Simulation in Produktion und Logistik
Entscheidungsunterstützung von der Planung bis zur Steuerung
Wilhelm Dangelmaier, Christoph Laroque & Alexander Klaas (Hrsg.)
Paderborn, HNI-Verlagsschriftenreihe 2013

The Impact of Accuracy in Lot Arrival Prediction on Solution Quality for the Parallel Batch Machine Scheduling Problem in Wafer Fabrication

Genauigkeit in Losankunftvorhersagen bei der Losablaufplanung an parallelen Batch-Anlagen in der Halbleiterfertigung

Robert Kohn, Oliver Rose, Universität der Bundeswehr München, Neubiberg (Germany), robert.kohn@unibw.de, oliver.rose@unibw.de

Abstract: Solving Parallel Batch Machine Scheduling Problems (PBMSP) with dynamic arrivals profits to a large extend from information about future arrivals, which need to be predicted by using equipment models. Since any prediction comes with errors, we face the problem that scheduling systems may take decisions on the basis of erroneously predicted job arrivals. This study evaluates the impact of job arrival prediction errors on solution quality, respectively their influence on optimization potential. The experimental system solves the PBMSP with incompatible job families and dynamic arrivals using the Variable Neighbourhood Search (VNS) combined with the Time Window Decomposition (TWD). In the case of non-existing prediction errors compared to FIFO dispatching, the results show up to 10% improvements in total cycle time for look-ahead horizons around the average processing time. We observed a loss of up to 50% of the improvements for prediction error schemes where prediction errors are equal to the average processing time.

1 Introduction

The economic competitiveness of semiconductor manufacturers relies to a not negligible extend on an effective shop floor control system. Under constant pressure of cost reduction proper material flow control policies enable companies to manage production according to their goals, for example in terms of cycle times or customer delivery dates

Batch machines represent a significant proportion in semiconductor wafer fabrication and consequently significantly contribute to overall lot cycle times. Here a batch defines a quantity of jobs grouped together to be processed on a machine as one operation. Managing cycle times, work-in-progress (WIP), throughput and tardiness measures of batch machines have been continuously considered as points of interest for both research and industry. Usually furnaces that perform oxidation

and diffusion processes serve as representatives for batch machines, providing space for up to 200 wafers in their reactor. But also wet benches used for cleaning and etching the wafer surface process jobs as batches, where up to 100 wafers are processed together in one run.

Commonly the problem is to schedule WIP waiting in front of a set of parallel batch machines, where forming batches as well as sequencing and partitioning batches characterize the decision-making process. For that purpose and given that dispatching is still state of the art in production control, various dispatching heuristics have been proposed in literature. Earlier approaches focus on finding optimal or near-optimal batch sizes, bearing in mind that near-optimal batch sizes directly change with the level of utilization. Determining proper values for batch sizes particularly becomes problematic in a multi-product fabrication characterized by incompatible job families.

In order to overcome the problem of finding suitable batch sizes newer approaches incorporate future job arrivals, where the approach that includes information about next arriving jobs is commonly referred to as "look-ahead strategy". To provide the greatest leverage for effective dispatching and scheduling of batch machines the employing of upstream information into the decision-process is considered.

Beside dispatching strategies, scheduling approaches powered by optimization techniques continuously gain more importance for shop floor control. Because of rapidly emerging optimization methods and computer systems, scheduling approaches will likely play an important role within the next years and stepwise replace today's dispatching systems.

Scheduling solutions powered by optimization have been focused stronger than ever before in cause of effective search methods and constantly increasing computing power of modern computer systems. Research and industry work intensely on scheduling topics in order to replace (state of the art) dispatching rule based systems by more powerful scheduling solutions sooner or later.

Operational scheduling promises to realize optimization potentials in production logistics unreachable for common dispatching systems. Especially (meta)heuristics seem to be practicable to tackle complex scheduling problems of dimensions interesting for practitioners in industry. Here especially PBMSPs are in the focus of researchers and fab managers. Equivalent to related dispatching approaches, batch scheduling strategies profit extensively from look-ahead information, which provide predicted job arrivals over a certain time horizon.

