International Journal of Production Research

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Dispatching for make-to-order wafer fabs with machine-dedication and mask set-up characteristics

Muh-Cherng Wu [a] , Shau-Jie Chiou [a] & Chen-Fu Chen [a]

[a] Department of Industrial Engineering and Management , National Chiao Tung University , Hsin-Chu 300, Taiwan (ROC)

Published online: 12 Jun 2008.

To cite this article: Muh-Cherng Wu , Shau-Jie Chiou & Chen-Fu Chen (2008) Dispatching for make-to-order wafer fabs with machine-dedication and mask set-up characteristics, International Journal of Production Research, 46:14, 3993-4009, DOI: 10.1080/00207540601085919

To link to this article: http://dx.doi.org/10.1080/00207540601085919

International Journal of Production Research,
Vol. 46, No. 14, 15 July 2008, 3993–4009

Taylor & Francis
Taylor & Francis Group

Dispatching for make-to-order wafer fabs with machine-dedication and mask set-up characteristics

MUH-CHERNG WU*, SHAU-JIE CHIOU and CHEN-FU CHEN

Department of Industrial Engineering and Management, National Chiao Tung University, Hsin-Chu 300, Taiwan (ROC)

(*Revision received October 2006*)

This paper develops a dispatching algorithm to improve on-time delivery for a make-to-order semiconductor wafer fab with two special characteristics: mask set-up and machine dedication. A new algorithm is proposed for dispatching series workstations. Simulation experiments show that the algorithm outperforms the previous methods both in on-time delivery rate, cycle time, and only slightly less than the best benchmark in throughput. The experiments are carried out in 10 test scenarios, which are created by the combination of two product-mix ratios and five mask set-up times.

Keywords: Semiconductor dispatching; Make to order; On-time delivery rate; Machine dedication; Mask set-up

1. Introduction

Semiconductor manufacturing is a complex process. Hundreds of operations are required to produce a wafer. A semiconductor factory (also called a fab) typically involves several dozen workstations. A workstation is a group of functionally identical machines that process several operations on the same wafer. A job (also called a lot), which is a cassette that typically carries 25 wafers, may have to enter a workstation several times. Owing to the *re-entry* characteristics, we may have a great many types of WIP (work in process) waiting before a workstation for dispatching—a decision for determining which job should be processed first while a machine is available. The dispatching decision for a semiconductor fab is very important because it could significantly affect the fab performance such as on-time delivery, cycle time and throughput.

Dispatching decisions for a wafer fab have been studied extensively in the literature. Most studies have aimed to develop dispatching rules to reduce cycle time and increase throughput (Lu *et al*. 1994, Li *et al*. 1996, Yoon and Lee 2000). Some have intended to reduce the tardiness (Lu and Kumar 1991, Kim *et al*. 2001), while others have aimed to improve on-time delivery, in addition to an improvement in cycle time and throughput (Kim *et al*. 1998a, Lee *et al*. 2002, Dabbas and Fowler 2003). Yet most previous studies have assumed that there is no mask set-up time for a stepper.

*Corresponding author. Email: mcwu@cc.nctu.edu.tw

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2008 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540601085919

Steppers are very important machines in a fab, which essentially perform the *exposure operations*. An exposure operation is to 'photo-print' a circuit pattern onto a wafer by light projection through a mask, which records the circuit pattern. Different exposure operations require different masks. The change of mask on a stepper requires a set-up time. *Mask set-up* for a stepper should therefore be considered in wafer dispatching. Yet only a few studies on semiconductor dispatching (Kim *et al*. 1998b, Chern and Liu 2003) have been concerned with the mask set-ups of steppers.

In an up-to-date fab, one of the distinguished features is *machine dedication*. It demands a job being dedicatedly processed by a particular machine while it enters a workstation. That is, machines in a workstation cannot be taken as *exactly* identical for some critical operations. With the *machine-dedication* characteristic, the capacity of a workstation would be reduced because its machines cannot mutually support in capacity when processing the critical operations. Imposing a significant constraint on workstation capacity, the machine-dedication characteristic is therefore indispensable while developing dispatching decisions for an up-to-date fab.

Steppers in a fab can be categorized into two types: *high-resolution* and *low-resolution*. In a fab, the machine-dedication constraint is imposed *only* on *high-resolution steppers*. That is, once a wafer has been processed by a high-resolution stepper, its remaining exposure operations have to be processed by the same stepper. Other high-resolution steppers, even with the same specification, cannot process the wafer. The purpose of machine dedication is to ensure good manufacturing quality because, in reality, any two machines cannot be 'completely identical'—slight differences always exist. A low-resolution stepper workstation, on the contrary, has no machine-dedication feature. Therefore, any two steppers in such a workstation can support each other in capacity.

In the research literature, *mask set-up* and *machine dedication* are either only partially dealt with or both deal with in an 'old-technology' context where the set-up time is much longer than the state-of-art technology. Kim *et al*. (1998b) consider mask set-up but ignore machine dedication. Chern and Liu (2003) consider both mask set-up and machine dedication. However, their work was developed for a relatively old-technology fab, in which the set-up time for each mask is about 6 min while an up-to-date stepper in 2005 takes only about 1.5 min. Significant change in set-up time may affect the performance of dispatching policies and therefore the performances of dispatching algorithms should be compared under various mask set-up times.

The algorithm in Chern and Liu's (2003) work was essentially developed for *make-to-stock* (MTS) fabs, which have *high-volume* and *low-variety* characteristics, such as those making DRAM (dynamic random access memory). In contrast, the products manufactured by *make-to-order* (MTO) fabs usually have *low-volume* and *high-variety* characteristics. The main performance of an MTS fab is *throughput*, while that of an MTO fab is *on-time delivery*. Therefore, dispatching algorithms that are effective for an MTS fab may not perform as well in an MTO fab.

Owing to its *high-volume* and *low-variety* characteristics, an MTS fab is usually equipped with *multiple masks* for each exposure operation. In contrast, an MTO fab, with its *low-volume* and *high-variety* characteristics, usually adopts a *one-mask* policy in order to reduce manufacturing costs. In comparing the performance of

dispatching algorithms for an MTO fab we have to adopt a one-mask policy instead of a multiple-mask policy.

