Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

SIMULATION-BASED ASSESSMENT OF ORDER RELEASE STRATEGIES FOR A DISTRIBUTED SHIFTING BOTTLENECK HEURISTIC

Lars Mönch

Institute of Information Systems
Technical University of Ilmenau
Helmholtzplatz 3
98684 Ilmenau, GERMANY

ABSTRACT

In this paper, we investigate the influence of several order release strategies on the performance of a distributed shifting bottleneck heuristic. The shifting bottleneck heuristic is a decomposition approach that solves the overall scheduling problem by solving a sequence of tool group scheduling problems and determines the overall solution by using a disjunctive graph. We discuss a distributed version of the original shifting bottleneck heuristic. By using a hierarchical approach we first assign planned ready and completion dates to all lots with respect to a certain work area where a work area is defined as a set of tool groups. We study several order release strategies. It turns out that the distributed shifting bottleneck heuristic performs well compared to dispatching rules only in high loaded job shops. We present the results of computational experiments.

1 INTRODUCTION

The electronics industry is one of the largest industries in the world. Semiconductor manufacturing is at the heart of this industry. The wafer fabrication part of semiconductor manufacturing is rather complex, consisting of hundreds of process steps, diversity of product mix, reentrant flows, sequence dependent setups and batch processing. Currently, it seems that the improvement of operational processes creates the best opportunity to realize the necessary cost reductions. Therefore, the development of efficient planning and control strategies is very beneficial in the semiconductor manufacturing domain.

Semiconductor wafer fabrication facilities (wafer fabs) are examples for complex job shops. Complex job shops are defined as flexible job shops that are characterized additionally by the process conditions of semiconductor wafer fabrication (Ovacik and Uzsoy 1997, Mason, Fowler, and Carlyle 2002). In this paper, we investigate a special class of complex job shops which we call decomposable complex job shops. Decomposable job shops are characterized by a proper physical decomposition of the shop into work areas. Each work area consists of groups of parallel machines. The photolithography area, the diffusion, and the etching area are examples for work areas in a wafer fab.

Scheduling problems can be represented in the form $\alpha \mid \beta \mid \gamma$ (Graham, Lawler, Lenstra, and Rinnooy Kan 1979). The α field describes the machine environment (single machine, parallel machine, job shop, etc.), the β field describes the process characteristics, restrictions, constraints (such as release dates, batch, set-up dependent operations), and the γ field contains the information on which performance measure being considered. For the problem being researched in this paper the notation is

$$FJ_m \mid batch, incompatible, r_j, s_{jk}, recrc \mid TWT, \quad (1)$$

where we denote a decomposable flexible job shop by FJ_m. We denote batching tools with incompatible families by *batch* and *incompatible*. Batching tools allow for the processing of several lots at the same time on the same tool. We call the set of these lots a batch. Only lots belonging to the same lot family can be batched together. Each lot has a weight w_j, a due date d_j and a release date/ready time r_j. We indicate sequence dependent set-up times by s_{jk} and reentrant flows by *recrc*. Our objective is to minimize the total weighted tardiness $TWT = \sum w_j T_j$ of the lots. The scheduling problem of interest is more complex than the problem $1||\Sigma w_j T_j$ for single machines which is known to be NP-hard. Therefore, we propose a heuristic scheduling approach to solve it. We suggest a distributed version of the shifting bottleneck heuristic. So far, only little is known on the performance of this heuristic in different order release and load situations.

The paper is organized as follows. We describe the distributed shifting bottleneck heuristic (DSBH) that is investigated in this paper in the next section. We discuss several order release strategies in the third section. In the final section, we present and discuss the results of computational experiments. Finally, we present conclusions and identify some future research topics.

2 DISTRIBUTED SHIFTING BOTTLENECK HEURISTIC

In this paper, we apply a two-layer approach that considers explicitly tardiness related performance measures. The top layer works on an aggregated model. Therefore, we form macro operations. Each macro operation consists of a set of consecutive process steps. A single macro operation is related to a specific work area. We assign start dates and end dates to each single macro operation based on an infinite capacity approach (ICA). The main idea of ICA consists in considering the ratio

$$h_j := \frac{d_j - t}{\sum p_{jl}}, \tag{2}$$

where we denote

h_j : dynamic flow factor for lot j,
t : current time,
p_{jl} : sum of the processing time of the process steps of macro operation l for lot j.