Unfortunately, any prediction, regardless of the scope or intention, comes with errors - and so does job arrival prediction. Motivated by the fact that look-ahead information remarkably contribute to optimization potentials and by the fact that predictions are generally erroneous, we are particularly interested in the impact of look-ahead prediction errors on solution quality.

2 Literature Review

As a lead-in to the vast variety of literature dealing with scheduling in the area of semiconductor manufacturing operations, we like to refer to a recent survey given in Mönch (2013). In the following we will give a short review of the most important

batching heuristics. For a more detailed review of batch scheduling (and dispatching) one will find extensive overviews in Mathirajan (2006).

Glassey (1991) presents the Dynamic Batching Heuristic (DBH) that for the first time incorporates information about future job arrivals, referred to as "look-ahead" in the following text. Fowler (1992) introduces the more sophisticated Next Arrival Control Heuristic (NACH) that exclusively considers the next job arrival (cf. Fowler 2000, Solomon 2002). Van der Zee proposes the Dynamic Job Assignment Heuristic (DJAH) that also considers look-ahead information (cf. van der Zee 2007). Habenicht (2003) discusses batching heuristics combined with Genetic Algorithms (GA) in order to minimize tardiness measures. Balasubramanian (2004) presents the Batched Apparent Tardiness Cost (BATC) heuristic for weighted tardiness minimization, empowered by a search scheme based on GA. Gupta (2006) proposes a due-date oriented control strategy using look-ahead information in order to minimize maximum tardiness and the number of tardy jobs. Sha also discusses look-ahead dispatching heuristics (cf. Sha 2007). Moreover, several batching heuristics aimed to minimize total weighted tardiness are discussed in Kim (2010).

In the area of scheduling empowered with optimization methods, there exists a variety of approaches to solve PBMS problems under consideration of various subsets of constraints. Reichelt (2006) examines a Multi-Population Genetic Algorithm (MPGA) and a Multi-Objective Genetic Algorithm (MOGA) for the PBMSP with the goal to minimize TWT and makespan at the same time. In Wang (2010), Klemmt (2011) one will find MIP formulations for PBMSPs which aim to minimize the makespan or the total weighted tardiness. Only for makespan optimization, Wang (2010) presents a comparison between MIP and approaches based on Simulated Annealing (SA) and Genetic Algorithms (GA), both enhanced with multi-stage dynamic programming (MSDP). Various Evolutionary Algorithms (EA), respectively nature-inspired search schemes, based on the GA concept have also been published, e.g. Chiang (2010), Kashan (2008), Malve (2007), Mönch (2005), Balasubramanian (2004) and Habenicht (2003). Damodaran (2011) presents a GRASP and an implementation of the SA search in Damodaran (2012), where both approaches minimize the makespan. There exist approaches employing the ACO concept in order to minimize the total weighted tardiness (TWT) for PBMSPs (cf. Mönch 2009; Li 2008; Raghavan 2006). In Cakici (2013) heuristic algorithms employing VNS schemes are discussed and compared to a mathematical model developed for the PBMSP with dynamic arrivals and incompatible job families. Almeder (2011) studies popular metaheuristics applied to the PBMS with incompatible job families in order to minimize TWT. They examine variants of ACO, GA and VNS and compare their performance. Klemmt (2009) studies PBMS with incompatible job families and dynamic job arrivals, comparing MIP and VNS with respect to TWT minimization. Jula (2010) presents an algorithm based on Linear-Programming (LP), an approach based on Integer Programming (IP) and a heuristic-based algorithm to solve non-homogenous PBMSPs with non-identical job sizes and incompatible job families. Yugma (2008) presents a solution approach based on Simulating Annealing (SA) applied for PBMSPs improving throughput, batch efficiency and flow time factor.

3 Design of Experiments

3.1 Experimental system

The experimental environment primarily consists of a simulation-based optimization framework developed to solve PBMSPs existing in wafer fabrication front-end facilities. The underlying system covers various PBMSPs that differ in their sets of constraints and objectives. For searching improved schedules, we implemented a generalized concept of VNS, offering numerous VNS variants, including most of those mentioned in related literature. The difference between those variants is basically the balance between exploitation and exploration of the search space, beside countless different parameter combinations to choose.