Considering the requirement of mask set-up, this research aims to develop dispatching methods for a fab that has *make-to-order* and *machine-dedication* features. Three performance metrics are considered, which involve *on-time delivery rate*, *throughput* and *cycle time*. Of these three, MTO fabs are most concerned with *on-time delivery* in order to retain or attract customers. A dispatching algorithm—*LBSA-F*—that utilizes ideas of line balance (LB), starvation avoidance (SA), and family-based dispatching (F) has been developed.

Simulation experiments based on the data provided by a real MTO fab are performed to evaluate the proposed algorithms. In the simulation experiments, a one-mask policy is adopted to reflect the characteristic of MTO fabs. And to justify the robustness of LBSA-F, scenarios with various mask set-up times are compared.

Four benchmark algorithms are used for comparison with the LBSA-F algorithm. These four include the LBSA-I (line balance, starvation avoidance, individual based) algorithm, the SDA-F algorithm by Chern and Liu (2003), the LWL-F algorithm by Kim *et al*. (1998b), and the FCFS-F (first-come-first-served family-based) algorithm. Results show that the LBSA-F algorithm outperforms the four benchmarks in terms of on-time delivery and cycle time; and is only slightly worse than the best benchmark in terms of throughput.

The remainder of this paper is organized as follows. Section 2 describes the two dispatching decisions that we focus on. Section 3 presents the proposed LBSA-F dispatching algorithm. Simulation experiment results are given in section 4. The reasons why LBSA-F would perform well are discussed in section 5. Concluding remarks are presented in the final section.

2. Research problem

The various decisions associated with the shop-floor control of a fab are first described. Among these decisions, only two—the *dispatching of dedicated and non-dedicated workstations*—are investigated in this research. The other decisions, but not a focus of this research, are dealt with by some existing methods in our simulation experiments.

2.1 *Releasing decisions*

Releasing decisions are to determine *when to release* a job to a fab, and determine *which job to release*. Methods for releasing decisions can be classified into two types: open-loop control and closed-loop control. Open-loop control denotes that a job is released to a fab based on a predetermined schedule, which is independent of the current status of the fab. Uniform releasing policy, a typical method of open-loop control, releases jobs 'uniformly' (Glassey and Resende 1988a). That is, the release rate and release pattern on each day is identical. Closed-loop control denotes that the time when a job is released depends on the current WIP status of the fab. Along the lines of closed-loop control, Glassey and Resende (1988b) developed a starvation avoidance (SA) algorithm; Wein (1988) developed a workload regulation (WR) algorithm; Bechte (1988) used a queuing model to compute the WIP threshold for

releasing new jobs; and Spearman *et al*. (1990) proposed a CONWIP (constant WIP) method. This research adopts the uniform releasing policies in the simulation experiments.

In a fab with *machine dedication*, at the time point of releasing a job, a *stepper-assignment decision* must be made. That is, the job should be assigned to a *high-resolution* stepper for processing the critical exposure operations of the job. In this research, the decision is based on the *accumulated load* of each *high-resolution* stepper. That is, at a job releasing time point, the job to be released is assigned to the high-resolution stepper that is the lowest in terms of accumulated load. The main idea of this *stepper-assignment* decision is to keep each high-resolution stepper balanced in load from a *long-term* perspective.

The stepper-assignment or machine-assignment decision can also be intricately formulated as a linear programming (LP) program if the cycle time between any two subsequent operations on a high-resolution stepper is certain and available (Gamila and Motavalli 2003, Liaw 2004). However, the cycle time in an MTO fab is usually with stochastic behaviour; the adoption of such LP formulations needs to be further justified. In the simulation experiments, we adopt the heuristic of balancing accumulated load because it is widely used in practice.

2.2 *Dispatching decisions*

Dispatching is the determination of which job to process among the jobs waiting before a workstation. Different types of workstations need various dispatching methods. In general, workstations in a fab can be classified into two types: a batch workstation and a series workstation. A *batch machine* processes several jobs at a time; for example, a furnace machine may process six jobs (150 wafers) simultaneously to reduce processing cost. In contrast, a *series machine* (e.g. a stepper machine) processes one wafer at a time until all the wafers in the job have been completed.

Many algorithms for the dispatching of batch workstations have been published (Weng and Leachman 1993, Kim *et al*. 1998b). Among these, the most commonly used one in industry is the minimum batch size (MBS) method. The MBS method denotes that the batch size (the number of jobs processed simultaneously) should exceed a predefined threshold, which can be determined by a queuing model (Neuts 1967, Phojanamongkolkij *et al*. 2002). While two or more batches meet the MBS threshold, the first-in-first-out (FIFO) rule is applied to break the tie in determining dispatching priorities.

High-resolution steppers are usually the *bottleneck* of a fab because they are very expensive and relatively limited in quantity. In a fab, only the high-resolution steppers have machine-dedication feature, while the others (either series or batch machines) do not have these characteristics. Since high-resolution steppers are a type of series machine, we therefore classify *series workstations* into two types: *dedicated* and *non-dedicated*. A typical dedicated workstation includes several high-resolution steppers, which are accommodated in a particular area but cannot support each other in capacity owing to the constraint imposed by machine dedication.

This research focuses on developing the dispatching algorithms for two types of *series workstations*, by assuming that the MBS dispatching algorithm has been applied to the *batch workstation*. The main objective is to maximize the *on-time delivery*,

and the other two performance criteria are *throughput* and *cycle time*. A semiconductor product (also called IC—integrated circuit) is a component of a consumer product such as a mobile (cell) phone and computer. Late delivery of MTO ICs would postpone the delivery of the consumer product, whose assembly needs many other components. As a result, the effect of IC delay would be amplified and lead to a substantial increase in the inventory of non-IC components. Therefore, on-time delivery is the most important objective in this research.

3. Dispatching algorithms

As stated above, series workstations involve two types: *dedicated* and *non-dedicated*. The proposed dispatching algorithm for each type is presented, where a series workstation is called simply a workstation for short.

3.1 *Dispatching for a dedicated workstation*

A dedicated workstation involves only high-resolution steppers, which require a mask in processing an operation. Different operations require different masks. A group of jobs that use the same mask is called a *job family*. When the mask needs to be changed, two dispatching decisions have to be made: (1) *choosing a job family*, and (2) *prioritizing jobs in the chosen job family*.

