8-2013

IMPROVED PHOTOLITHOGRAPHY SCHEDULING IN SEMICONDUCTOR MANUFACTURING

Sreenath Chalil madathil
Clemson University, schalil@clemson.edu

IMPROVED PHOTOLITHOGRAPHY SCHEDULING IN SEMICONDUCTOR MANUFACTURING

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Industrial Engineering.

---

by
Sreenath Chalil Madathil
August 2013

Accepted by:
Dr. Scott J Mason, Committee Chair
Dr. Anand K Gramopadhye
Dr. Mary E Kurz

ABSTRACT

Photolithography is typically the bottleneck process in semiconductor manufacturing. In this thesis, we present a model for optimizing photolithography job scheduling in the presence of both individual and cluster tools. The combination of individual and cluster tools that process various layers or stages of the semiconductor manufacturing process flow is a special type of flexible flowshop. We seek separately to minimize total weighted completion time and maximize on-time delivery performance. Experimental results suggest that our mathematical- and heuristic-based solution approaches show promise for real world implementation as they can help to improve resource utilizations, reduce job completion times, and decrease unnecessary delays in a wafer fab.

DEDICATION

This thesis is dedicated to the four pillars of my life: Mother, Father, Teacher and God

- To my lovely mother, Remani Chalil Madathil, for her unconditional love and support
- To my father, Haridas Chenicherri Veettil, who is my role model and inspiration
- To all my teachers who helped me to achieve my aspirations
    - C M Narayanan, my uncle, in teaching compassion and selflessness
    - Dr. C Sheela and Mrs. Archana Sreenivasan of the Federal Institute of Science and Technology (FISAT)
    - Mr. Senthilkumar Neelamegam, a friend, philosopher and guide
- To Paradevatha of Chalil Madam Family and Goddess Shree Mookambika Devi for their blessings and for enhancing my spiritual inclinations

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor and committee chair, Dr. Scott J. Mason, who passed on to me his enthusiasm and spirit of adventure in doing research. Without his guidance and timely advice, this thesis would not have been possible. The efforts that he made in conveying the ideas helped me to make this research an interesting journey. I am thankful to Dr. Anand K. Gramopadhye for providing suggestions and ways to improve my research. I would like to thank Dr. Mary E. Kurz for her advice on heuristic construction and suggestions to improve my results. I also express my deepest appreciation to Ms. Saundra Holland for helping me to schedule the meetings and presentations in order to accomplish my thesis requirements.

My colleagues and friends Kasin Ransikarbum, Sherif Masoud, Mariah Magagnotti, Githin Alapatt, Reshmi Koikkara, Melissa Paul, Aby Thyparambil, and Mahesh Sreedharan deserve a special mention in providing help and support during various stages of my thesis development. I also would like to thank Martin Clark for his help on computer and software related issues.

The unconditional support, love, and affection from my family always provided me great strength to sail through the various challenges that the last two years posed to me. I express my deep gratitude to my family members especially Preethi Dhanvi, Arekh Raghunathan, Aparna Ashok, Kapil Chalil Madathil, Dr. Sushama Chalil Madathil, Suja Nambiar, Jyothi Chalil Madathil, Amita Nambiar, Sunil Vijayan, Deepthi Rajagopal, Reshma Mohan, Rahul Mavila, Sreekanth Mavila, Shanu Jain, Anoop Raghunathan, Prisha Nambiar and Anika Nambiar. Finally and most importantly I would like to thank my parents, my relatives, all the teachers during the various stages of my life, and God Almighty.

TABLE OF CONTENTS

Page

TITLE PAGE i

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

1. INTRODUCTION 1

2. LITERATURE REVIEW 3

3. PROBLEM DESCRIPTION 6

4. MATHEMATICAL MODEL 9

4.1. Notation 9

4.2. Model Formulation 10

5. MODEL VALIDATION 14

6. EXPERIMENTAL STUDY 18

6.1. Experimental Plan 18

6.2. A Constructive Heuristic 21

6.3. Experimental Results and Analysis 23

Table of Contents (Continued)

Page

6.4. Conclusions and Future Work..... 25

REFERENCES.....27

LIST OF TABLES

Table Page

1. Notation for flexible flowshop scheduling model 9

2. Input data for example problem..... 14

3. Input data for example problem..... 15

4. Resulting schedule for the sample model 17

5. Experimental design 20

6. Machine counts for each experimental scenario..... 21

7. Average performance ratios for each experimental factor level 24

8. Average heuristic ratios for each experimental factor level 25

LIST OF FIGURES

Figure Page

1. Schematic Diagram for the Photolithography Process 7

2. Gantt chart for the sample problem 16

1. INTRODUCTION

Scheduling and sequencing are indispensable processes in industry. A well-designed scheduling system helps the industry focus on increasing throughput by reducing the run time of machines, thereby saving money. Processing jobs on the basis of "first-come, first-serve" may not be an optimal policy on the factory floor [1]. The semiconductor wafer fabrication industry is one of the largest industrial manufacturing segments. Implementing a proper scheduling system in wafer fabrication can help increase profit margins as well as reduce the time required to produce the wafers that contain integrated circuits.

In semiconductor manufacturing, photolithography is normally one of the bottleneck processes that require high capital investments [2]. Hence optimizing the photolithography process by efficiently scheduling the jobs could be beneficial for the industry. Machines that perform various steps in photolithography can be organized as a flexible flowshop system. A flexible flowshop is defined as a system in which the jobs need to be processed at different sequential stages and at least one of the stages has more than one machine operating in parallel. With the advancement of technology and because of their efficiency and profitability, cluster tools were added to the wafer fabrication processes in recent years. A cluster tool combines various types of machines that performs individual processes and organizes them around a robotic wafer transport device [3]. These tools consist of those machines that are capable of processing two or more stages and combine several processing modules into a single machine [4].