We consider the developed scheduling system as ready-for-pilot on an operational level, meaning that we are able to load (and validate) a currently existing problem instance with actual data from fab databases. After loading the problem instance, the scheduling procedure creates an improved schedule, which is immediately written back to the manufacturing execution system (MES) for execution. In addition to implemented real-world and real-time features a model generator offers to create user-defined PMSP instances with specific characteristics. The model generator sets important model variables using random numbers following standard statistical distributions in order to generate a set of independent model instances that fundamentally show equal characteristics, but also differ slightly from each other.

We use a database to establish the data management which is necessary to run the experimental system in an effective (and comfortable) manner. The entire experiment data representing the input parameters (including the model instances) as well as the output results are accessible via database connections. The entire system is limited to PMSP instances that do not exceed 5GB in their compressed size, which equals to the maximum size for a Character Large Object (CLOB). Since large simulation (optimization) experiments often suffer from a lack of computing power and time availability, we delegate extensive studies to a High Performance Computing (HPC) cluster with 64 cores connected to the database. The entire system is written in C# with a greater focus on the code comprehensibility than on the speed of computation, so far.

3.2 Parallel Batch Machine Scheduling Problem (PBMSP)

We describe the scheduling problems examined in this study by the use of the α|β|γ-classification scheme proposed in Graham (1979). Under study the PBMSP incorporates unequal processing times, dynamic arrivals, job specific machine dedications, parallel batching with incompatible job families and arbitrarily maximum batch sizes for a job family on a machine, respectively $Rm|M_j, r_j, p\text{-}batch, incompatible, bmax|TCT$. The following list introduces the α|β|γ-notations used throughout the rest of the paper:

- *Rm:* unrelated parallel machines (with unequal processing times)
- M_j: machine dedications (a job is dedicated to a restricted set of machines)
- r_j : non-zero release date of a job (dynamic arrivals)
- *p-batch:* parallel batching (a number of jobs is processed simultaneously on a machine)

- *incompatible:* incompatible job families (jobs of different families cannot be processed together)
- $bmax_j$: arbitrarily maximum batch size for a job family on a machine
- $bmin_j$: arbitrarily minimum batch size for a job family on a machine
- tb_j: arbitrary time bound constraint for a job

For the experiments of this study we created numerous test instances, where each represents a PBMS problem with certain characteristics. The default model contains five machines and 500 jobs to be scheduled at the utilization level of 0.9, where the predicted job arrivals are uniformly distributed between zero and the expected makespan. Five incompatible job families are dedicated to five machines, which results in a square dedication matrix with 25 cells. Using a dedication density factor of 0.7 we create 17 to 18 combinations between a certain machine and a certain job family. The process time for each job family on a machine is uniformly distributed between 240 and 360 minutes, which results in average processing time equals to 300 minutes. The maximum batch size equals eight lots without exception. The default model is then extended with a certain arrival error scheme, which results in a model template with specific characteristics. Finally, we created 30 independent instances for each model template with specific characteristics, especially with regard to various arrival error schemes, which we intend to examine.

3.3 Variable Neighborhood Search (VNS)

Mladenović (1997) proposes the VNS heuristic based on neighbourhood structures used to solve large scale combinatorial problems. The simulation-based optimization framework that we use to solve PBMSPs employs VNS to create optimized schedules with respect to focused objectives. We implemented VNS as an abstraction of the proposed schemes, which allows us to freely configure two nested search levels. Both levels can be parameterized independently from each other, where each search level defines a set of neighbourhood structures: the local search procedure (first improvement or best improvement) and the shaking policy managing the shaking range (either constant or increasing). This generalized implementation of VNS covers a wide range of VNS variants described in literature, namely Reduced VNS (RVNS), Variable Neighbourhood Descent (VND), Generalized VNS (GVNS), Variable Neighbourhood Decomposition Search (VNDS) and Skewed VNS (SVNS). For detailed descriptions see Hansen (2009). By combining strategies and parameter, we get hundreds of VNS search schemes - deterministic variants that only employ local search as well as stochastic variants that manage to escape from local optima. Those variants basically differ in their balance between exploring and exploiting search space. Additionally, the system supports MOO, whereas multiple objectives are combined hierarchically or weighted or are equally combined in order to improve pareto fronts.