Research on mask change involves two main approaches—*individual based* and *family based* (Chern and Liu 2003). The *family-based* approach tends to keep processing the same job family. That is, the current mask will not be changed unless it has no job to process. In contrast, the *individual-based* approach requests that a mask-changing decision must be made whenever an operation is completed.

Adopting the family-based approach, this research develops a *line-balanced* (LB) method (Ignall 1965, Yamada and Matsui 2003) in dispatching job families. In the LB approach, the process route is decomposed into many *process segments*. One each process segment, its last operation is processed by the dedicated workstation while the others are processed by a non-dedicated workstation (see figure 1).

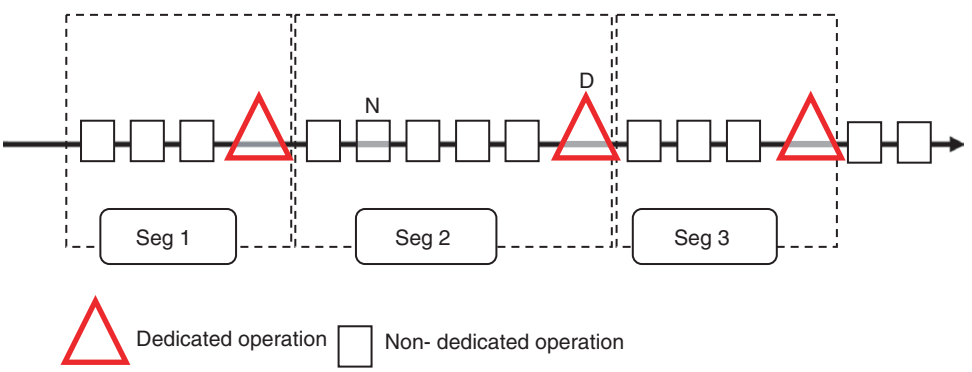

Figure 1. Segments in a process route.

The fab of interest produces a single product family that involves I products. Each product, with the same process route but different in terms of operation times, has J segments. Whenever a mask needs to be changed, the number of job families to be chosen is $(I * J) - 1$.

The procedure for dispatching a dedicated stepper is described below. To undergo the procedure, a pre-simulation has to be performed to determine CT_{ij}, which denotes the mean cycle time required to complete *all* the operations of product i in segment j. The estimation of CT_{ij} in the simulation assumes that the fab adopts the FCFS-F (first-come-first-served family-based) dispatching algorithm.

Procedure: Dispatching_Dedicated_Workstations

Step 1: Compute the flow rate (v_{ij}) for each job family as below, where WIP_{ij} denotes the number of jobs for the job family of product i at segment j:

$$v_{ij} = \frac{WIP_{ij}}{CT_{ij}}.$$

Step 2: Compute the normalized flow rate (λ_{ij}) as below, where R_i denotes the ratio of release rate (jobs per unit of time) for product i:

$$\lambda_{ij} = \frac{v_{ij}}{R_i} = \frac{WIP_{ij}}{CT_{ij} \cdot R_i}.$$

Step 3: Select the job family that has the maximum normalized flow rate:

$$(i^*, j^*) = Arg \max(\lambda_{ij}).$$

Step 4: Use the CR (critical ratio) rule to prioritize the jobs in the selected family.

The main idea of the above procedure is *line balancing*. Consider an *ideally* line-balanced production line where the flow rate (jobs per day) of each product in each segment (v_{ij}) can be so well controlled that it always equals its release rate R_i. Then the fab output rate equals the release rate R_i. That is, in the ideally line-balanced case, $\lambda_{ij} = v_{ij}/R_i = 1$ for each i and j. The deviation of λ_{ij} from 1 indicates the degree of imbalance for product i in segment j.

While ideally line balanced, the standard WIP level for segment j of product i is $Std\_WIP_{ij} = CT_{ij} \cdot R_i$. That is,

$$\lambda_{ij} = \frac{WIP_{ij}}{CT_{ij} \cdot R_i} = \frac{WIP_{ij}}{Std\_WIP_{ij}}.$$

The job family with the highest λ_{ij} should be processed first in order to smooth the WIP distributions among segments and head for line balancing.

In Step 4, to maximize the on-time delivery rate, we use the CR (critical ratio) method to prioritize the jobs in the selected job family. The CR of a job denotes the ratio of its remaining time over its remaining processing time, which is intended to measure the possibility of on-time delivery. The lower the CR value, the lower the possibility of on-time delivery and therefore the job should be processed first. The use of another dispatching rule such as SRPT (shortest remaining processing time) might be a good heuristic for other performance criteria such as

throughput (Walrand 1988). However, the present research is concerned more with on-time delivery and therefore CR is proposed.

3.2 *Dispatching for non-dedicated workstations*

For a non-dedicated workstation, the number of job families can be $I * J * K$, where I denotes the number of products, J denotes the number of route segments, and K denotes the number of dedicated steppers. Likewise, there are two decisions for the dispatching of non-dedicated workstations: (1) choosing job family, and (2) prioritizing the jobs for the chosen job family.

This research uses the concept of starvation avoidance (SA) (Glassey and Resende 1988b) to choose the job family. As stated above, the dedicated steppers are a bottleneck (Lee *et al*. 2002) in a fab; therefore, it is important to supply enough jobs to each dedicated stepper to prevent it from being starved.

The procedure for dispatching non-dedicated workstations is presented below, where N denotes the workstation for making the dispatching decision and D denotes the dedicated-stepper workstation (see figure 1). To undergo the procedure, a pre-simulation has to be carried out in order to determine CT_{ijk}, which denotes the mean cycle time of the job family (product i in segment j assigned to dedicated-stepper k) from workstations N to D.

Procedure Dispatching_Non-dedicated_Workstations

Step 1: Compute the flow rate (v_{ijk}) for each job family as below, where WIP_{ijk} denotes the WIP level of the job family (product i in segment j assigned to dedicated-stepper k) from workstations N to D:

$$v_{ijk} = \frac{WIP_{ijk}}{CT_{ijk}}.$$

Step 2: Compute the *normalized* flow rate (λ_{ijk}), as below, where R_i denotes the ratio of the release rate for product i:

$$\lambda_{ijk} = \frac{v_{ijk}}{R_i}.$$

Step 3: Select the job-family that has the minimum normalized flow rate:

$$(i^*, j^*, k^*) = Arg\min(\lambda_{ijk}).$$

Step 4: Use CR (critical ratio) to prioritize the jobs in the selected job family.