In my thesis research, I propose to develop a scheduling model for the photolithographic process, which is a special type of flexible flowshop (FFS) that has cluster tools along with the traditional individual photolithography tools. Each of the jobs that enter the system typically re-visits equipment visited at earlier manufacturing (i.e., reentrant flow). If the proposed model is tested successfully, it could be implemented in the semiconductor industry that employs

photolithography machines with advanced cluster tools. Wafer fabs will be able to schedule their machines to improve utilization of the machines, reduce the processing time for jobs and efficiently schedule without introducing unnecessary delays in the process.

2. LITERATURE REVIEW

The four basic processes involved in manufacture of integrated circuits are wafer fabrication, wafer probe, assembly and packaging, and final testing [5]. The wafer fabrication process includes complex procedures and technologies that involve high capital investments. The proper utilization of wafer fabs will eventually lead to increased profit for a semiconductor wafer fabricator. Each time a wafer passes through photolithography, a new layer of required circuitry is formed on the wafer. For most wafers there will be at least 25 such layers. Since the photolithography process is repeated during wafer fabrication, overall performance of the systems is improved by improving the photolithography process [6]. The high capital cost of the photolithography tools tends to force the wafer manufacturers to streamline the processes in such a way that these machines are utilized to the fullest possible extent.

The importance of proper production scheduling comes to light in this scenario when manufacturers need to satisfy customer demands with the help of a minimal number of photolithography tools without missing any committed completion time. This committed completion time is known as the due date [7]. Most manufacturing industries are exposed to various challenges like the arrival of high priority jobs, unforeseen breakdowns, scheduled maintenance, delayed processing of jobs, and meeting deadlines set by customers. Proper production planning and the development of process scheduling help to maintain or improve the efficiency of systems and control of operations [7]. Different types of scheduling rules, such as static and dynamic rules, are explained and reviewed in [8]. Static and dynamic rules depend on the time when the rule is applied. Static rules are applied at the beginning of the scheduling period and therefore have a fixed schedule whereas dynamic rules change over time. Various scheduling rules are briefly reviewed and their performance measures are compared for different environments. Even though the advantages and disadvantages of each type of rules are not

explained in detail, the authors conclude that the evaluation of performance depends on the objective under consideration [8].

The types of machine environments like single machine, parallel machines, flowshops, job shops, flexible flowshops and flexible job shops found in industries are well explained in many literatures and textbooks [7]. The various solution approaches for the FFS problems are discussed in [9], which includes exact methods, heuristics and meta-heuristics. In exact methods approaches such as branch and bound algorithms are given for solving problems to optimality. The problem with branch and bound algorithms is that they utilize a high amount of computer processing resources and are able to solve only problems with a few jobs and stages. Often, they are also deemed to be too complex for real world problems.

Solving FFS problems by heuristic methods like dispatching rules and variants of shifting the bottleneck procedure (SBP) [10] are explained in [4]. Dispatching rules include certain rules of thumb for the priority assignment of jobs onto machines. Some examples of dispatching rules include Shortest Processing Time (SPT), Longest Processing Time (LPT), and Shortest Remaining Processing Time (SRPT). The SBP uses a divide and conquer strategy and has been proven very effective when used in combination with exact methods for solving problems. The scheduling of a flexible flowshop with cluster tools is performed via simulated annealing [3] to obtain a near-optimal solution. However, the study does not consider the re-entry of jobs to previous stages.

Many mixed-integer programming (MIP) models for scheduling FFS are explained in [11]. The book considers various scenarios of flowshop modeling with multiple machines in each stage and finite or infinite buffers between each stage. Even though most of the papers reviewed have mentioned either the scheduling of flowshops, the scheduling of flexible flowshops and/or scheduling of cluster tools separately, there exist no efficient models that analyze a flexible

flowshop that contains cluster tools and reentrant job flow across multiple product types. Using this as the basis for my thesis research, I will also consider job ready times and the continuous flow of jobs inside cluster tools.

3. PROBLEM DESCRIPTION

The photolithography FFS system is arranged in such a way that the individual machines at each stage are organized as a general FFS with a few sets of cluster tools included. As jobs routed through the various stages of photolithography process could return to one or more of these stages during their processing path, photolithography is a re-entrant flexible flowshop [12]. The multistep "photo" process is now described in detail.

In the first step of the photolithography process, a semiconductor wafer may be cleansed in a sink (tool "S" in Figure 1) [13]. The wafer is coated (tool "C" in Figure 1) with photosensitive resist and is exposed ("E" in Figure 1) to light. Wafers are exposed to light with the help of a pattern mask that controls the wafer areas that receive light exposure. This helps to define the required circuit functionality. The exposed wafer is then developed ("D" in Figure 1) so that the required patterns are imprinted on to the wafer by removing the exposed photoresist. The final photolithography stage is baking ("B" in Figure 1). Sometimes, wafers are baked before and/or after the develop stage. The flow diagram of the photolithography process for a single layer of wafer fabrication is illustrated in Figure 1. Normally, a wafer repeats this process 20-30 times during its process flow. Figure 1 also depicts cluster tools that are used in the photolithography process. Cluster tool "CEDB" processes the coat, expose, develop and bake steps in order. Similarly cluster tools "CED", "CE" and "ED" process Coat-Expose-Develop (CED), Coat-Expose (CE), and Expose-Develop (ED), respectively.