The quantity h_j basically distributes the remaining time until the due date equally among the remaining macro operations of lot j. It is only well defined if $d_j - t \geq \sum p_{jl}$ holds. In this case, we determine start dates r_{jl} for macro operation l of lot j in a recursive manner via

$$r_{jl} := r_{jl-1} + p_{jl-1} + \{h_j - 1\} p_{jl-1} \tag{3}$$

and $r_{j1} := r_j$ as initial condition. The planned end date of macro operation $jl-1$ is given by r_{jl}. If $d_j - t < \sum p_{jl}$ is valid then the lot will be late and we have to assign a new due date to the lot. Then, we can again apply the ICA procedure in order to determine start dates and end dates. We apply ICA in a rolling horizon manner every δ_Δ time units.

As a result of ICA we obtain internal ready times $r_j^{(k)}$ and internal due dates $d_j^{(k)}$ for all lots j with respect to work area k. ICA is a simple heuristic. It does not take into account any capacity constraints. Furthermore, it is not a good idea to distribute the remaining time equally among the remaining macro operations. However, for the purpose of our hierarchical approach ICA is good enough.

The shifting bottleneck heuristic is a prominent decomposition approach for job shops. It decomposes the overall scheduling problem into scheduling problems for single tool groups. A scheduling graph connects the results of the scheduling problems for the single tool groups and provides a view on the overall problem. The main steps of the shifting bottleneck heuristic can be described as follows (Ovacik and Uzsoy 1997, Mason, Fowler, and Carlyle 2002):

1. Denote the set of all tool groups by M. We use the notation M_0 for the set of tool groups that have already been sequenced or scheduled. Initially, set $M_0 := \emptyset$.
2. Identify and solve the subproblems for each tool group $i \in M - M_0$.
3. Identify a critical tool group $k \in M - M_0$.
4. Sequence the critical tool group using the subproblem solution obtained by Step 2 by incorporating the related conjunctive arcs into the scheduling graph. Set $M_0 := M_0 \cup \{k\}$ for update purposes.
5. (Optionally) re-optimize the schedule for each tool group $m \in M_0 - k$ by exploiting the information provided by the newly added disjunctive arcs for tool group k.
6. If $M = M_0$, terminate the heuristic. Otherwise, go to Step 2.

Mason, Fowler, and Carlyle (2002) discuss modifications of the shifting bottleneck heuristic for complex job shops. Batch processing tools and reentrant flows are modeled by adding additional arcs to the disjunctive graph.

The usage of the shifting bottleneck heuristic for complex job shops has a number of drawbacks. The size of the scheduling graph increases tremendously with a larger scheduling horizon. As a result, the runtime performance is poor and the software application needs a large amount of memory. Therefore, it makes sense to distribute the shifting bottleneck. In case of decomposable complex job shops we can use the shifting bottleneck heuristic for each single work area. The shifting bottleneck heuristic uses the start dates $r_j^{(k)}$ and end dates $d_j^{(k)}$ of lot j with respect to work area k given by ICA in order to determine detailed schedules for the lots that have to be processed on the tool groups of work area k within a certain scheduling horizon. Note that because of the re-entrant flows multiple ready times and due dates appear with respect to a certain work area. In order to solve this problem, we add artificial arcs between nodes that represent process steps of the same lot belonging to consecutive macro operations.

The hierarchical approach can be summarized as follows:

1. Determine start dates and end dates of the lots with respect to the fixed work area k in a rolling horizon manner every δ_Δ time units.
2. Determine schedules for each single work area by using the modified shifting bottleneck heuristic of Mason, Fowler, and Carlyle (2002) and the start dates and end dates from Step 1 for a scheduling horizon of length $\tau_\Delta + \tau_{ah}$ in a rolling horizon manner. Here, we denote by τ_Δ the scheduling interval and by τ_{ah} the additional scheduling horizon.

We show the overall scheduling approach in Figure 1. We use an order pool to stop arrived lots until they will be released into the shop floor.