3.3 *Comparison of the two dispatching algorithms*

Of the above two dispatching algorithms, the one for *dedicated steppers* is to balance the *throughput among segments*, and is called *line-balancing* (LB) dispatching. The other one, for non-dedicated workstations, is to prevent dedicated steppers from being 'starved', and is called *starvation-avoidance* (SA) dispatching.

Line-balancing dispatching is designed from the perspective of controlling the *output mix of job families* that leave from dedicated steppers (bottlenecks). In contrast, SA dispatching is designed from the perspective of controlling the

input mix of job families that arrive to dedicated steppers. The output control aims to produce a product at a rate as close as possible to its release rate. The input control aims to provide enough WIPs to dedicated steppers for them to effectively realize the output control.

The mask allocation decision in the dispatching algorithm for dedicated steppers is discussed below. When only one stepper is available at a particular time instant, we keep using the same mask if WIP jobs exist that need the mask for processing. If there are no such jobs, we have to change mask. The mask we can choose is limited to those that are not presently used by other steppers.

Notice that, in very rare cases, two steppers may be available and need to change mask at the same time. In such an eventuality we prioritize the two steppers randomly in choosing a new mask.

4. Simulation experiments

4.1 *Data, assumptions, and benchmarks*

The proposed dispatching algorithms are compared with four benchmark methods by discrete-event simulation. The test data for the process route and processing times are provided by an MTO fab in industry. The fab involves 60 workstations, of which nine are batch workstations and 51 are series workstations, and the workstations in total involve 262 machines. The MTBF (mean time between failure) and MTTR (mean time to repair) of machines are also available, with the assumption of exponential distributions.

The fab—an MTO fab—adopts the one-mask policy. A single product family, involving five logical products, is produced. Each product has the same process route, which involves 12 segments and 344 operations (see table 1). Taking a particular product as a standard, the processing time of the other four products

Table 1. Process route and processing times of the test fab.

Segment	Number of operations	Processing time (h)
1	12	27.97
2	67	87.91
3	89	78.15
4	19	17.81
5	15	11.23
6	19	19.74
7	15	11.23
8	19	19.74
9	15	11.23
10	19	19.74
11	15	11.23
12	40	39.09
Total	344	355.07

is modelled by multiplying the standard by a uniform distribution, UNIF(0.95, 1.05). The exposure operation time for a lot (25 wafers) is 1.66 h = 100 min.

Each job or lot has 25 wafers. The due date of lot k is defined by $\delta_k = a_k + u \cdot pt_k$, where δ_k denotes the due date, a_k denotes the release time, and pt_k denotes the total processing time, and u denotes a scale factor for defining due date (Kim *et al*. 1998a). Note that $u \cdot pt_k$ is also called the *committed cycle time*, which indicates the cycle time committed to customers. If the *production cycle time* of a lot is longer than $u \cdot pt_k$, the lot will be late. Since the fab delivers the wafer lots to the customer once a day, δ_k (in units of a day) is rounded up to an integer.

Notice that the value of u (the due date factor) is a constant for all lots. We determine the value of u by a heuristic approach. First, we carry out a simulation for a particular scenario (for example, $s = 90$ s, product mix $= R_A$) that adopts the FCFS-F dispatching algorithm. Then, we try using various values of u to obtain a near 50% on-time delivery rate. Referring to table 3, we have chosen a value for u such that the on-time delivery rate is 48%.

Three performance metrics, the *on-time delivery rate*, *throughput*, and *mean cycle time* are to be compared. Of the three, *on-time delivery rate* is most critical for a competitive MTO fab to retain or attract customers. Adopting a uniform-releasing policy, the fab releases 31 lots per day in total.

Two product mix ratios, which are described by $R_A = (1{:}1{:}1{:}1{:}1)$ and $R_B = (1{:}2{:}3{:}1{:}2)$, are used to evaluate the dispatching algorithms. Five cases of mask set-up time ($s = 0, 30, 90, 180, 360$, in units of seconds) are evaluated, where $s = 90$ is the current practice of an up-to-date fab. In total, 10 cases—the combination of two product mixes and five mask set-up times, are tested.

Each simulation experiment is performed with 20 runs; each run is with a different random seed. The time horizon for a simulation run is 270 days; the first 90 days are taken as 'warm-up' time, after which the WIP and throughput will have reached a steady state (see figure 2). The output data of the subsequent 180 days is collected for analysis. Simulation programs, coded in eM-plant (http://www.tecnomatix.com), are run on a personal computer equipped with an AMD-3000$^+$ CPU.

The proposed dispatching algorithm is designated as *LBSA-F*, where *LB* denotes line balance, *SA* denotes starvation avoidance, and *F* denotes a family-based approach for mask dispatching. Four algorithms are compared with the *LBSA-F*. The first one—*LBSA-I*—also developed by us, is a variation of *LBSA-F*, with *I* denoting the use of the individual-based approach to mask dispatching. The second one—*SDA-F*—denotes the algorithm proposed by Chern and Liu (2003). The third one—*FCFS-F*—used widely in industry, denotes the first-come-first-served algorithm with a family-based approach to mask dispatching. The fourth one—*LWL-F* (loop workload levelling, family based)—denotes the algorithm proposed by Kim *et al*. (1998b). A comparison of these algorithms is summarized in table 2.

4.2 *Experiment results for s = 90*

As stated above, the experiments involved five mask set-up times ($s = 0, 30, 90, 180, 360$, in units of seconds). The case of $s = 90$ is of most interest

Table 2. A comparison of dispatching algorithms.