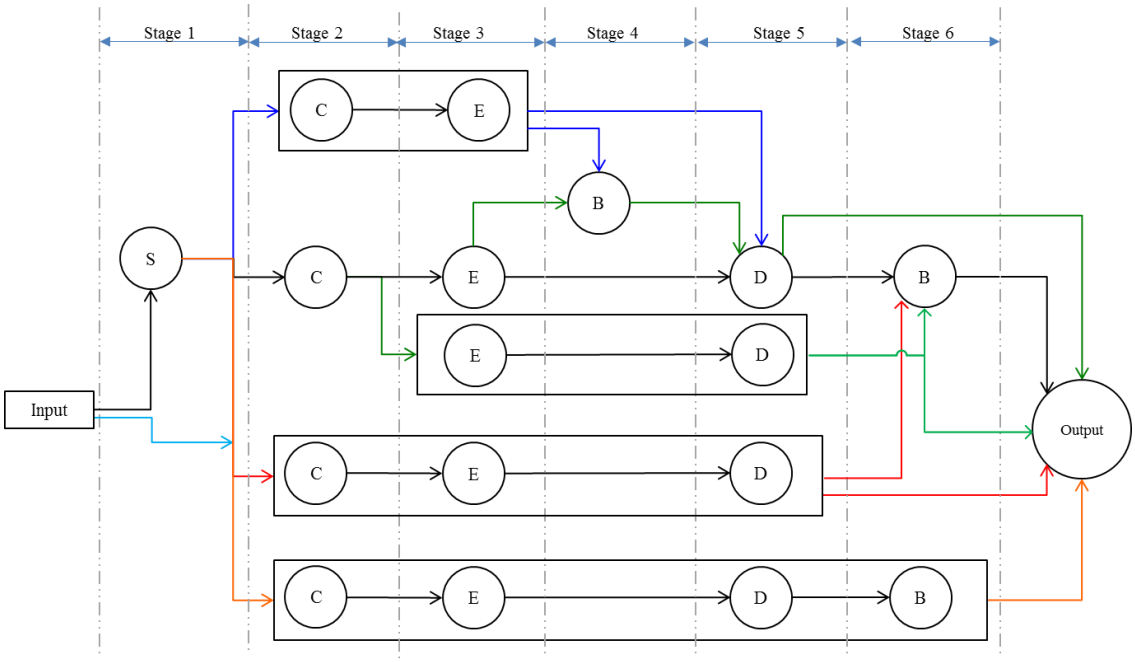

Figure 1: Schematic Diagram for the Photolithography Process

An initial step in optimizing industrial processes often includes improving the total execution time of machines, also known as makespan (C_{max}). A schedule that minimizes makespan can be obtained by applying mixed-integer programming (MIP) techniques. A simple, two-stage flexible flowshop is strongly NP-hard [14]. Hence by extension, the complexity of scheduling a larger flexible flowshop with multiple machines in almost every stages of its processing is also strongly NP hard. When compared to traditional flowshops, a photolithography system involving cluster tools, constraints for multiple wafer routes, re-entrant flow and no buffers inside the cluster tool are therefore also strongly NP hard [3].

The set of jobs entering the system for processing can be characterized by their ready times (r_j), the time at which the job is released to the shop by some external job scheduler [1]. In this thesis, a MIP formulation is developed to optimize the makespan of the photolithography

process containing reentrant jobs with ready times. The MIP formulation for scheduling flowshops with parallel machines and infinite in-process buffers [11] is used as the base model for this FFS and additional constraints are added to expand this formulation to incorporate the scheduling of cluster tools and reentrant flow. In terms of the standard α|β|γ scheduling notation introduced in [15] the problem under consideration is defined as FFm | r_j, rcrc | Cmax [7].

Other objectives that could be optimized in a scheduling system include total weighted completion time (WCT) and total weighted tardiness (TWT). The total weighted completion time is represented as $\sum w_j C_j$ with w_j denoting the weight or priority of job j and C_j representing the completion time of job j. In practice, total WCT is a surrogate measure of the inventory or holding cost incurred by the schedule [7]. The total weighted tardiness, $\sum w_j T_j$, where T_j is the tardiness of the job j, is generally an objective that relates to on-time delivery.

4. MATHEMATICAL MODEL

This section explains a proposed model formulation along with the various notation used in the model, followed by an explanation of the models constraint sets.

4.1. Notation

The notation that is used in the formulation for the flexible flowshop scheduling problem is explained in Table 1.

Table 1: Notation for flexible flowshop scheduling model

Sets	
I	set of processing stages indexed by $i \in I = \{1,..,m\}$
J_i	set of processors in each stage i indexed by $j \in J_i = \{1,..,m_i\}$
K	set of jobs that needs to be processed indexed by $k \in K = \{1,..,n\}$
$CEDB$	set of cluster machines for C-E-D-B indexed by i_1
CED	set of cluster machines for C-E-D indexed by i_2
CE	set of cluster machines for C-E indexed by i_3
ED	set of cluster machines for E-D indexed by i_4
Parameters	
m	number of processing stages
m_i	number of machines at each stage i
n	number of jobs
p_{ik}	processing time for job k in stage i
r_k	ready time for job k
w_k	priority of job k
M	sum of the processing time of all jobs in the system
Decision Variables	

C_{ik}	completion time of job k at stage i
$C_{\max}$	makespan or time at which all jobs complete their operations on all stages
x_{ijk}	1, if job k is assigned to machine j in stage i
	0, Otherwise
y_{kl}	1, if job k precedes job l in the processing sequence
	0, Otherwise

