Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SHORT-INTERVAL EXPOSITORY REAL-TIME SCHEDULING OF SEMICONDUCTOR MANUFACTURING WITH MIXED INTEGER PROGRAMMING

Myoungsoo (Andy) Ham
Industrial & Systems Engineering
Liberty University
Lynchburg, VA, USA

Siyoung Choi
R&D
CSPI Inc.
Seoul, SOUTH KOREA

ABSTRACT

Efficiently managing the production speed of multiple competing products in semiconductor manufacturing facilities is extremely important from the line management standpoint. Industries have exploited the real time dispatching (RTD) to cope with the problem for the last decade, but the top tier companies have started looking at modern scheduling techniques based on mathematical modeling. We provide real-time scheduling based on mixed integer programming (MIP) capturing the salient characteristics such as shift production targets, machine dedication, sequence-dependent setups, foup queue time, foup priority, schedule stability, etc. Then the reason of specific sequence of foup schedule is communicated to the floor through a self-expository Gantt-Chart. The computer code is written in ezDFS/OPL which provides an all-in-one environment of data manipulation, optimization model development, solving, post processing, and visualization.

1 INTRODUCTION

MIP based detailed production scheduling-systems have not been successful in semiconductor industries. The complex coding methods based on lengthy lines of C/C++ programming and standard query languages (SQL) seemingly contribute to the failures.

Imagine a very plausible story, which occurs at Industrial Engineering (IE) Department, responsible for production scheduling. The IE developers expanded a sizable effort to model the initial user requirements and finally completed the model development in SQL/C++. In a couple of weeks, the developers received a very different user requirements such as new line-management policies, new queue-time management policies, new hot-lot management policies, integration of advanced process control with scheduler, etc. The production control manager strongly requests his requirements be completed by the weekend. The IE developers soon realized that the very complex coding mechanics of SQL/C++ in an ever-changing dynamic environment of semiconductor manufacturing is not an appropriate.

Then, what are the next alternatives? IE Department typically gives up an MIP based scheduler and went back to a 20 years old method of real-time dispatching (RTD) heuristic method, which has been accused of tunnel vision due to the scope of schedule covering only a limited number of foups and machines at each transaction (Dabbas et al. 2001). Also, Govind et al. (2008) point out that the dispatching serves only as the last minute preflight checks in the complex areas in Intel facilities. Alternatively, the IE Department can limit the scope of the integer-programming model to a volume allocation. In other words, the model does not consider the detailed properties of foups, but considers an abstract on real system.

The following is a small integer-programming model of a real-system. There are three types of sets: product *p*, steps *s*, and machines *m*. The decision variables and parameters are shown below.

- X_{psm} Amount of production allocation (in wafer count) of step s of product p on machine m
- T_{ps}^{+} Amount of overachieve (in wafer count) at step s of product p
- T_{ps}^{-} Amount of underachieve (in wafer count) at step s of product p
- $PROGR_{ps}$ Production already made at step s of product p
- TGT_{ps} Target production quantity for step s of product p
- PT_{psm} Processing time of step s of product p on machine m
- RT_{m} Available minutes of machine m

Now, we build a simple MIP model as follows:

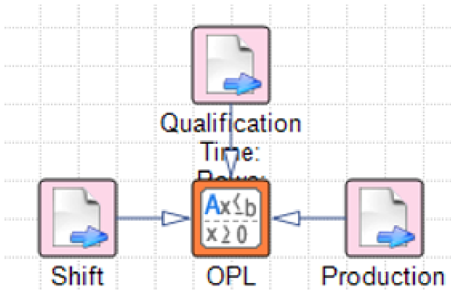
$$Minimize \sum_{p}\sum_{s} \mathrm{w}T_{ps}^{-} - \sum_{p}\sum_{s}\sum_{m} X_{psm} \tag{1}$$

$$\sum_{p}\sum_{s}\left(PT_{psm} \times X_{psm}\right) \leq RT_{m} \qquad \forall m \tag{2}$$

$$\left(\sum_{m} X_{psm}\right) - \mathrm{TGT}_{ps} + \mathrm{PROGR}_{ps} = T_{ps}^{+} - T_{ps}^{-} \qquad \forall p, s \tag{3}$$

Objective (1) tries to minimize the amount of underachieve while maximizing the total throughput. Constraint (2) dictates that the assigned minutes of each machine cannot exceed its available minutes. Constraint (3) calculates that the amount of overachieved and underachieved. In practice, a much more complex model is used, but this simple model still demonstrates the role of MIP model in real industries. Then, this rough-cut of volume allocation (similar to an aggregate planning in a material requirements planning system) is transferred to the middleware which calculates the detailed-schedule (similar to master schedule) at the foups level. Industries use AMAT/RTD, ezDFS, C++, or other platforms as this middleware.