Six neighbourhood structures create subspaces of the entire search space by encapsulating a certain set of operations used to modify the schedule. The implemented neighbourhoods are defined as follows:

- Merge two batches: find two batches to merge to one of them (and new position).
- Split a batch: find a batch to split and insert the newly emerging batch at a new position.

- Swap two batches: find to batches and swap their positions.
- Move a batch: move a batch to another position.
- Swap two jobs: find two jobs out of different batches and swap them.
- Move a job: find a job and move it to another batch.

Since heuristic search procedures operate on given solutions, we need to provide an initial schedule as start solution for each problem instance. We use dispatching rules executed in a simulation system to generate initial schedules, which also provide reference objective measures for analysing improvements gained by optimization. We use First-In-First-Out (FIFO) as simple dispatching rule that also serves as reference when determining the improvements obtained. The dispatching interval is set to three minutes, which means that every three minutes the dispatching procedure is executed, which probably results in a new batch started. For the purpose of this paper, we prefer to use simple dispatching rules instead of more sophisticated dispatching heuristics (BATC, NACH) described in literature. This papers' primary objective is neither to evaluate search method performances, nor to evaluate exact optimization potentials for certain model specifics, but to study the impact of accuracy in lot arrival prediction on solution quality.

For the optimization method we chose the VND as appropriate search strategy to demonstrate the effects of prediction errors. The objective to optimize is total cycle time (TCT). The maximum deadline for improving a schedule framed by a single time window is set to one minute. However, the average computing time never exceeds five seconds computing time per time window on average. The VND can be considered as a deterministic (best improvement) local search strategy operating on a limited set of neighbourhoods specifically designed to solve the parallel batch machine problem. There are two reasons justifying our decision to apply a deterministic VNS variant, instead to choose a stochastic one, which has been proven to outperform deterministic approaches. On the one hand, the deterministic behaviour of VND reduces the total number of experimental runs, since there is no need to run multiple replications that guarantee a certain level of statistical reliability, in contrast to stochastic VNS derivates. On the other hand, the deterministic local search provides a better understanding of scheduling complexity with regard to the size of problem instances. The examination of measured computing times and/or number of search moves, combined with the analysis of improvements by optimization related to performance measures, makes clear whether local optima can be found for certain problems in compliance to given computational deadlines. Additionally, in order to minimize the burden of analysing averages and variances in experiments and increase understandability and reliability of experimental results at the same time, we waived the use of stochastic search. In turn, deterministic search avoids analysing variances resulting from stochastic effects, caused by multiple replications. We also like to point to the fact that potentials in optimization considerably depend on model characteristics. Based on extensive studies, we observed that there don’t exist non-negligible variances for simulation (optimization) results among a set of independent instances belonging to the same model type.

Combined with the Time Window Decomposition (TWD), the VNS opens up the possibility to create optimized schedules for even large problem instances. Applied as simulation procedure, TWD also makes it possible to study additional factors, e.g.

the effect of rescheduling frequency or length of look ahead horizons (cf. Ovacik & Uzsoy 1995). For the presented experiments we use TWD as decomposition and simulation technique, where the interval for each time window is set to 30 minutes. This means, every 30 minutes a new time window begins and a new scheduling problem is solved. The maximum look-ahead horizon is subject to the examined experiment parameters.

4 The impact of prediction errors

Following the three laws of forecasting (cf. Hopp 2001), any prediction comes with errors due to the fact that models used for prediction are always simplified abstractions. The lack of modelled system behaviour, considered as negligible model details for the sake of model simplicity, necessarily results in prediction errors.