4.2. Model Formulation

The MIP formulation for the proposed model is depicted as below and the functions of each constraint sets are explained after the formulation

$$\text{minimize } c_{\max} \tag{1}$$

subject to

$$c_{1k} \geq p_{1k} + r_k \qquad \forall\, k \in K \tag{2}$$

$$c_{ik} - c_{(i-1)k} \geq p_{ik} \qquad \forall\, k \in K, i \in I, i > 1 \tag{3}$$

$$c_{ik} + M(2 + y_{kl} - x_{ijk} - x_{ijl}) \geq c_{il} + p_{ik} \qquad \forall\, i \in I, j \in J_i, k \in K, l \in K, l > k \tag{4}$$

$$c_{il} + M(3 - y_{kl} - x_{ijk} - x_{ijl}) \geq c_{ik} + pe_{il} \qquad \forall\, i \in I, j \in J_i, k \in K, l \in K, l > k \tag{5}$$

$$c_{mk} \leq c_{\max} \qquad \forall\, k \in K \tag{6}$$

$$\sum_{j \in J_i} x_{ijk} = 1 \qquad \forall\, i \in I, k \in K, \mathrm{p}_{ik} > 0 \tag{7}$$

$$\sum_{j \in J_i} x_{ijk} = 0 \qquad \forall\, i \in I, k \in K, \mathrm{p}_{ik} = 0 \tag{8}$$

$$\sum_{i \in I} x_{ii_1k} = 3x_{2i_1k} \qquad \forall\, i_1 \in \text{CEDB}, k \in K, \quad i_1 \neq 0,\ 3 \leq i \leq \mathrm{m},\ i \neq 4 \tag{9}$$

$$x_{4jk} \le 1 - x_{2i_1k} \qquad \forall\ i_1 \in CEDB \cup \text{CED},\ i_1 \ne 0, j \in J_4, k \in K \tag{10}$$

$$c_{2k} + M(2 + y_{kl} - x_{2i_1k} - x_{2i_1l}) \ge c_{ml} + p_{2k} \qquad \forall\ i_1 \in \text{CEDB}, i_1 \ne 0, k \in K,\ l \in K, l > k \tag{11}$$

$$c_{2l} + M(3 - y_{kl} - x_{2i_1k} - x_{2i_1l}) \ge c_{(m-1)k} + p_{2l} \qquad \forall\ i_1 \in \text{CEDB}, i_1 \ne 0, k \in K,\ l \in K, l > k \tag{12}$$

$$\sum_{i \in I} x_{ii_2k} = 2x_{2i_2k} \qquad \forall\ i_2 \in \text{CED}, k \in K,\ i_2 \ne 0,\ 3 \le i \le \text{m-1},\ i \ne 4 \tag{13}$$

$$c_{2k} + M(2 + y_{kl} - x_{2i_2k} - x_{2i_2l}) \ge c_{(m-1)l} + p_{2k} \qquad \forall\ i_2 \in \text{CED}, i_2 \ne 0, k \in K,\ l \in K, \text{l} > k \tag{14}$$

$$c_{2l} + M(3 - y_{kl} - x_{2i_2k} - x_{2i_2l}) \ge c_{(m-1)k} + p_{2l} \qquad \forall\ i_2 \in \text{CED}, i_2 \ne 0, k \in K,\ l \in K, l > k \tag{15}$$

$$x_{3i_3k} = x_{2i_3k} \qquad \forall\ i_3 \in \text{CE}, k \in K, i_3 \ne 0 \tag{16}$$

$$c_{2k} + M(2 + y_{kl} - x_{2i_3k} - x_{2i_3l}) \ge c_{3l} + p_{2k} \qquad \forall\ i_3 \in \text{CE}, i_3 \ne 0, k \in K,\ l \in K, l > k \tag{17}$$

$$c_{2l} + M(3 - y_{kl} - x_{2i_3k} - x_{2i_3l}) \ge c_{3k} + p_{2l} \qquad \forall\ i_3 \in \text{CE}, i_3 \ne 0, k \in K,\ l \in K, l > k \tag{18}$$

$$x_{5i_4k} = x_{3i_4k} \qquad \forall\ i_4 \in \text{ED}, k \in K,\ i_4 \ne 0 \tag{19}$$

$$c_{3k} + M(2 + y_{kl} - x_{3i_4k} - x_{3i_4l}) \ge c_{5l} + p_{3k} \qquad \forall\ i_4 \in \text{ED}, i_4 \ne 0, k \in K,\ l \in K, l > k \tag{20}$$

$$c_{3l} + M(3 - y_{kl} - x_{3i_4k} - x_{3i_4l}) \geq c_{5k} + p_{3l} \qquad \forall\, i_4 \in \text{ED}, i_4 \neq 0, k \in K, l \in K, l > k \tag{21}$$

$$c_{4k} + M(2 + y_{kl} - x_{4ik} - x_{6il}) \geq c_{6l} + p_{4k} \qquad \forall\, i \in J_4 \cap J_6, i \neq 0, k \in K, l \in K, l > k \tag{22}$$

$$c_{6l} + M(3 - y_{kl} - x_{4ik} - x_{6il}) \geq c_{4k} + p_{6l} \qquad \forall\, i \in J_4 \cap J_6, i \neq 0, k \in K, l \in K, l > k \tag{23}$$

$$c_{6k} + M(2 + y_{kl} - x_{6ik} - x_{4il}) \geq c_{4l} + p_{6k} \qquad \forall\, i \in J_4 \cap J_6, i \neq 0, k \in K, l \in K, l > k \tag{24}$$

$$c_{4l} + M(3 - y_{kl} - x_{6ik} - x_{4il}) \geq c_{6k} + p_{4l} \qquad \forall\, i \in J_4 \cap J_6, i \neq 0, k \in K, l \in K, l > k \tag{25}$$