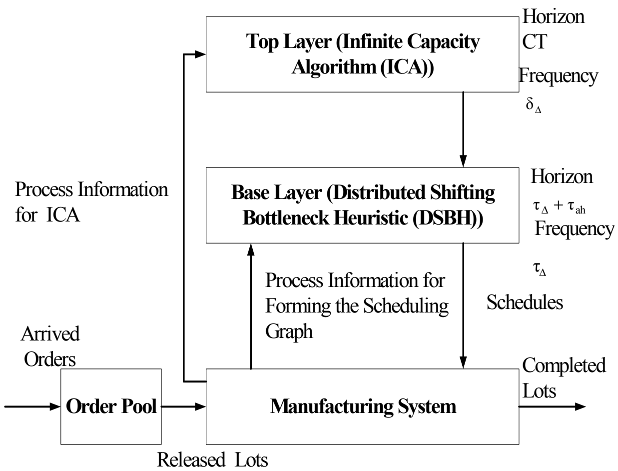

Figure 1: Hierarchical Scheduling Approach

Note that the shifting bottleneck heuristic has to be used separately for each work area because of the decoupling effect of the top layer. DSBH can be improved by using correction intervention techniques, i.e., starting from a schedule for a fixed work area, we send start dates and (planned) completion dates obtained by the schedule to the remaining work areas and make so the start date and completion time estimates from the top layer obsolete. For a more detailed description of the novel distributed shifting bottleneck heuristic including more sophisticated variants we refer to (Mönch and Driessel 2005).

So far, we consider only a moderate and a very high load scenario in our simulation experiments. The lots are released by a simple push strategy. A more systematic investigation of different lot release strategies is highly desirable. Some initial steps towards this goal are described in this paper.

3 ORDER RELEASE STRATEGIES

We start with discussing related work. Then, we describe the order release strategies used in this research.

3.1 Literature Review

Order release strategies are a subject of intensive discussion in the literature. The discussion is to a certain extend controversial and the results are not always consistent. Usually, it is distinguished between order review and order release. An order review decision consists in determining whether an order should be accepted or rejected. Once an order is accepted for processing then an order release decision consists in determining a point of time for launching the lots obtained from the order.

Four different order release schemes are compared in (Roderick, Phillips, and Hogg 1992). Philipoom and Fry (1992) discuss different order review and order release schemes. We refer to the survey papers written by Bergamaschi, Cigolini, Perona, and Portioli (1997) for a detailed discussion of order review and order release techniques in manufacturing. A more recent work dealing with order review and order release schemes for a make-to-order manufacturing system is presented by Nandi and Rogers (2003).

A recent survey on order release strategies and related work load control for the semiconductor industry is presented by Fowler, Hogg, and Mason (2002). Most of the papers deal with order release issues in the context of dispatching. In the semiconductor domain, we refer to the more recent papers of Rose for CONLOAD (Rose 1999) and CONWIP type lot release strategies (Rose 2001).

Order release schemes and scheduling are usually treated independently. There is only little known on the interaction of order release schemes and sophisticated scheduling approaches like the shifting bottleneck heuristic. The interaction of a scheduling approach and a lot release scheme is discussed in a sequence dependent set-up situation by Ashby and Uzsoy (1995).

3.2 Order Release Strategies Used in this Research

We investigate the performance of a simple PUSH strategy, a CONWIP type order release strategy, and a CONLOAD type strategy.

The PUSH strategy releases lots as required by the customer due dates. Only simple capacity considerations are taken into account during lot release. The release time is calculated by a simple backward calculation based on flow factors, i.e., we set the release date r_j von lot j as

$$r_j := d_j - FF \sum_{i=1}^{n} p_{ji}$$

Here, FF is a flow factor that is chosen by capacity considerations. We release lot j at time r_j.

The usage of the CONWIP strategy requires shop characteristic curves that provide the relationship between work in process (WIP) and production rate (output of lots per day). A WIP level consistent with the requested output is determined. When lots are completed, new lots are released to reach the target WIP level.

