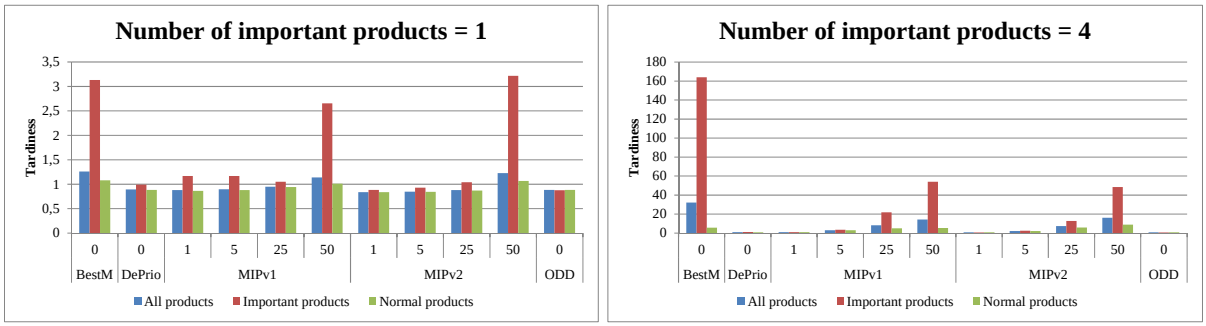

Figure 6: Results for tardiness for one and four important products.

The last investigated objective is the flow factor. The results are shown in Figure 7.

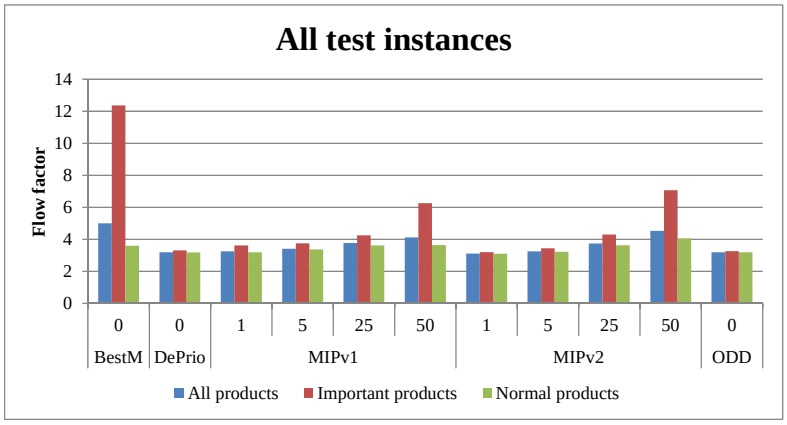

Figure 7: Result for flow factor for all test instances.

Here a similar result as seen for tardiness could be estimated. Also the minimum health values for the machines have an influence on the results. So in Figure 8 the result for a minimum health value of 0.7 is shown.

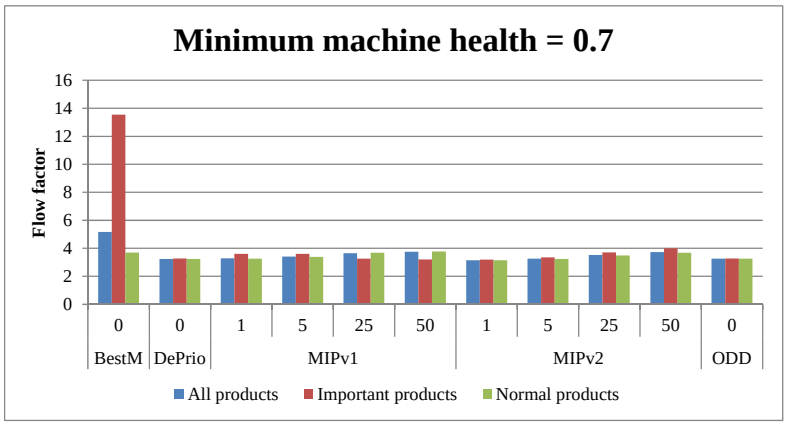

Figure 8: Result for flow factor and a minimum machine health of 0.7.

Here the deviation of flow factor is much smaller than in Figure 7. This can be explained by the objective function of the MIP model. In the case where the deviation of the minimum health is smaller, also the possible optimization for quality objective is smaller and so the influence of parameter ω_2 drops.

In summary, the presented results show that the results from the mixed integer programming approaches hardly depend on the parametrization. The extreme values of 25 and 50 for ω_2 gain results close to the optimum for quality, but also perform worse results for the tardiness and flow factor. The Best Machine rule is only useable to get the upper bound for the quality. All in all the mixed integer approaches even with a low ω_2 outperform the practical orientated dispatching rule DePrio. Also for the two other objectives the results are practicable for these mathematical models.

6 CONCLUSION AND OUTLOOK

In this paper a yield integrated scheduling method, which uses a machine health parameter, is implemented. For this, different dispatching rules and mixed integer programming models are compared with the well-known ODD dispatching rule. The results show that the mathematical approach gains slightly better results compared to the practical dispatching rule. Further, an improvement in the quality often concludes to a worsening of other objectives. The results in our study show an improvement in

quality by an average of up to 20% compared to the ODD rule. Here, the formulation used for the quality objective is just an abstract improvement of the yield. The result cannot be directly converted into yield. This has to be done for each problem area specifically. Overall, the presented investigation is a proof of concept for implementing machine health parameter to scheduling methods.

Further research regarding the parameterization of the practical dispatching rule and the mathematical approach should be performed. Also more complex test instances as well as real data should be used to get a better overview of possible improvement.

ACKNOWLEDGMENTS

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.

REFERENCES

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

Klemmt, A., and L. Moench. 2012. "Scheduling Jobs with Time Constraints between Consecutive Process Steps in Semiconductor Manufacturing." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 2173-2182. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lange, J., D. Doleschal, G. Weigert, and A. Klemmt. 2014. "Scheduling Preventive Maintenance Tasks weith Synchronization Constraints for Human Resources by a CP Modeling 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, 2454-2465. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Potoradi, J., O. S. Boon, S. J. Mason, J. W. Fowler, and M. E. Pfund. 2002. "Using Simulation-Based Scheduling to Maximize Demand Fulfilment in a Semiconductor Assembly Facility." In: *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1857–1861. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rose, O. 2003. "Accelerating Products under Due-Date Oriented Dispatching Rules in Semiconductor Manufacturing." In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 1346-1350. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wein, L.M. 1992. "On the Relationship between Yield and Cycle Time in Semiconductor Wafer Fabrication." In *Semiconductor Manufacturing,* 156-158

Yurtsever, T., E. Kutanoglu, and J. Johns. 2009. "Heuristic Based Scheduling System for Diffusion in Semiconductor Manufacturing." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 1677–1685. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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

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.