$$c_{\max} \geq 0 \tag{26}$$

$$c_{ik} \geq 0 \qquad \forall\, i \in I, k \in K \tag{27}$$

$$x_{ijk} \geq 0 \qquad \forall\, i \in I, j \in J_i, k \in K \tag{28}$$

$$y_{kl} \geq 0 \qquad \forall\, k \in K, l \in K \tag{29}$$

The model's objective function (1) minimizes C_{max}. The objective could be changed to minimizing the weighted completion time (WCT) if desired. Constraint sets (2) and (3) ensure that a job starts processing at stage 1 and processes successively on all downstream machines. The overlapping of more than one job on a single machine at a time is prevented by constraint sets (4) and (5). These constraints act as "either-or" constraints, which imply that one of the constraints will be active for a particular value of y_{kl}. When job k precedes job l, then constraint

set (4) will be active and constraint set (5) will be inactive, because of the value of "M" and vice-versa. The value of M is assigned as the sum of processing time of all jobs in the system. The maximum completion time, C_{max}, should be greater than or equal to the completion time of last job in the final stage of the processing. This is achieved by including the constraint sets (6). Constraint sets (7) and (8) ensure that if the processing time for a job in any stage is a non-zero, positive number, then one of the machines in that stage must process the job [16]. If the processing time for a job at any stage is zero, then that stage is skipped for that particular job.

The remaining sets of constraints are developed for cluster tools and re-entrant flow processes. Constraints sets (9) - (12) model the cluster process of coat, expose, develop and bake in a single machine. Constraints sets (9), (13), (16) and (19), ensure that a job that enters a cluster machine will stay inside that machine until it completes all the processes performed by the cluster tool. The constraint set pairs (11) - (12), (14) - (15), (17) - (18), and (20) - (21) stop jobs from entering the cluster tool if the machine is already processing some other job. The constraint set (10) sets x_{ijk}, the assignment variable for job k, for the first bake stage to 0, if job k does not require the baking. Constraint sets (22) - (25) model the re-entry of jobs in the bake process of photolithography. The photolithography process under consideration has re-entry at stages 4 and 6. Hence, the machines of fourth and sixth stage J_4 and J_6 are referenced in these equations. These constraints guarantee that if any of the jobs is being processed in the bake oven at any of its two stages, i.e. the first bake process or the re-entrant second baking stage, then no other job will enter the machine. Finally, constraint sets (26) - (29) are non-negativity constraints, which imply that these variables should have a value greater than or equal to zero.

5. MODEL VALIDATION

A sample problem is created to test the model formulation. After implementing the model in AMPL, a 20-job instance with the properties as described in Table 3 is evaluated for minimizing the makespan. The ready time for each job is randomly generated. The ready times are assigned for the job during its initial processing stage. The random generation of the ready time is obtained using MS Excel with 30% of the jobs in each group have 0 ready times while the remaining 70% of jobs have a randomly assigned ready times between 1 and 2/3 C_{max}. The processing time for each stage is assigned a common discrete value for each of the stages. The number of standalone and cluster tools used for processing the 20 jobs is provided in Table 2 (15 individual machines and 3 cluster machines).

Table 2: Input data for example problem

Machine	**Count**
Sink	4
Coat	2
Expose	4
Develop	2
Bake	3
CEDB	1
CED	2

Table 3: Input data for example problem

Job #	**Ready Time** (r_j)	**Processing Time** (P_{ij})					
		Stage 1	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**	**Stage 6**
1	19	40	20	75	0	30	45
2	0	40	20	75	45	30	45
3	0	40	20	75	45	30	45
4	0	0	20	75	45	30	45
5	108	40	20	75	0	30	0
6	160	40	20	75	45	30	45
7	16	40	20	75	45	30	45
8	0	40	20	75	0	30	0
9	103	40	20	75	45	30	45
10	0	0	20	75	45	30	45
11	159	40	20	75	0	30	0
12	0	40	20	75	45	30	45
13	0	40	20	75	0	30	0
14	0	0	20	75	45	30	45
15	13	0	20	75	45	30	45
16	197	0	20	75	45	30	45
17	94	40	20	75	0	30	0
18	50	0	20	75	45	30	45
19	0	40	20	75	0	30	0
20	0	0	20	75	0	30	0

The solution was produced using Gurobi 5.1.0 solver within a 7200 CPU seconds time limit on a Windows 7 platform with Intel® Core™2 Quad CPU Q6600 @2.40 GHz with 16GB of RAM. Although Gurobi 5.1.0 did not converge to the optimal solution within the allowed 7200 seconds, Gurobi 5.1.0 was able to obtain a good solution quickly. The resulting schedule has C_{max}= 485 and the details scheduled is presented in a Gantt chart in Figure 2 and in Table 4.

Table 4 is sorted based on ascending job number with each job being listed in ascending order of its start time. It can be observed that each job that utilizes the cluster tool leaves the cluster tool only after the processing in all of the tool's stages is complete. In addition to this example problem instance, additional example problems confirmed the accuracy of the proposed formulation's constraints and objectives when compared to manual calculations.