Scheduling, simulation and sophisticated dispatching systems have one thing in common: all of them make use of wafer fabrication equipment models. Those equipment models build the backbone for simulation systems and provide predictions for job completion dates and job arrivals, which facilitate scheduling and dispatching decisions. The most widespread equipment model, which consists of a set of formulas embedded in a proper algorithm, is analytical. Analytical equipment models are considered to provide a balanced trade-off between accuracy in prediction and speed of computation, both equally important for scheduling and simulation.

Usually, studies discussing PBMSPs with dynamic arrivals take those arrivals as errorless in their models. Due to the fact that look-ahead information is widely considered to be the largest source of optimization potentials in PBMSPs and taking into account that any prediction is erroneous, the impact of prediction errors on scheduling benefits is of particular interest for practitioners.

In order to evaluate the effect of errors in job arrival predictions we extended the TWD scheme in a way that predicted arrivals may be subject to disturbances. Therefore, the implemented model contains two dates for any job arrival: the predicted arrival date and the disrupted date that is finally used for schedule evaluation. The arrival dates are exposed to a normal distribution of offset errors, which may add a positive or negative delay to the originally predicted arrival. Therefore, we implemented two schemes in order to examine the effect of prediction errors. Within the first scheme the mentioned offset is given as an integer value, which represents the prediction error in minutes. Here the prediction error is simply taken from randomly generated distribution of integers within the given range. The second one applies an error scheme considering the actual look-ahead horizon for any predicted arrival. Here we calculate the prediction error for a certain job as the product of the time of the predicted arrival and a factor taken from a normal distribution. This way we take the fact into account that the more far a predicted event lies in future, the worse is the prediction in terms of the accuracy. According to TWD, any new time window is considered as a new scheduling problem. The VNS-based search procedure improves the schedule by taking the predicted job arrivals without errors into consideration. Then the improved schedule is simulated again in order to incorporate the erroneous arrival dates. Figure 1 shows the entire computational results in one diagram.