The CONLOAD strategy exploits also a characteristic curve that provides the relationship between work load and production rate. The workload of the manufacturing system is measured at equidistant points of time. Here, we simply measure the workload as the sum of the processing times of the remaining processing steps of all released lots, i.e.,

$$WL := \frac{\sum_{j=1}^{n} \sum_{i=k+1}^{n_j} p_{ji}}{nCT}, \tag{4}$$

where we denote by jk the last performed process step of lot j and the notation CT is used for the target cycle time. We denote the number of already released and not completed lots by n. Note that $WL \in [0,1]$ is valid. WL provides a more accurate picture of the load situation of a certain job shop because it takes into account how many process steps the jobs already have completed.

4 PERFORMANCE ASSESSMENT

In this section, we describe the used simulation framework including a description of simulation models. Then, in the second part, we present our research methodology. In the third part, we show results of computational experiments.

4.1 Simulation Framework for Experimentation

We use the simulation framework that is suggested by Mönch, Rose, and Sturm (2003). The center point of the framework is a data layer that is between the simulation model and the distributed shifting bottleneck heuristic.

We use two simulation models in our experiments. The first one is a small complexity model, called MiniFab model, suggested by researchers from INTEL Corporation and described by El Adl, Rodriguez and Tsakalis (1996). In its original form it contains only three tool groups and two product routes with seven steps. The process flow is organized into two layers. Among the tool groups, there is a batch processing one and a tool group with sequence-dependent set-up times. The model mimics some important features of wafer fabs. We derive a new model that contains three work areas. Each of them contains the machinery of the MiniFab model. The process flows are organized into two layers. We denote this model by Model A.

The second model is a reduced variant of the MIMAC Testbed Data Set 1 (Fowler and Robinson 1995). It contains two routes with 100 and 103 steps respectively. The process flow is highly reentrant. The lots are processed on 146 machines that are organized into 37 tool groups. Among the tools are batching tools. The model contains four work areas. We denote the second model by Model B.

We take 180 days of simulation run time after an appropriate warm-up time. We do not consider any machine break downs.

We use the parameters $\delta_\Delta = 2h$, $\tau_\Delta = 2h$, and $\tau_{ah} = 0h$ for the top and base layer of our hierarchical approach respectively. Note that we know from (Mönch and Driessel 2005) that this setting leads to a good performance of DSBH in many situations.

4.2 Methodology

We measure the average weighted tardiness (AWT), the average cycle time (ACT), and the throughput (TP) for each single scenario. These performance measures are defined as follows. We define the average weighted tardiness of the lots as

$$AWT := \frac{1}{n} \sum_{j=1}^{n} w_j \, max(c_j - d_j, 0), \tag{5}$$

where we denote the completion time of lot j by c_j and by n the number of completed jobs. The second performance measure is the average cycle time. It is defined as

$$ACT := \frac{1}{n} \sum_{i=1}^{n} (c_j - r_j), \tag{6}$$

where we denote by r_j the ready time of lot j. The third performance measure of interest is the throughput, i.e.,

$$TP := \#\{j \mid c_j < T\}, \tag{7}$$

where we denote by T the simulation horizon. It is the number of completed lots within a certain time period. The fourth measure of interest is the average work in process (WIP) measured in lots. We compare the performance of the distributed shifting bottleneck against a pure First In First Out (FIFO) dispatching scheme. We consider two different weight distributions for the lots. The distribution D_1 is defined as follows:

$$D_1 := \begin{cases} w_j = 1 & with \quad p_1 = 0.50, \\ w_j = 5 & with \quad p_2 = 0.35, \\ w_j = 10 & with \quad p_3 = 0.15. \end{cases} \tag{8}$$

Distribution D_1 mimics the situation that a small number of lots have a high weight and a large number of lots have a medium weight.

The distribution D_2 is given by

$$D_2 := \begin{cases} w_j = 1 & \text{with} \quad p_1 = 0.50, \\ w_j = 2 & \text{with} \quad p_2 = 0.45, \\ w_j = 10 & \text{with} \quad p_3 = 0.05. \end{cases} \quad (9)$$

The second distribution is used to model manufacturing systems where a very small portion of the lots have a high priority and the remaining lots have a small weight.

We use different flow factors in order to set the due dates according to equation (3).

The used experimental design for Model A is summarized in Table 1. We consider a similar design for Model B, however, we use different flow factors *FF* for the due date setting.