Dispatching algorithm	Dedicated workstation	Non-dedicated workstation	
	Steppers	Steppers	Non-steppers
LBSA-F	LB-F	SA-F	SA
LBSA-I	LB-I	SA-I	SA
SDA-F	SDA-F	SDA-F	FCFS
FCFS-F	FCFS-F	FCFS-F	FCFS
LWL-F	LWL-F	LWL-F	FCFS

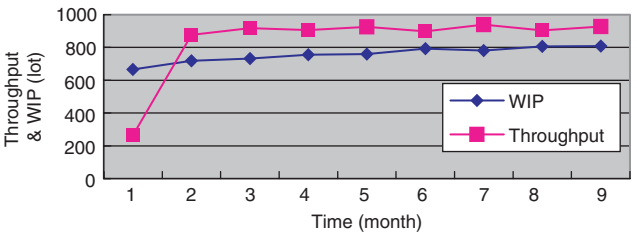

Figure 2. Time plots of throughput and WIP.

because this is the current practice of an up-to-date fab. Experiment results for the two product mixes (R_A and R_B) at $s=90$ are analysed below.

Table 3 shows the mean and standard deviation of each performance metric under various dispatching algorithms. An analysis of variance (ANOVA) was carried out to justify the effects of the dispatching rules (Montgomery 1991). The ANOVA results (see table 4) show that the dispatching rules had a significant effect on each performance metric (at a significance level of 0.01) in each product mix. Duncan's multiple range tests were also performed to categorize the dispatching rules based on their performances, the results of which are given in table 5.

From these results, we can conclude that *LBSA-F* outperforms the four benchmarks in terms of *on-time delivery rate* and *cycle time*, in each product mix (R_A and R_B). Yet, in terms of throughput, *LBSA-F* performs the best in product mix R_A but ranks third in product mix R_B.

The reason why *LBSA-F* in R_B does not perform as well as that in R_A is analysed below. Compared with $R_A=(1{:}1{:}1{:}1{:}1)$, the production volume of each product in $R_B=(1{:}2{:}3{:}1{:}2)$ is less uniform. Using the normalized flow rate (λ_{ij}) as the main dispatching criterion, *LBSA-F* tends to make the on-time delivery rate of each product as close as possible. Therefore, in dealing with *small-volume* products, masks have to be changed more frequently. This implies an increase in total mask set-up time, which consequently leads to a decrease in bottleneck utilization and fab throughput. The above analysis is supported by the experiment results of *LBSA-F*, which indicated that the average utilization of the dedicated stepper in R_A is 99.38% while that in R_B is 99.25%.

Table 3. Experiment results for $s = 90$ s: (a) product mix R_A, and (b) product mix R_B.

Dispatching algorithm	On-time delivery rate		Cycle time		Throughput	
	Mean (%)	SD (%)	Mean (days)	SD (days)	Mean (lot)	SD (lot)
(a) R_A						
LBSA-F	88	3.2	23.4	1.3	5523.4	9.6
LBSA-I	78	2.4	23.8	2.7	5507.0	11.3
SDA-F	9	1.8	28.8	2.8	5341.9	27.3
FCFS-F	48	10.1	25.2	1.4	5504.7	12.9
LWL-F	49	6.4	25.3	1.9	5520.2	13.2
(b) R_B						
LBSA-F	89	2.7	23.4	1.3	5512.9	8.0
LBSA-I	82	1.3	23.6	3.6	5507.5	9.3
SDA-F	21	4.6	27.6	2.9	5338.9	14.7
FCFS-F	70	6.0	24.5	1.1	5541.7	10.9
LWL-F	55	5.4	25.0	1.7	5533.1	17.4

Table 4. ANOVA for $s = 90$ s: (a) product mix R_A and (b) product mix R_B.

	SS	df	MS	F	p
(a) R_A					
Throughput					
Dispatching rules	4.78E + 05	4	1.20E + 05	457	0
Error	2.48E + 04	95	2.62E + 02		
Cycle time					
Dispatching rules	360.74	4	90.18	1418	0
Error	6.04	95	0.06		
On-time delivery rate					
Dispatching rules	7.62618	4	1.90655	585.761	0
Error	0.30921	95	0.00325		
(b) R_B					
Throughput					
Dispatching rules	5.63E + 05	4	1.41E + 05	892	0
Error	1.50E + 04	95	1.58E + 02		
Cycle time					
Dispatching rules	229.79	4	57.45	1070	0
Error	5.1	95	0.05		
On-time delivery rate					
Dispatching rules	5.71755	4	1.42939	743.15	0
Error	0.18272	95	0.00192		

Compared with *LBSA-I*, *LBSA-F* performs better in each performance metric. This finding seems reasonable because the *LBSA-I*, an individual-based algorithm, tends to change mask more frequently and consequently reduce throughput. Since both *LBSA-F* and *LBSA-I* use the normalized flow rate (λ_{ij}) as the main dispatching criterion, the reduction of throughput in *LBSA-I* tends to reduce its on-time delivery. This finding indicates that $s = 90$ s is a substantial amount of time in terms of mask set-up, and cannot be ignored in developing dispatching algorithms.

Table 5. Duncan's multiple range test for $s = 90$: (a) product mix R_A and (b) product mix R_B.

Rule	Throughput	Results	Rule	CT	Results	Rule	On-time delivery rate	Results
(a) R_A								
LBSA-F	5523.400	A	LBSA-F	23.4481	A	LBSA-F	0.877284	A
LWL-F	5520.200	A	LBSA-I	23.8143	B	LBSA-I	0.780176	B
LBSA-I	5507.000	B	FCFS-F	25.1950	C	LWL-F	0.492436	C
FCFS-F	5504.700	B	LWL-F	25.2575	C	FCFS-F	0.476221	C
SDA-F	5341.850	C	SDA-F	28.8196	D	SDA-F	0.088803	D
(b) R_B								
FCFS-F	5541.700	A	LBSA-F	23.40683	A	LBSA-F	0.888542	A
LWL-F	5533.100	B	LBSA-I	23.58496	B	LBSA-I	0.815768	B
LBSA-F	5512.900	C	FCFS-F	24.48898	C	FCFS-F	0.700032	C
LBSA-I	5507.500	C	LWL-F	25.00969	D	LWL-F	0.548924	D
SDA-F	5338.900	D	SDA-F	27.61668	E	SDA-F	0.213940	E

4.3 *Experiment results for various cases of s*

Over the years, the mask set-up time has been progressively reduced as a result of technological advances. To justify the performance of the *LBSA-F* algorithm in various fabs, from *a traditional fab* to a *future one*, simulation experiments were performed for 10 test cases, representing the combination of five mask set-up times ($s = 0, 30, 90, 180, 360$) and two product mixes (R_A and R_B).