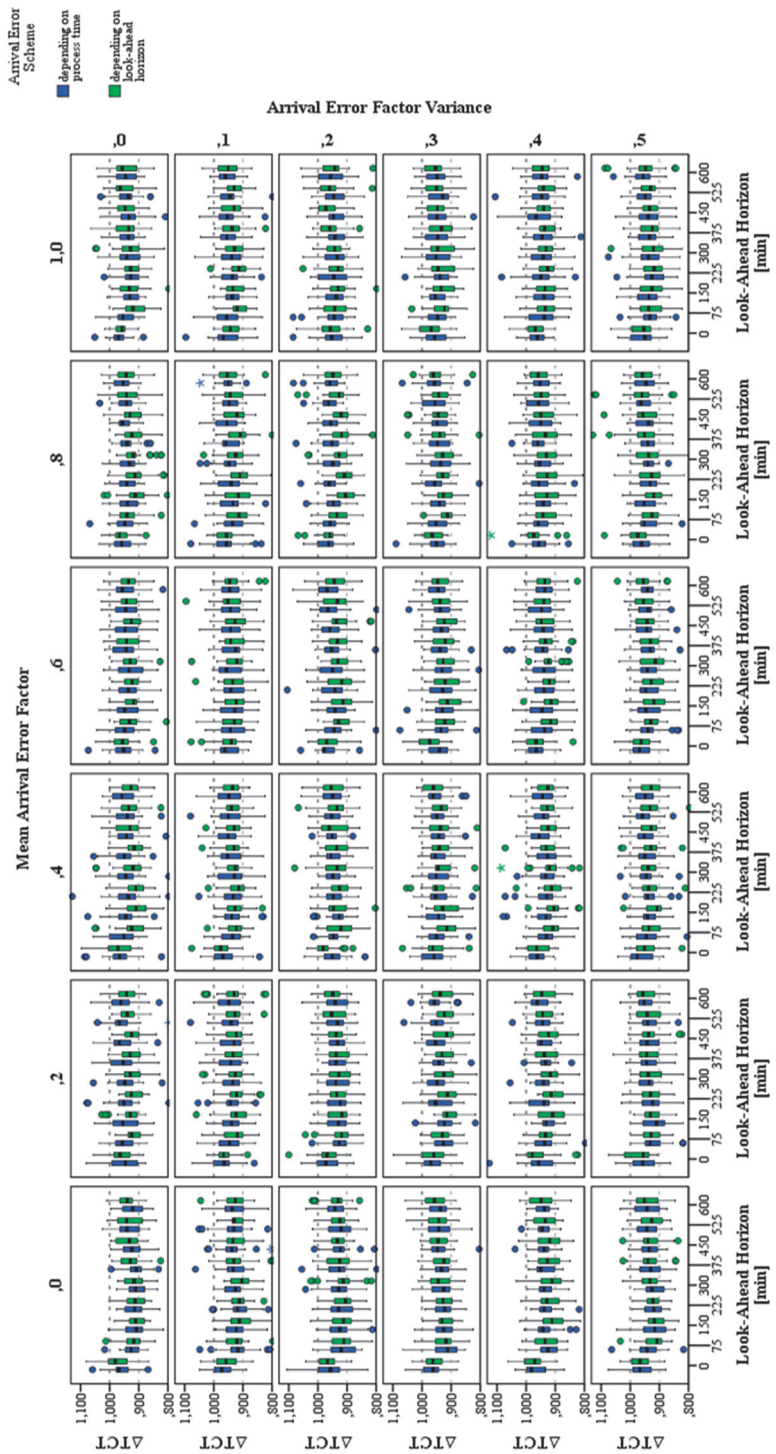

Figure 1: *Computational results*

4.1 Prediction errors depending on the average process time

This experiment provides results that are used to discuss the effect of prediction errors on optimization potential. We defined 36 model types that vary in their arrival error scheme, where six levels for the mean value and six levels for standard deviation value have been examined. The mean error varies between zero and the average processing time, respectively chosen from the array {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}. The value for the standard deviation lies between zero and one half of the average processing time, where the value belongs to the array {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}. For each model type we created 30 independent instances which result in 1080 problem instances in total. Additionally, the look-ahead horizon varies from zero to twice the average processing time, chosen from {0, 75, 150, 225, 300, 375, 450, 525, 600}. In total, the entire experiment took 9720 runs without those runs that provide the reference solutions.

When analysing the experimental results, we identified the following observations. We observed higher improvements for higher look-ahead horizons. Beginning with values for the look-ahead horizon greater than the average processing time, the observed benefit decreases slightly. As expected, higher mean error values generally result in lower improvements. Comparing the zero error case (best solution 0.9) with the case where the mean value equals the average processing time (best solution 0.95), we observed a loss of half of the benefit. The impact of the standard deviation in errors is less than expected: Compared to the zero standard deviation cases we observed two percent loss in benefit at maximum

Higher values for standard deviation partially lead to better solutions; in the high mean error cases, a higher spread of errors results in slightly better solutions, whereas in the low mean error cases a higher spread leads to worse solutions.

4.2 Prediction errors depending on the look-ahead horizon

In correspondence to the previous discussed experiment, we carried out 9720 simulation runs providing the results we used to discuss the effect of prediction errors on optimization potential. The difference is that we calculate the error in prediction as a function of the actual look-ahead horizon. We defined six levels for the mean value and six levels for standard deviation value, which results in 36 model types that vary in their arrival error scheme. The mean error varies between zero and the actual look-ahead horizon, respectively chosen from the array {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}. The value for the standard deviation lies in between zero and one half the actual look-ahead horizon, where the value belongs to the array {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}. Given that, we create 30 independent instances for each model type, we get 1080 problem instances in total. As in the previous experiment, the look-ahead horizon varies from zero to twice the average processing time, chosen from {0, 75, 150, 225, 300, 375, 450, 525, 600}.

5 Conclusions

Based on our experiments and in accordance with previous studies, we observed greater improvements for higher look-ahead horizons. As expected, higher mean error values generally result in lower improvements. Comparing the zero error case

(best solution 0.9) with the case where the mean value equals the average processing time (best solution 0.95), we observed a loss of half of the benefit.