Table 1: Factorial Design for the Experiments

Factor	Level	Count
Lot Release Scheme	PUSH; CONWIP; CONLOAD	3
Load of the Wafer Fab	Low Moderate; High; Very High	4
Due Dates	*FF=1.3; FF= 1.5; FF=1.7*	3
Weight Setting	$D_1; D_2$	2

We present the relationship between WIP and output for Model A in Figure 2. It turns out that DSBH leads to a higher WIP level for a fixed output.

Based on the relationship provided by Figure 2, we choose certain output levels. For Model A, we use $\lambda_1 = 14$ lots per day as output rate obtained by a WIP of 80 lots (high load).

A WIP of 60 lots leads to an output rate of 13.3 lots per day (moderate load). In the low load case, a WIP of 40 lots leads to an output rate of 11.5 lots per day.

We also use these output rates to define different load levels that have to be used within the experiments. We also define the lot release rates for the PUSH strategy based on the fixed output rates. Additionally, we obtain a very high loaded system by increasing the lot release rate that leads to a high loaded system. In this case, we simulate only 90 days after an appropriate warm-up time.

In all experiments, we first use the FIFO dispatching scheme and the order release scheme of interest to obtain reference values for the performance measures. We present all results as ratio of the performance measure value obtained by DSBH and the corresponding performance measure value obtained by FIFO.

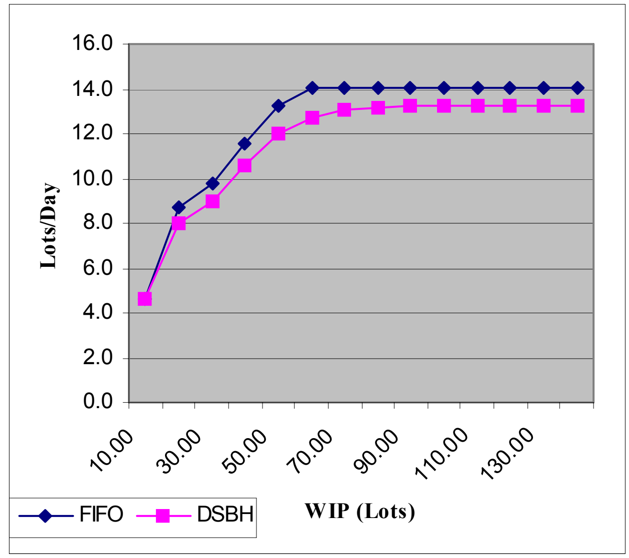

Figure 2: Characteristic Curve for Model A

4.3 Results of Computational Experiments

We present the results of computational experiments in this section. First, we consider the PUSH, the CONWIP, and the CONLOAD strategy separately by using the different load situations described in the last subsection. Then we compare these three strategies.

4.3.1 Results for the PUSH Strategy

We present the computational results for the PUSH strategy in Table 2. It turns out that DSBH outperforms FIFO only in the very high loaded system case. In the case of a high loaded system, the FIFO dispatched system is stable, while the system under a DSBH type scheduling regime produces an increasing WIP. Large cycle times are the result of this behaviour.

We obtain the highest improvement rates in the case of tight due dates (*FF=1.3*) and moderate due dates (*FF=1.5*). We found no significant difference between the two weighting schemes.

Based on the experiments it turns out that the usage of DSBH under a PUSH regime makes only sense in very congested job shops. In the remaining cases, the FIFO dispatching rule is a better choice. This results was partially already obtained by (Mönch and Driessel 2005).

Table 2: Results for the PUSH Strategy (Model A)