Figures 3 and 4, respectively, show the experiment results in product mixes R_A and R_B. In the two figures, the performance of *FCFS-F* is taken as a *baseline* for comparison. That is, the figures show the performance difference between a dispatching algorithm and *FCFS-F*. As the trends in the two figures appear quite consistent, we refer to figure 3 in analysing the experiment results.

Figure 3(a) indicates that, in terms of on-time delivery rate, *LBSA-F* outperforms the other four algorithms for each s. A comparison between *LBSA-F* and *SDA-F* indicates that their difference in performance is increased when s becomes smaller. It reveals that *SDA-F* performs well in the case of $s = 360$ but not so well when $s = 90$ or smaller. Seemingly, the smaller the mask set-up time, the higher their performance differences. This indicates that the variation in mask set-up time indeed affects the performance of dispatching algorithms and cannot be ignored.

Figure 3(b) reveals that *LBSA-F* also outperforms the other four algorithms for each s, in terms of mean cycle time. In the experiments, the due date of each job was predetermined. Therefore, the shorter the production cycle time, the higher the on-time delivery rate. The finding in relation to on-time delivery and cycle time appear quite consistent.

Figure 3(c) reveals that *LBSA-F* performs well for each s, in terms of throughput. The performance of *LBSA-F* differs only slightly from the best benchmark for each s. As shown in the figure, the throughput of *LBSA-I* performs well when $s = 0, 30, 90$, but drops significantly when $s = 180$ and 360. This implies that family-based dispatching algorithms are preferred in those cases requiring long mask set-up time.

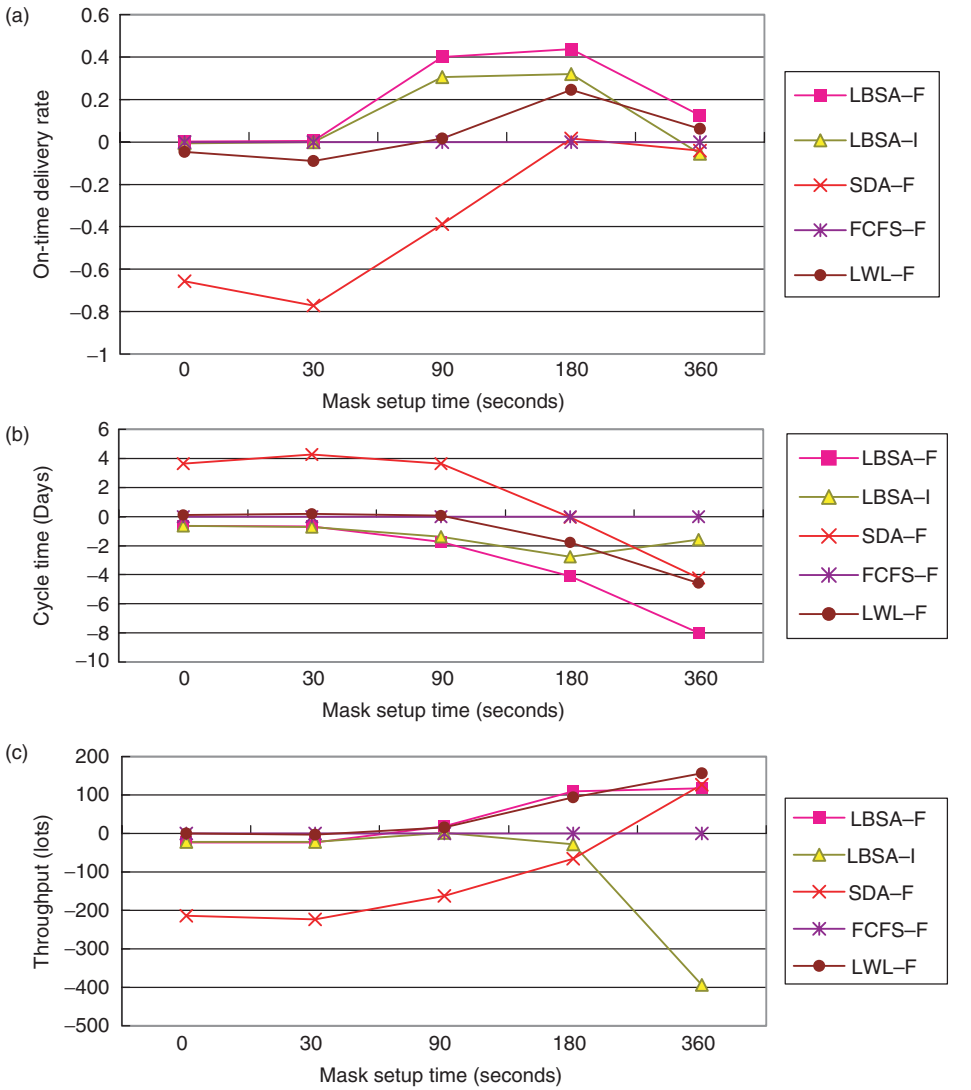

Figure 3. Performance comparison in various mask set-up times, with product mix R_A: (a) on-time delivery rate, (b) cycle time, and (c) throughput.