Now the question is how to develop and maintain the complex MIP model in a dynamically changing semiconductor manufacturing without losing the details of real system. We here use the new software called ezDFS/OPL in order to address the industries' main concerns: complexity of coding, slow speed of development, inefficiency of maintenance, and difficulty of data modeling. The software is very similar to typical optimization programming languages such as IBM/OPL, GAMS, and SAS/OR. Figure 1 shows the codes for the above MIP model. Objective (1) is linked to E_1A and E_1B in Figure 1. Similarly, Constraints (2) and (3) are linked to E_2 and E_3.



▪ Set Data

Name	Type	Mode	Value	Column
p	integer	Max	Qualification	P
s	integer	Max	Qualification	S
m	integer	Max	Qualification	M

▪ Parameter Data

Name	Mode	Type	Task	Index	Column
PROGR	Normal	integer	Production	p, s	PROG
TGT	Normal	integer	Production	p, s	TGT
PT	Normal	integer	Qualification	p, s, m	PT
RT	Normal	integer	Shift	m	RT
QUAL	Positive		Qualification	p, s, m	

▪ Decision / Interim

Name	Type	Index	Constraint
X	General	p, s, m	QUAL
TP	General	p, s	
TN	General	p, s	

▪ Equation Add Dele

Name	Category	Index	Expression
E 1A	Objective		SUM((p,s), 999 *TN(p,s))
E 1B	Objective		- SUM((p,s,m), X(p,s,m))
E 2	Constraint	m	SUM((p,s), PT(p,s,m) * X(p,s,m)) <= RT(M)
E 3	Constraint	p, s	SUM(M, X(p,s,m)) + TN(p,s) - TP(p,s) = TGT(p,s) - PROGR(p,s)

Figure 1: Source Code for the Simple Production-Target Model.

2 PROBLEM DESCRIPTION

The small MIP model in the previous section does not cover most of business objectives and constraints. Instead, a practical model must include the salient characteristics such as shift production targets, machine dedication, sequence-dependent setups, foup queue time, foup priority and schedule stability with objectives of minimizing makespan and maximizing throughput.

2.1 Objectives

Multiple objectives are considered simultaneously within a single objective function and ranked in importance as shown in the following example as proposed by Bixby, Burda, and Miller (2006).

Minimize {
+ W_1 * CMAX
– W_2 * Throughput
+ W_3 * Priority Lot Assignment
+ W_4 * Queue Time Violation Penalty
+ W_5 * Change Over Penalty
+ W_6 * Setup Penalty
}

The weight values illustrated by Ws were adjusted to respond to dynamic operational goals. For example, in a foundry business, the priority lot assignment and the queue time penalty may be ranked higher than the other objectives. On the other hand, an integrated device manufacturer (IDM) may rank the throughput and the cycle time objectives higher than other objectives.

2.2 Constraints

- Shift targets – The amount of production target should be fulfilled by the end of each shift. The amount of underachieve is typically penalized.
- Schedule stability – The schedule which was previously generated must be conserved at the next schedule-run in order to minimize the disruption of other coordination works including tool port management, reticle management, delivery, and so on.
- Future foup arrival – Scheduler must include the future incoming foups for the next several hours. We assume the arrival times of foups can be pre-determined.
- Manual foup reservation – There is still a need of reserving foup manually occasionally. The constraint allows manually schedule foups on a specific machine at a specific sequence.
- Foup priority – Industries use a foup prioritization method: tagging foups as priority so that they expedite those foups over normal foups. The maximum tolerable waiting-time can be differentiated depending on the urgency of foups, Ultra Hot, Hot, Priority 1, Priority 2, Priority 3, etc.
- Queue time – Certain foups must start processing before a prescribed expiration time in order to minimize air exposure.
- Sequence dependent setup constraints – There is a changeover loss when one type of step/product is changed to another. This penalty can be found in most of areas such as diffusion, litho, etch, cmp, and implanter.
- Foup delivery constraints – The machines that belong to the same station family can be spread out in multiple bays, which causes long delivery time. Therefore, it is important to minimize deliveries across different bays by scheduling the foups to the closest machines.

3 IMPLEMENTATION

We model a typical semiconductor manufacturing area, which has 10 to 20 machines, 100 to 300 foups, and 10 to 60 products. The time slot is defined as the smallest processing time of machine. The scheduling horizon covers the current shift and the next shift so it can last up to 24 hours.