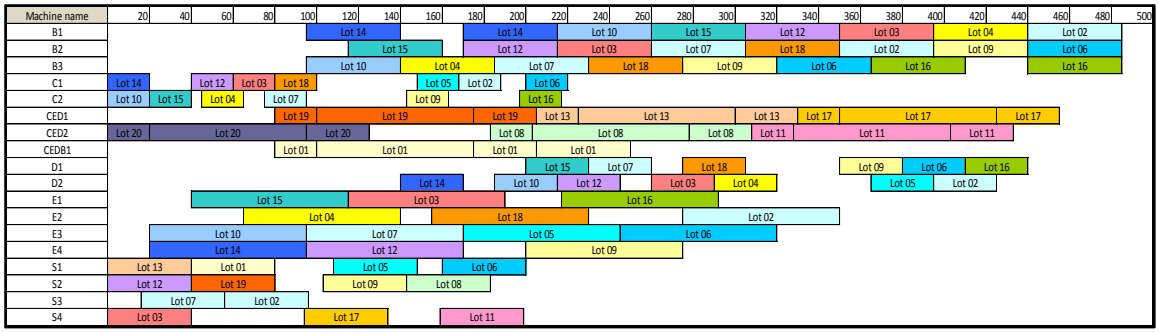

Figure 2: Gantt chart for the sample problem

Table 4: Resulting schedule for the sample model

Job	ST	CT	M/C
1	40	80	S1
1	80	100	CEDB1
1	100	175	CEDB1
1	175	205	CEDB1
1	205	250	CEDB1
2	56	96	S3
2	168	188	C1
2	275	350	E2
2	350	395	B2
2	395	425	D2
2	440	485	B1
3	0	40	S4
3	60	80	C1
3	115	190	E1
3	215	260	B2
3	260	290	D2
3	350	395	B1
4	45	65	C2
4	65	140	E2
4	140	185	B3
4	290	320	D2
4	395	440	B1
5	108	148	S1
5	148	168	C1
5	170	245	E3
5	365	395	D2
6	160	200	S1
6	200	220	C1
6	245	320	E3
6	320	365	B3
6	380	410	D1
6	440	485	B2

Job	ST	CT	M/C
7	16	56	S3
7	75	95	C2
7	95	170	E3
7	185	230	B3
7	230	260	D1
7	260	305	B2
8	143	183	S2
8	183	203	CED2
8	203	278	CED2
8	278	308	CED2
9	103	143	S2
9	143	163	C2
9	200	275	E4
9	275	320	B3
9	350	380	D1
9	395	440	B2
10	0	20	C2
10	20	95	E3
10	95	140	B3
10	185	215	D2
10	215	260	B1
11	159	199	S4
11	308	328	CED2
11	328	403	CED2
11	403	433	CED2
12	0	40	S2
12	40	60	C1
12	95	170	E4
12	170	215	B2
12	215	245	D2
12	305	350	B1
13	0	40	S1
13	205	225	CED1
13	225	300	CED1
13	300	330	CED1

Job	ST	CT	M/C
14	0	20	C1
14	20	95	E4
14	95	140	B1
14	140	170	D2
14	170	215	B1
15	20	40	C2
15	40	115	E1
15	115	160	B2
15	200	230	D1
15	260	305	B1
16	197	217	C2
16	217	292	E1
16	365	410	B3
16	410	440	D1
16	440	485	B3
17	94	134	S4
17	330	350	CED1
17	350	425	CED1
17	425	455	CED1
18	80	100	C1
18	155	230	E2
18	230	275	B3
18	275	305	D1
18	305	350	B2
19	40	80	S2
19	80	100	CED1
19	100	175	CED1
19	175	205	CED1
20	0	20	CED2
20	20	95	CED2
20	95	125	CED2

6. EXPERIMENTAL STUDY

6.1. Experimental Plan

Our experimental plan evaluates three objective functions with the proposed mixed-integer program for photolithography scheduling: minimizing makespan (Cmax), minimizing total weighted completion time (TWC) and minimizing total weighted tardiness (TWT). The experimental design used in our random problem generation [17] is given in Table 5. We investigate three different levels of the number of jobs to be scheduled: 5, 10, and 25. Each set of jobs tested with two levels of ready times. For the first condition, all jobs have zero ready time and for the second condition, some portion of the jobs have a ready time that is a non-zero, randomly generated value while the remaining jobs of the same sets have zero ready time.

The due date value is generated using a discrete uniform distribution (Table 5). The calculation of the estimated makespan includes the total number of jobs, the processing time of the photolithography process's bottleneck stage, the total number of machines that process the bottleneck stage, and the sum of the processing times for all non-bottleneck stages in photolithography. The parameter T is the expected percentage of tardy jobs and we consider two cases for the value of T in our experimentation: 0.3 and 0.6. Further, R is a range parameter that we study at two levels: 0.5 and 2.5 [17].

The weights (w_j) are calculated based on a random distribution of all integers between 1 and 5, 1 being a low priority job and 5 being a high priority job. Considering the three levels for the number of jobs, two scenarios for job ready times, and four combinations of due date parameters T and R, 24 unique combinations of data are run for two different resource (machine) levels (Table 6). As we investigate three different objective functions, a total of 24(2)(3)=144 unique scenarios exist for investigation. Based on 10 replications for each unique scenario, a total

of 1,440 files are generated for analysis and comparison. Microsoft Excel 2010 is used to generate random numbers for the various cases in our experimental plan.

Table 5: Experimental Design

Experimental Factor		**Levels**
Number of jobs, n	5	15 25
Ready time	$r_j = 0$ for all j	30% of jobs, $r_j = 0$ 70% of jobs, r_j = RANDOM[1, 2/3 × makespan]
Job due date d_{ij}	d_{ij}	= Uniform [$\mu(1-0.5R)$, $\mu(1+0.5R)$]
	with	
	μ	= makespan × (1-T)
	makespan	= 1.5 × (n × ((P_{BN} / m_{iBN}) + P_{NBN}
	P_{BN}	= Processing time of the bottleneck stage
	m_{iBN}	= Number of machines that processes the bottleneck stage
	P_{NBN}	= Sum of the Processing time of all other non-bottleneck stages
	T	= 0.3 and 0.6
	R	= 0.5 and 2.5
Processing time	Stage 1	80% of jobs $P_{1j} = 40$ 20% of jobs $P_{1j} = 0$
	Stage 2	100% of jobs $P_{2j} = 20$
	Stage 3	100% of jobs $P_{3j} = 75$
	Stage 4	20% of jobs $P_{4j} = 45$ 80% of jobs $P_{4j} = 0$
	Stage 5	100% of jobs $P_{5j} = 30$
	Stage 6	50% of jobs $P_{6j} = 45$ 50% of jobs $P_{6j} = 0$
Weight, w_j	RANDOM[1,5]	