4.4 *Comparison between LBSA-F and the benchmarks*

In highlighting the distinguishing features of *LBSA-F* we hope to explain why it outperforms the benchmarks. Compared to the benchmarking methods, our research is distinct in providing an *integrated dispatching architecture*. That is, the LB dispatching criterion is used in dedicated steppers, and the *SA* dispatching criterion is used in other series workstations. The *LB* and *SA* dispatching criteria are so *integrated* because they are both designed to make effective use of the dedicated steppers. In contrast, benchmarking methods such as those described in

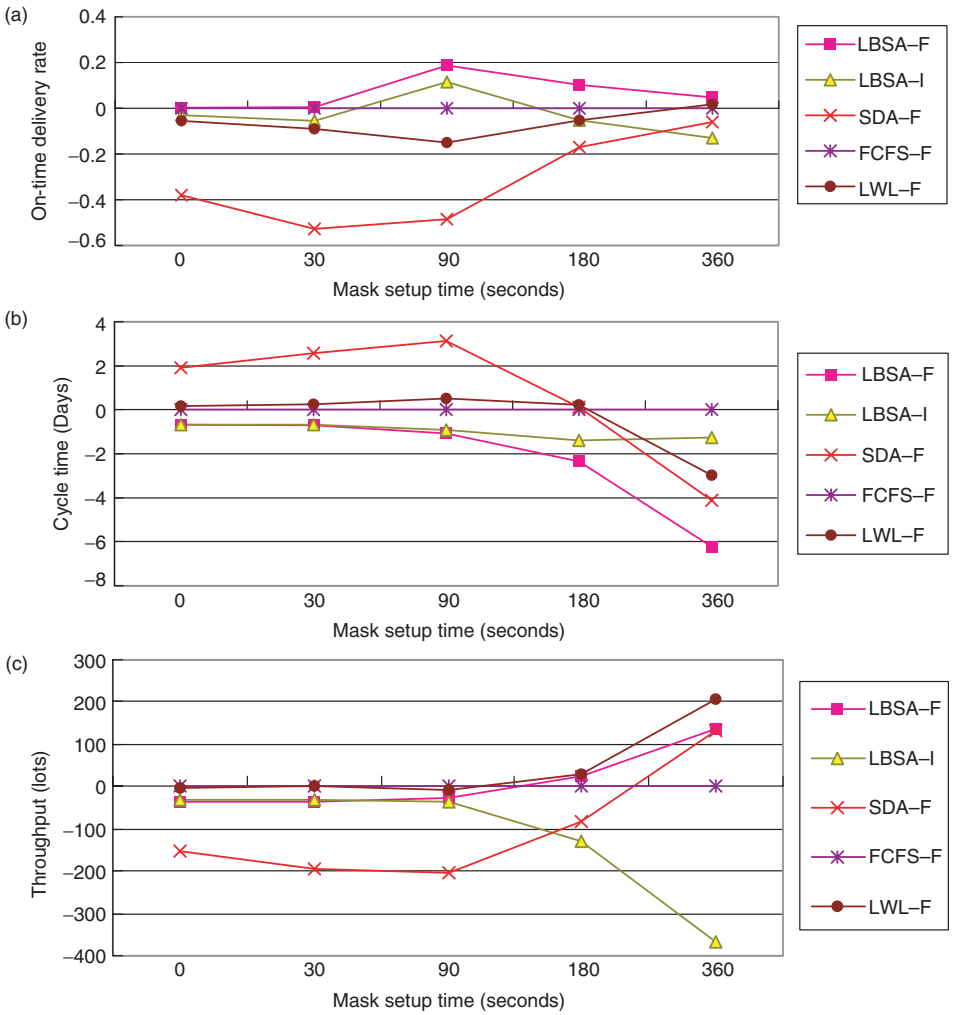

Figure 4. Performance comparison for various mask set-up times, with product mix R_B: (a) on-time delivery rate, (b) cycle time, and (c) throughput.

Kim *et al.* (1998b) pay more attention to the dispatching of steppers than to other workstations. Also, typical dispatching methods such as FCFS are used for non-stepper workstations (see table 2).

For the dispatching of steppers, the LB method, similar in part to the LWL (loop workload levelling) method (Kim *et al.* 1998b), has its own merits. A brief comparison of LWL and LB is presented below. In the LWL method, a loop is just like a segment in our research. They define the *loop workload* of segment k as $W_k^L = p_k \cdot Q_k / T_k$, where p_k denotes the exposure processing time per unit in segment k, Q_k denotes the total number of jobs in segment k, and T_k denotes the total processing times of all the operations (excluding the exposure one) in segment k. In the dispatching for steppers, they essentially use W_k^L as the decision criteria.

In contrast, our research models the *normalized flow rate* of each segment for each product as:

$$\lambda_{ij} = \frac{v_{ij}}{R_i} = \frac{1}{R_i}\left(\frac{WIP_{ij}}{CT_{ij}}\right).$$

Compared with W_k^L, λ_{ij} is distinct in two ways. First, the inclusion of R_i (the release rate of product i) helps to determine the dispatching priority between different products. Second, the inclusion of CT_{ij} (the cycle time of a product in a particular segment) helps determine the dispatching priority among segments more appropriately.

5. Discussion

As stated above, the fab of interest involves two distinguishing features: *make to order* and *machine dedication*. We attempt to explain why *LBSA-F* would perform well in such a fab.

The *make-to-order* feature would lead to *a high variety of job families* waiting before the bottleneck (a dedicated stepper). In the test examples, the process route contains 12 segments in which 11 segments involve dedicated operations (see table 1). That is, a dedicated stepper has to process a product 11 times with five products produced simultaneously. This implies that 55 types of job family would be waiting before a dedicated stepper. In practice, this number could be 10 times larger.

The *machine-dedication* feature would lead to a *significant reduction* in *the total WIPs* waiting before a dedicated stepper. Consider a workstation that involves 11 steppers and is having Q jobs waiting for processing. If the steppers are non-dedicated, the total WIPs available for a particular stepper is Q. While the steppers become dedicated, the total WIPs available for a particular stepper on average reduce to $Q/11$.

The above analysis indicates that a *make-to-order* fab with a *machine-dedication* feature would yield such a result—the WIPs waiting before a dedicated stepper have *high-variety* and *low-volume* characteristics. These characteristics also hold for the *non-dedicated workstations* on the upstream of the dedicated stepper. By contrast, in the case of *make-to-stock* fabs without a *machine-dedication* feature, the WIPs waiting before a workstation have relatively *low-variety* and *high-volume* characteristics.

The main performance metric for a make-to-order fab is *on-time delivery*. To maximize on-time delivery, *LBSA-F* attempts to smooth *the normalized flow rate of each job family* at each segment. That is, each segment of a product is urged to have the same *output rate*, preferably as close to its *release rate* as possible. This tends to reduce the variation in segment flow rate, which leads to a reduction in the variation of output rate. This implies that the cycle time deviation of each product tends to be reduced; as a result, its on-time delivery rate may be improved. Contrariwise, if the output rate of each product is not well controlled, some lots of the product may output earlier and some may output later than the due dates. This tends to lower on-time delivery rate.