DSBH/FIFO	WIP	TP	ACT	AWT
Low Load				
D_1				
FF=1.3	1.1622	0.9990	1.1534	1.9587
FF=1.5	0.8222	0.9966	1.1588	1.0000
FF=1.7	0.8409	0.9966	1.1609	1.0000
D_2				
FF=1.3	1.1622	0.9980	1.1152	1.9013
FF=1.5	0.9250	0.9980	1.1050	1.0000
FF=1.7	0.9487	0.9990	1.1228	1.0000
Moderate Load				
D_1				
FF=1.3	1.0577	0.9412	1.1270	0.9833
FF=1.5	1.0578	0.9367	1.1270	1.2332
FF=1.7	1.0575	0.9367	1.1366	28.9552
D_2				
FF=1.3	1.0769	0.9553	1.1056	1.1213
FF=1.5	1.0577	0.9528	1.1141	1.5557
FF=1.7	1.0385	0.9503	1.1216	36.5385
High Load				
D_1				
FF=1.3	1.5972	0.9802	1.5572	0.9789
FF=1.5	1.8056	0.9726	1.6100	1.1149
FF=1.7	2.0833	0.9607	1.8222	1.8238
D_2				
FF=1.3	1.5000	0.9794	1.5123	0.9969
FF=1.5	1.4375	0.9830	1.4654	1.6072
FF=1.7	1.6250	0.9782	1.5181	2.3544
Very High Load				
D_1				
FF=1.3	1.1429	0.9767	1.0609	0.3941
FF=1.5	0.9857	0.9758	1.0632	0.3345
FF=1.7	1.0714	0.9642	1.1368	0.3519
D_2				
FF=1.3	0.9333	0.9792	1.0402	0.5837
FF=1.5	0.9333	0.9842	1.0632	0.5841
FF=1.7	0.9333	0.9792	1.1207	0.6356

We present computational results for Model B for a PUSH strategy in Table 3. Based on the results for Model A, we decided to consider only the case of a high load and a very high load. Furthermore, we conduct only experiments for the weight setting D_1.

The results in Table 3 indicate the same behavior for Model B as for Model A in the high load and the very high load case. Note that Model B has compared to Model A an reasonable size. Again, the used lot release rate leads to a stable system for FIFO in the case of a high load whereas we have to face with an increasing WIP in the DSBH case.

Table 3: Results for the PUSH Strategy (Model B)

DSBH/FIFO	WIP	TP	ACT	AWT
High Load				
FF=1.4	1.0385	0.9948	1.0589	1.0533
FF=1.8	1.0538	0.9916	1.0746	4.4303
Very High Load				
FF=1.4	1.1120	1.0111	0.9933	0.6023
FF=1.8	1.1177	1.0059	1.0152	0.4011

4.3.2 Results for the CONWIP Strategy

We present the results of computational experiments for Model A in Table 4. We show only results for the high and the very high load case. The results for the other load situations are basically the same as for the PUSH strategy, i.e., DSBH is outperformed by FIFO. We use a WIP level of 80 lots. We perform only experiments for the weight setting D_1 because we do not find a significant difference between D_1 and D_2.

Table 4: Results for the CONWIP Strategy (Model A)

DSBH/FIFO	WIP	TP	ACT	AWT
High Load				
D_1				
FF=1.3	1.0556	0.9667	1.1234	0.6469
FF=1.5	1.0556	0.9635	1.1421	0.5932
FF=1.7	1.0833	0.9504	1.1499	0.6174
D_2				
FF=1.3	1.0405	0.9754	1.1070	0.8341
FF=1.5	1.0135	0.9762	1.1061	1.4148
FF=1.7	1.0270	0.9719	1.1153	0.9657
Very High Load				
D_1				
FF=1.3	0.9897	0.9771	1.0167	0.4395
FF=1.5	0.9794	0.9615	1.0248	0.3684
FF=1.7	0.9691	0.9714	1.0248	0.3214
D_2				
FF=1.3	0.9897	0.9894	1.0058	0.6042
FF=1.5	0.9794	0.9845	1.0181	0.5772
FF=1.7	0.9794	0.9795	1.0100	0.5404

It turns out that CONWIP leads for a high loaded system in most situations to a AWT reduction of 30 percent or more. This behavior is different to the PUSH case. We obtain that in case of a CONWIP type strategy it is useful to use DSBH. We show computational results for a CONWIP strategy applied to Model B and a WIP level of 130 lots in Table 5. Again, we consider only the case of a high and a very high loaded system. We use only the D_1 weight setting scheme.