Table 6: Machine Counts for each experimental scenario

Machine Type	**Scenario 1**	**Scenario 2**
Sink	4	2
Coat	2	1
Expose	4	2
Develop	2	1
Bake	3	2
CE	2	1
CED	2	1
CEDB	2	1
ED	1	1

Since the scheduling problem under study is strongly NP-Hard, heuristic and/or metaheuristic approaches may provide good, near optimal solutions that are better the solution obtained by a time-limited MIP [18]. Various heuristics such as cyclic heuristics [19] or genetic algorithms (GAs) could be employed to achieve high quality solutions. We now present a constructive heuristic approach for the problem under study, and then compare its performance with that of the proposed MIP model under a time limit restriction.

6.2. A Constructive Heuristic

A review of the available literature confirmed that no heuristic is currently available for analyzing the flexible flowshop scheduling problem with cluster tools and job ready times. Therefore, we created our own constructive heuristic (CH) in order to obtain good solutions to the research problem under study very quickly.

Procedure CH

1. Sort jobs to be scheduled in ascending order of ready time.
2. In the event that jobs have the same ready times, break ties by arranging the tied jobs in ascending order of the ratio of due date to weight (d_j/w_j).
3. If a tie still exists, break ties by arranging the tied jobs in descending order of the number of cluster tools that they are eligible to pass through.
4. Jobs are dispatched according to machine availability at each stage of the process. This is done on a rolling time horizon basis until all processes are completed. The triggering event is a job-processing resource (e.g., a single tool or a cluster tool) becoming available (idle) for processing while there exists at least one job that has yet to complete its required processing.
    a. At a given stage, if a job is eligible to process on a cluster tool, it is assigned to that cluster tool only if the tool's first module (e.g., coat) is available <u>and</u> the job is ready to be processed on that stage and module.
    b. In the case of multiple available tools, the decision of which tool processes the job is based on the number of photolithography stages the tool could process. For example, a cluster tool with four modules of processing would be selected for job processing over a two module cluster tool or a single tool if the cluster tool is available and the job is eligible to pass through it.
    c. Time is elapsed according to the next earliest time a machine or module or job becomes available for subsequent processing.
5. The maximum of the completion time at stage 5 and 6 is the makespan for all problems. Total weighted tardiness and total weighted completion time is calculated once the completion time for each machine at the jobs final stage is obtained.

Procedure CH was implemented using Visual Basics for Application (VBA) in Microsoft Excel 2010. Procedure CH also analyses the same data that was generated for the mathematical model. Typically, Procedure CH required less than two seconds per problem instance to obtain a feasible solution.

6.3. Experimental Results and Analysis

We compare the performance of Procedure CH and the proposed mathematical model by computing a performance ratio. Let performance ratio PR be defined as the ratio of the objective function value obtained by Procedure CH ("TWT_{CH}") for a problem instance to the optimal objective function value produced by the mathematical model ("TWT_{opt}") for the same problem instance [20]. While the PR ratio can be computed for any objective function case of interest, it is only valid when the MIP model produces an optimal solution. In this way, we obtain an estimate of the quality of the Procedure CH solution in terms of its percent above the optimal solution value.

Once the results are obtained for each instance, the PR values can be averaged across all experimental instances for a given type of problem type (e.g., all instances with five jobs). We can characterize any set of like problem instance in terms of (n, r_j, T, R, mc). In this expression, n is the number of jobs and $r_j = 0$ denotes all 0 job ready times while $r_j = 1$ denotes the presence of non-zero ready times. Further, T and R are the due date-related parameters described above and mc represents the machine configuration (scenario 1 or scenario 2). An example for this instance characterization approach is (15, 0, 0.3, 2.5, 1), which represents the *average PR values* for problems with 15 jobs that have zero job ready times, T and R values of 0.3 and 2.5, respectively, and machine configuration of Scenario 1 from Table 6.

The average performance ratios for each experimental factor level are shown in Table 7 for each objective function. The number of instances that were solved optimally are given in parentheses after the average PR values. For the makespan objective function, the average performance of Procedure CH is 7% above optimal. From the result we could see that the performance ratio for the instances with lesser number of jobs provided a better PR, while the overall performance ratio was affected by a few scenarios that produced worse results. For example, for one case of (15,0,*,*,*), the PR for WCT was 45.84, which adversely contributes to the averages.

Table 7: Average performance ratios for each experimental factor level

	C_{max}	TWT	WCT
Overall PR	1.07 (329)	1.87 (390)	1.21 (220)
(5,*,*,*,*)	1.05 (160)	1.13 (160)	1.01 (160)
(5,0,*,*,*)	1.09 (80)	1.21 (80)	1.01 (80)
(5,1,*,*,*)	1.01 (80)	1.07 (80)	1.00 (80)
(15,*,*,*,*)	1.09 (91)	3.13 (124)	1.88 (52)
(15,0,*,*,*)	1.45 (12)	7.33 (44)	45.84 (1)
(15,1,*,*,*)	1.03 (79)	1.77 (80)	1.02 (51)
(25,*,*,*,*)	1.09 (78)	1.58 (106)	1.02 (8)
(25,0,*,*,*)	N/A	1.48 (31)	N/A
(25,1,*,*,*)	1.09 (78)	1.59 (75)	1.02 (8)
(*,0,*,*,*)	1.13 (92)	2.74 (155)	1.56 (81)
(*,1,*,*,*)	1.04 (237)	1.48 (235)	1.01 (139)

We now compute the heuristic ratio (HR) metric, which is defined as the ratio of the objective function value obtained by Procedure CH ("TWT_{CH}") for a problem instance to the non-optimal objective function value produced by the mathematical model in 7200 seconds ("TWT_{7200}") for the same problem instance. The average heuristic ratio for each experimental factor level by objective function is shown in the Table 8. We could see that Procedure CH

produced results that are 36% above the time-limited MIP model solution for the makespan objective function. However, the 36% above optimal performance is obtained in less than two seconds. One other observation that we could obtain from Table 8 is that when the problem instance is small, the mathematical problem solved the instances to optimality for all scenarios and hence a HR is not available for those instances that had five jobs.