The impact of the standard deviation in errors is less than expected: Compared to the zero standard deviation cases, we observed two percent loss in benefit at maximum. Even higher values for the standard deviation partially lead to better solutions; in the high mean error cases a higher spread of errors results in slightly better solutions, whereas in the low mean error cases a higher spread leads to worse solutions.

Acknowledgement

We like to thank our colleges from Infineon Technologies for their support at any time. Special thanks go to Jens Doleschal and Manfred Benesch from Dresden University of Sciences who maintain server and systems essential for our experiments with all their expertise and commitment. This work was supported by Grant 13N11588 of the Federal Ministry of Education and Research Germany.

References

Almeder, C.; Mönch, L.: Metaheuristics for scheduling jobs with incompatible families on parallel batching machines. JORS 62 (12), 2011, pp. 2083–2096.

Balasubramanian, H.; Mönch, L.; Fowler, J.W.; Pfund, M.E.: Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. International Journal of Production Research 42 (8), 2004, pp. 1621–1638.

Cakici, E.; Mason, S.J.; Fowler, J.W.; Geismar, H.N.: Batch scheduling on parallel machines with dynamic job arrivals and incompatible job families. International Journal of Production Research 51 (8), 2013, pp. 2462–2477.

Chiang, T.-C.; Cheng, H.-C.; Fu, L.-C.: A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival. Computers & Operations Research 37 (12) 2010, pp. 2257–2269.

Damodaran, P.; Vélez-Gallego, M.C.: A simulated annealing algorithm to minimize makespan of parallel batch processing machines with unequal job ready times. Expert Systems with Applications 39 (1), 2012, pp. 1451–1458.

Damodaran, P.; Vélez-Gallego, M.C.; Maya, J.: A GRASP approach for makespan minimization on parallel batch processing machines. Journal of Intelligent Manufacturing 22 (5), 2011, pp. 767–777.

Fowler, J.; Phillips, D.; Hogg, G.: Real-time control of multiproduct bulk-service semiconductor manufacturing processes. Semiconductor Manufacturing, IEEE Transactions on 5 (2), 1992, pp. 158–163.

Fowler, J.W.; Hogg, G.L.; PHILLIPS, D.T.: Control of multiproduct bulk server diffusion/oxidation processes. Part 2: multiple servers. IIE Transactions 32 (2), 2000, pp. 167–176.

Glassey, C.; Weng, W.: Dynamic batching heuristic for simultaneous processing. Semiconductor Manufacturing, IEEE Transactions on 4 (2), 1991, pp. 77–82.

Graham, R.; Lawler, E.; Lenstra, J.; Rinnooy Kan, A.: Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey, in: Hammer, P.L., Johnson, E., Korte, B. (Eds.), Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and

Systems Applications, Aha. and Vancouver. Elsevier Science, 1979, pp. 287–326.

Gupta, A.K.; Sivakumar, A.I.: Optimization of due-date objectives in scheduling semiconductor batch manufacturing. International Journal of Machine Tools and Manufacture 46 (12–13), 2006, 1671–1679.

Habenicht, I.; Mönch, L.: Simulation-based assessment of batching heuristics in semiconductor manufacturing, in: Chick, S.E., Sanchez, P.J., Ferrin, D.M., Morrice, D.J. (Eds.), Proceedings of the 35th Winter Simulation Conference. ACM, 2003, pp. 1338–1345.

Hansen, P.; Mladenovi´c, N.M.; Brimberg, J.; P´erez, J.A.M.: Variable Neighborhood Search, in: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of metaheuristics, 2nd. New York: Springer 2009.

Hopp, W.J.; Spearman, M.L.: Factory physics. Foundations of manufacturing management, 2 ed. Boston: Irwin/McGraw-Hill 2001.

Jula, P.; Leachman, R.C.: Coordinated Multistage Scheduling of Parallel Batch-Processing Machines Under Multiresource Constraints. Operations Research 58 (4-Part-1), 2010, pp. 933–947.

Kashan, A.H.; Karimi, B.; Jenabi, M.: A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. Computers & Operations Research 35 (4), 2008, pp. 1084–1098.