Table 5: Results for the CONWIP Strategy (Model B)

DSBH/FIFO	WIP	TP	ACT	AWT
High Load				
FF=1.4	0.9032	0.9851	1.0358	0.9744
FF=1.8	0.9919	0.9770	1.0345	1.0329
Very High Load				
FF=1.4	1.0072	0.9807	1.0158	0.8222
FF=1.8	1.0000	0.9877	1.0134	0.5415

We obtain from the results in Table 5 that the behavior of DSBH under a CONWIP strategy is basically the same as for Model A. Again, it makes sense to apply DSBH in high load and very high load situations.

4.3.3 Results for the CONLOAD Strategy

We present the results for this order release strategy only for the case of high and very high loaded systems by the same arguments as in the CONWIP case. We use *WL=0.76* for the high loaded case and *WL=0.78* for the very high loaded case. Here, we simply set the target cycle time as the raw processing time.

Table 6: Results for the CONLOAD Strategy (Model A)

DSBH/FIFO	WIP	TP	ACT	AWT
High Load				
D_1				
FF=1.3	1.0667	0.9614	1.1772	0.7033
FF=1.5	0.9333	0.9506	1.1003	0.5713
FF=1.7	0.8667	0.9414	1.0354	0.4323
D_2				
FF=1.3	1.1351	0.9774	1.2067	1.0177
FF=1.5	1.0541	0.9652	1.1094	0.8744
FF=1.7	1.0676	0.9551	1.0384	0.7061
Very High Load				
D_1				
FF=1.3	1.1176	0.9665	1.1878	0.6154
FF=1.5	1.1412	0.9607	1.1160	0.4782
FF=1.7	0.8500	0.9515	1.1199	0.4594
D_2				
FF=1.3	1.1176	0.9732	1.1773	0.8508
FF=1.5	1.0235	0.9757	1.0795	0.7010
FF=1.7	1.0000	0.9682	1.0398	0.6192

We obtain from Table 6 that DSBH is able to outperform the FIFO strategy in almost all situations. Clearly, it is useful to combine DSBH with a CONLOAD type strategy in the case of a manufacturing system with a high load.

4.3.4 Comparison of the Results

Based on the simulation results, it turns out that we do not obtain any significant difference between the three order release strategies in the low and in the moderate loaded case. In this situation, usually DSBH is outperformed by FIFO. This behavior is obtained for Model A and Model B. The average cycle time ACT is increased whereas the throughput is decreased by using DSBH. This behavior is caused by wide internal due dates and the resulting low quality scheduling decisions on the subproblem level of DSBH.

The situation is more interesting in the case of high and very high loaded systems. For high loaded systems, the PUSH strategy is not applicable in combination with DSBH. However, in case of a CONWIP or CONLOAD type lot release strategy DSBH is useful. CONLOAD outperforms CONWIP only in a few situations for Model A with respect to AWT whereas CONWIP is always outperformed by CONLOAD in very high load situations. For very high loaded systems, even the PUSH strategy might be applied in combination with DSBH and leads to AWT reductions. In this situation, CONLOAD leads to slightly better AWT results compared to CONWIP. CONWIP itself slightly outperforms the PUSH strategy. Note that the performance measures ACT and TP change only slightly in the high load case.

5 CONCLUSIONS AND FUTURE RESEARCH

In this paper, we discuss several lot release schemes applied to a distributed shifting bottleneck approach. It turns out that the distributed shifting bottleneck heuristic outperforms a FIFO type dispatching scheme with respect to total weighted tardiness only for very high loaded systems under a PUSH order release scheme. In situations where we cap the WIP by using CONWIP and CONLOAD type order release schemes usually the distributed shifting bottleneck heuristic performs similar like the FIFO dispatching rule. Hence in this situation we do not need to apply the distributed shifting bottleneck heuristic. The situation changes for high loaded systems. In this situation, DSBH outperforms FIFO for CONWIP and CONLOAD type lot release schemes. So far, we considered only the case of continuous lot arrivals, i.e, we try to release newly arrived lots approximately every two, three or four hours. However, according to the current industrial practice it makes sense to investigate lot release schemes with daily or weekly frequency. In this situation, we expect a reduced due date performance in case of CONWIP or CONLOAD type lot release schemes because the time that the lots spend for waiting on the shop floor is shifted to waiting time in the order pool. However, determining the concrete value of due date performance losses is part of future research. Furthermore, as discussed for example by Philipoom and Fry (1992), Bergamaschi, Cigolini, Perona, and Portioli (1997), Nandi and Rogers (2003) it makes sense to consider in addition to order release strategies also order review techniques, i.e., the possibility to reject specific lots based on capacity and due date considerations.