Table 8: Average heuristic ratios for each experimental factor level

	C_{max}	TWT	WCT
Overall HR	1.36 (151)	1.68 (90)	1.15 (260)
(15,*,*,*,*)	1.31 (69)	1.54 (36)	1.13 (108)
(15,0,*,*,*)	1.31 (68)	1.54 (36)	1.15 (79)
(15,1,*,*,*)	1.05 (1)	N/A	1.11 (29)
(25,*,*,*,*)	1.40 (82)	1.77 (54)	1.15 (152)
(25,0,*,*,*)	1.40 (80)	1.78 (49)	1.19 (80)
(25,1,*,*,*)	1.32 (2)	1.70 (5)	1.12 (72)
(*,0,*,*,*)	1.36 (148)	1.68 (85)	1.17 (159)
(*,1,*,*,*)	1.23 (3)	1.70 (5)	1.11 (101)

6.4. Conclusions and Future Work

A mixed-integer programming (MIP) formulation for the photolithography process with individual and cluster tool was developed for improved job scheduling. Due to this problem's complexity, a constructive heuristics was developed to analyze 1440 experimental cases. When comparing the two solution approaches, the MIP model provides better results but took a considerable amount of time. The heuristic approach achieved some good results in a very short span of time. Future work includes developing improved heuristic solutions to obtain better results. This could be achieved by an improvement phase in the heuristic or via the development of metaheuristic-based solution methods. The research can be extended to deal with finding a

solution for minimizing other objectives like minimizing the total number of tardy jobs, minimizing maximum lateness, or to extend the research to investigate solutions for multiple objective problems.

REFERENCES

[1] R. W. Conway, W. L. Maxwell and L. W. Miller. *Theory of Scheduling* 2003.

[2] D. Sha, S. Hsu, Z. Che and C. Chen. A dispatching rule for photolithography scheduling with an on-line rework strategy. *Comput. Ind. Eng. 50(3),* pp. 233-247. 2006.

[3] S. J. Yim and D. Y. Lee. Scheduling cluster tools in wafer fabrication using candidate list and simulated annealing. *Journal of Intelligent Manufacturing 10(6),* pp. 531-540. 1999. . DOI: 10.1023/A:1008904604531.

[4] T. E. Lee. A review of scheduling theory and methods for semiconductor manufacturing cluster tools. Presented at Simulation Conference, 2008. WSC 2008. Winter. 2008, .

[5] R. Uzsoy, C. Y. Lee and A. M. V. Louis. A review of production planning and scheduling models in the semiconductor industry part I: System characteristics, performance evaluation and production planning. *IIE Transactions 24(4),* pp. 47-60. 1992.

[6] A. Arisha and P. Young. Intelligent simulation-based lot scheduling of photolithography toolsets in a wafer fabrication facility. Presented at Proceedings of the 36th Conference on Winter Simulation. 2004, Available: http://dl.acm.org/citation.cfm?id=1161734.1162095.

[7] M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems* 2012.

[8] M. Montazeri and L. N. Van Wassenhove. Analysis of scheduling rules for an FMS. *Int J Prod Res 28(4),* pp. 785. 1990. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=5781602&site=ehost-live.

[9] R. Ruiz and J. A. Vázquez-Rodríguez. The hybrid flow shop scheduling problem. *Eur. J. Oper. Res. 205(1),* pp. 1-18. 2010.

[10] J. Cheng, Y. Karuno and H. Kise. A shifting bottleneck approach for a parallel-machine flowshop scheduling problem. *Journal of the Operations Research Society of Japan-Keiei Kagaku 44(2),* pp. 140-156. 2001.

[11] T. Sawik. *Scheduling in Supply Chains using Mixed Integer Programming* 2011.

[12] S. C. Graves, H. C. Meal, D. Stefek and A. H. Zeghmi. Scheduling of re-entrant flow shops. *J. Oper. Manage. 3(4),* pp. 197-207. 1983.

[13] T. C. McGuigan. Modeling the lot selection process in semiconductor photolithography processing. Presented at Proceedings of the 24th Conference on Winter Simulation. 1992, .

[14] J. Hoogeveen, J. K. Lenstra and B. Veltman. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *Eur. J. Oper. Res. 89(1),* pp. 172-175. 1996.

[15] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics 5(2)*, pp. 287-326. 1979.

[16] R. Fourer, D. M. Gay and B. W. Kernighan. A modeling language for mathematical programming. *Management Science* pp. 519-554. 1990.

[17] S. V. Mehta and R. Uzsoy. Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE Transactions 30(2)*, pp. 165-178. 1998.

[18] T. Urlings. *Heuristics and Metaheuristics for Heavily Constrained Hybrid Flowshop Problems* 2011.

[19] M. E. Kurz and R. G. Askin. Comparing scheduling rules for flexible flow lines. *Int J Prod Econ 85(3)*, pp. 371-388. 2003.

[20] V. Erramilli and S. J. Mason. Multiple orders per job compatible batch scheduling. *Electronics Packaging Manufacturing, IEEE Transactions On 29(4)*, pp. 285-296. 2006.