Kim, Y.-D.; Joo, B.-J.; Choi, S.-Y.: Scheduling Wafer Lots on Diffusion Machines in a Semiconductor Wafer Fabrication Facility. Semiconductor Manufacturing, IEEE Transactions on 23 (2), 2010, pp. 246–254.

Klemmt, A.; Weigert, G.; Almeder, C.; Mönch, L.: A comparison of MIP-based decomposition techniques and VNS approaches for batch scheduling problems, in: Rossetti, M.D., Hill, R.R., Johansson, B., Dunkin, A., Ingalls, R.G. (Eds.), Proceedings of the Winter Simulation Conference, 2009, pp. 1686–1694.

Klemmt, A.; Weigert, G.; Werner, S.: Optimisation approaches for batch scheduling in semiconductor manufacturing. European Journal of Industrial Engineering 5 (3), 2011, pp. 338–359.

Li, L.; Qiao, F.; Wu, Q.: ACO-Based Scheduling of Parallel Batch Processing Machines with Incompatible Job Families to Minimize Total Weighted Tardiness, in: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A. (Eds.), Ant Colony Optimization and Swarm Intelligence. Springer 2008, pp. 219–226.

Malve, S.; Uzsoy, R.: A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families. Computers & Operations Research 34 (10), 2007, pp. 3016–3028.

Mathirajan, M.; Sivakumar, A.I.: A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. The International Journal of Advanced Manufacturing Technology 29 (9), 2006, pp. 990–1001. http://dx.doi.org/10.1007/s00170-005-2585-1.

Mladenović, N.; Hansen, P.: Variable neighborhood search. Computers & Operations Research 24 (11), 1997, pp. 1097-1100.

Mönch, L.; Almeder, C.: Ant Colony Optimization for Scheduling Jobs with Incompatible Families on Parallel Batch Machines, in: Proceedings of the 4th Multidisciplinary International Scheduling Conference (MISTA), 2009, pp. 106–114.

Mönch, L.; Balasubramanian, H.; Fowler, J.W.; Pfund, M.E.: Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. Computers & Operations Research 32 (11), 2005, pp. 2731–2750.

Mönch, L.; Fowler, J.W.; Mason, S.J.: Production planning and control for semiconductor wafer fabrication facilities. Modeling, analysis, and systems. New York: Springer 2013.

Ovacik, I.M.; Uzsoy, R.: Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. International Journal of Production Research 33 (11), 1995, pp. 3173–3192.

Raghavan, N.S.; Venkataramana, M.: Scheduling Parallel Batch Processors with Incompatible Job Families Using Ant Colony Optimization, in: Proceedings of the International Conference on Automation Science and Engineering, Shanghai 2006, pp. 507–512.

Reichelt, D.; Mönch, L.: Multiobjective Scheduling of Jobs with Incompatible Families on Parallel Batch Machines, in: Gottlieb, J., Raidl, G.R. (Eds.), Evolutionary Computation in Combinatorial Optimization. Berlin, Heidelberg: Springer 2006, pp. 209–221.

Sha, D.; Hsu, S.-Y.; Lai, X.: Design of due-date oriented look-ahead batching rule in wafer fabrication. The International Journal of Advanced Manufacturing Technology 35 (5-6), 2007, pp. 596–609.

SOLOMON, L.; Fowler, J.W.; Pfund, M.; JENSEN, P.H.: The inclusion of future arrivals and downstream setups into wafer fabrication bath processing decisions. Journal of Electronics Manufacturing 11 (02), 2002, pp. 149–159.

van der Zee, D.J.: Dynamic scheduling of batch-processing machines with non-identical product sizes. International Journal of Production Research 45 (10), 2007, pp. 2327–2349.

Wang, H.-M.; Chou, F.-D.: Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics. Expert Systems with Applications 37 (2), 2010, pp. 1510–1521.

Yugma, C.; Dauzere-Peres, S.; Derreumaux, A.; Sibille, O.: A Batch Optimization Solver for diffusion area scheduling in semiconductor manufacturing, in ASMC 2008. IEEE/SEMI 2008, pp. 327–332.