In *LBSA-F*, the dispatching for *non-dedicated* workstations has considered the *downstream machine-dedication constraint*. By contrast, most previous algorithms ignored this constraint and tended to lead to an unbalanced WIP profile for the dedicated stepper. As a result, their performance in on-time delivery was reduced.

6. Concluding remarks

This research considers the requirement of mask set-up and develops dispatching algorithms for a make-to-order fab with machine-dedication feature. The dispatching algorithms are evaluated by three performance metrics: on-time delivery rate, cycle time, and throughput. Of the three, on-time delivery rate is most critical to a make-to-order fab in order to retain and attract customers.

We have proposed a dispatching algorithm—*LBSA-F*, which uses the idea of *line balancing* (LB) to control the output pattern of a bottleneck, the idea of *starvation avoidance* (SA) to control the input pattern of a bottleneck, and the idea of a *family-based* approach to mask dispatching.

Simulation experiments have been performed in 10 test cases that are the combination of two product mixes and five mask set-up times. Four benchmark algorithms are used for comparison with *LBSA-F*. Experiment results show that *LBSA-F* outperforms the four benchmarks both in on-time delivery rate and cycle time, and is slightly worse than the best benchmark in terms of throughput.

Some extensions of this research could be investigated. First, the effects of other shop-floor control decisions on *LBSA-F* could be examined. These shop-control decisions include the determination of job releasing time, the assignment of jobs to dedicated machines at the time of job release, and the dispatching of batch machines. Second, dispatching algorithms for fabs with machine-dedication features in the context of producing both MTO and MTS products could be studied. As stated above, the main performance metric for MTO is on-time delivery and that for MTS is throughput. In such a hybrid production environment, the two performance metrics are both very important. To ensure good performance in each performance metric, the dispatching priorities between MTO and MTS products may have to be determined dynamically. This initiates a need for enhancing LBSA-F to perform well in such a hybrid product environment.

Acknowledgements

The authors are grateful to the anonymous referees for their careful reading of the manuscript and for providing positive feedback. We also thank Taiwan Semiconductor Manufacturing Corporation for providing the data for the simulation experiments. This research is partially supported by research project NSC90-2622-E009-001.

References

Bechte, W., Theory and practice of load-oriented manufacturing control. *Int. J. Prod. Res.*, 1988, **26**, 375–395.

Chern, C.C. and Liu, Y.L., Family-based scheduling rules of a sequence-dependent wafer fabrication system. *IEEE T. Semiconduct. Manuf.*, 2003, **16**, 15–25.

Dabbas, R.M. and Fowler, J.W., A new scheduling approach using combined dispatching criteria in wafer Fabs. *IEEE T. Semiconduct. Manuf.*, 2003, **16**, 501–510.

Gamila, M.A. and Motavalli, S., A modeling technique for loading and scheduling problems in FMS. *Robot. Comput.-Int. Manuf.*, 2003, **19**, 45–54.

Glassey, C.R. and Resende, M.G.C., A scheduling rule for job release in semiconductor fabrication. *Oper. Res. Lett.*, 1988a, **7**(5), 213–217.

Glassey, C.R. and Resende, M.G.C., Closed-loop Jop shop release control for VLSI circuit manufacturing. *IEEE T. Semiconduct. Manuf.*, 1988b, **1**, 26–46.

Ignall, E.J., A review of assembly line balancing. *J. Ind. Eng.*, 1965, **16**(4), 244–254.

Kim, Y.D., Kim, J.G., Choi, B. and Kim, H.U., Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. *IEEE T. Robot. Autom.*, 2001, **17**, 589–598.

Kim, Y.D., Kim, J.U., Lim, S.K. and Jun, H.B., Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. *IEEE T. Semiconduct. Manuf.*, 1998a, **11**, 155–164.

Kim, Y.D., Lee, D.H. and Kim, J.U., A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities. *J. Manuf. Syst.*, 1998b, **17**(2), 107–117.

Lee, Y.H., Park, J. and Kim, S., Experimental study on input and bottleneck scheduling for a semiconductor fabrication line. *IIE Trans.*, 2002, **34**, 179–190.

Li, S., Tang, T. and Collins, D.W., Minimum inventory variability schedule with applications in semiconductor fabrication. *IEEE T. Semiconduct. Manuf.*, 1996, **9**, 145–149.

Liaw, C.F., Scheduling two-machine preemptive open shops to minimize total completion time. *Comput. Res.*, 2004, **31**, 1349–1363.

Lu, S.C.H., Ramaswamy, D. and Kumar, P.R., Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE T. Semiconduct. Manuf.*, 1994, **7**, 374–388.

Lu, S.H. and Kumar, P.R., Distributed scheduling based on due dates and buffer priorities. *IEEE T. Automat. Contr.*, 1991, **36**, 1406–1416.

Montgomery, D.C., *Design and Analysis of Experiments*, 1991 (Wiley: New York).

Neuts, M.F., A general class of bulk queue with Poisson input. *Ann. Math. Stat.*, 1967, **38**, 759–770.

Phojanamongkolkij, N., Fowler, J.W. and Cochran, J.K., Determining operating criterion of batch processing operations for wafer fabrication. *J. Manuf. Syst.*, 2002, **21**, 363–379.

Spearman, M.L., David, L.W. and Wallace, J.H., CONWIP: a pull alternative to Kanban. *Int. J. Prod. Res.*, 1990, **28**, 879–894.

Walrand, J., *An Introduction to Queuing Networks*, 1988 (Prentice Hall: Reading).

Wein, L.M., Scheduling semiconductor wafer fabrication. *IEEE T. Semiconduct. Manuf.*, 1988, **1**, 115–130.

Weng, W.W. and Leachman, R.C., An improved methodology for real-time production decisions at a batch-process work station. *IEEE T. Semiconduct. Manuf.*, 1993, **6**, 219–225.

Yamada, T. and Matsui, M., A management design approach to assembly line systems. *Int. J. Prod. Econ.*, 2003, **84**, 193–204.

Yoon, H.J. and Lee, D.Y., A control method to reduce standard deviation of flow time in wafer fabrication. *IEEE T. Semiconduct. Manuf.*, 2000, **13**, 389–392.