In future research it seems to be highly desirable to investigate the connection between lot release decisions and anticipated scheduling decisions of the shifting bottleneck heuristic. We allow for releasing new lots into the manufacturing system based on the anticipated load situation of certain bottleneck tools due to newly released lots and WIP lots. This approach exploits directly the fact that we deal with scheduling rather than with dispatching. Furthermore, multi-product scenarios should be studied.

REFERENCES

Ashby, J. R. and R. Uzsoy. 1995. Scheduling and Order Release in a Single Stage Production System. *Journal of Manufacturing Systems*, 14(4), 290-305.

Bergamaschi, D., R. Cigolini, M. Perona, and A. Portioli. 1997. Order Review and Release Strategies in a Job Shop Enviroinment: a Review and a Classification. *International Journal of Production Research*, 35(2), 339-420.

El Adl, M. K., A. A. Rodriguez, and K. S. Tsakalis. 1996. Hierarchical Modelling and Control of Re-entrant Semiconductor Manufacturing Facilities. *Proceedings of the 35th Conference on Decision and Control*. Kobe, Japan.

Fowler, J. W. and J. Robinson. 1995. *Measurement and Improvement of Manufacturing Capacities (MIMAC): Final Report.* Technical Report 95062861A-TR, SEMATECH, Austin, TX.

Fowler, J. W., G. L. Hogg, and S. J. Mason. 2002. Workload Control in the Semiconductor Industry. *Production Planning. & Control,* 13(7), 568-578

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kann. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287 – 326.

Mason, S. J., J. W. Fowler, and W. M. Carlyle. 2002. A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops. *Journal of Scheduling*, 5(3), 247-262.

Mönch, L. and R. Driessel. 2005. A Distributed Shifting Bottleneck Heuristic for Complex Job Shops. To appear in *Computers & Industrial Engineering.*

Mönch, L., O. Rose, and R. Sturm. 2003. A Simulation Framework for Performance Assessment of Shop-Floor Control Systems. *SIMULATION: Transactions of the Society of Modeling and Computer Simulation International*, 79(3), 163-170.

Nandi, A. and P. Rogers. 2003. Behavior of an Order Release Mechanism in a Make-To-Order Manufacturing Systems with Selected Order Acceptance. *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1251-1259.

Ovacik. I. M. and R. Uzsoy. 1997. *Decomposition Methods for Complex Factory Scheduling Problems.* Kluwer Academic Publishers, Boston.

Philipoom, P. R. and T. D. Fry. 1992. Capacity-based Order Review/Release Strategies to Improve Manufacturing Performance. *International Journal of Production Research,* 30(11), 2559-2572.

Roderick, L. M., D. T. Phillips, and G. L. Hogg. 1992. A Comparison of Order Release Strategies in Production Control Systems. *International Journal of Production Research,* 30(3), 611-626.

Rose, O. 1999. CONLOAD – a New Lot Release Rule for Semiconductor Wafer Fabs. *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 850-855.

Rose, O. 2001. CONWIP-like Lot Release for a Wafer Fabrication Facility with Dynamic Load Changes. *Proceedings of the SMOMS '01 (ASTC '01)*, 41-46.

ACKNOWLEDGMENTS

The author would like to thank Jens Zimmermann, Technical University of Ilmenau, for his valuable programming efforts.

AUTHOR BIOGRAPHY

LARS MÖNCH is an Assistant Professor in the Department of Information Systems at the Technical University of Ilmenau, Germany. He received a master's degree in applied mathematics, a Ph.D. in the same subject from the University of Göttingen, Germany, and a Habilitation degree in Information Systems from Technical University of Ilmenau. His current research interests are in simulation-based production control of semiconductor wafer fabrication facilities, applied optimization, artificial intelligence applications in manufacturing and information systems for manufacturing processes. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), SCS and INFORMS. His email address is Lars.Moench@tu-ilmenau.de.