3.1 Development Tool

To answer for the business need of developing a complex-but-manageable MIP model in an ever-changing dynamic semiconductor-manufacturing environment, we have added an OPL feature to the existing graphical real-time dispatching/scheduling development tool called ezDFS, which is based on the fourth-generation programming language (4GL). This newly designed ezDFS/OPL software can therefore be expressed in a hybrid form of AMAT/RTD for a flexible data manipulation and OPL for a direct translation of the mathematical model into codes.

Figure 2 shows an example of the codes where data manipulation/modeling, linear programming modeling, solving, post processing, and visualization are developed in an all-in-one environment.

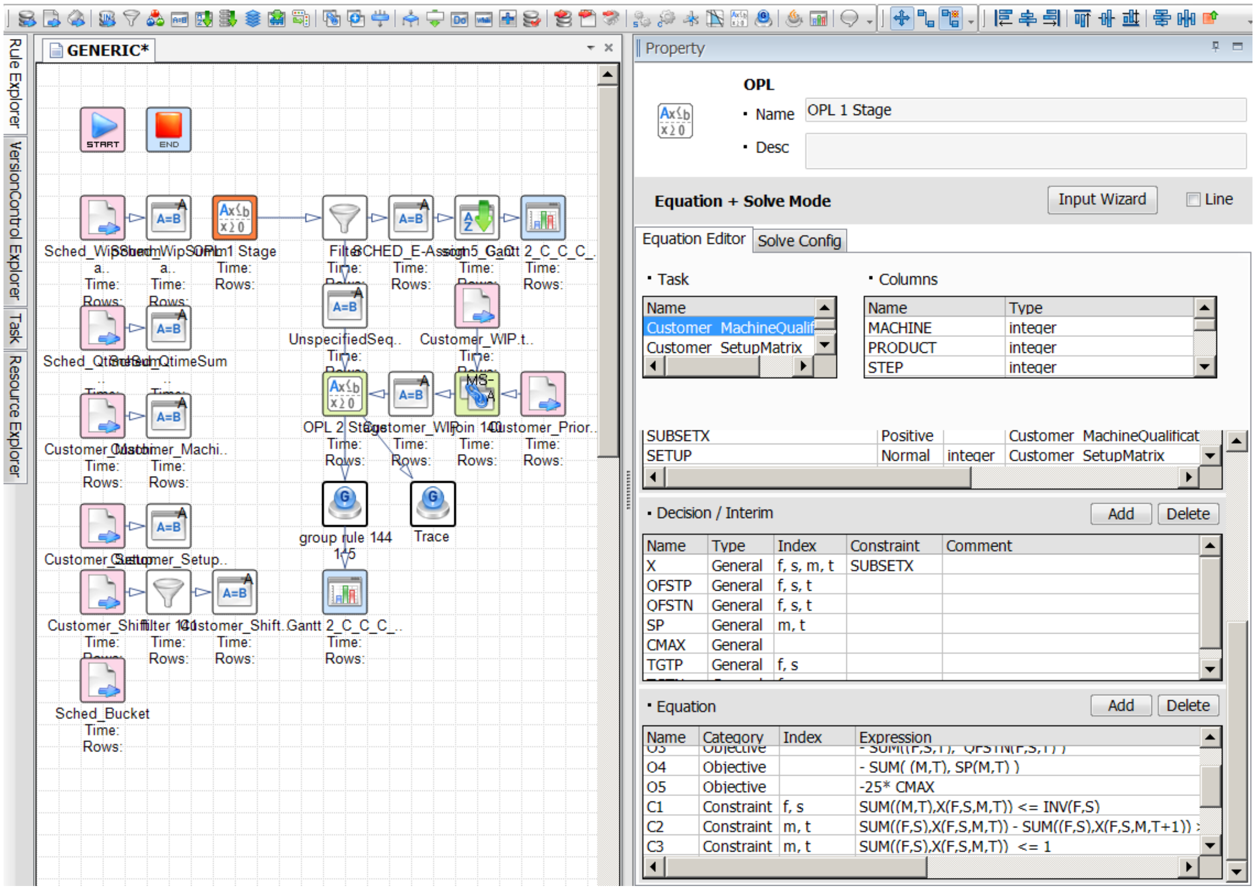


Figure 2: Source Code for the Practical model.

3.2 Self-Expository Gantt-Chart

We add a reason code of foup sequencing decision to the self-expository Gantt-Chart. For instance, foup 58 is scheduled on machine 2 because of its dedication, foup 77 is scheduled earlier due to its high priority, foup 154 is scheduled earlier to meet a production target, and foup 143 is scheduled earlier due to its imminent queue time expiration as shown on Figure 2.

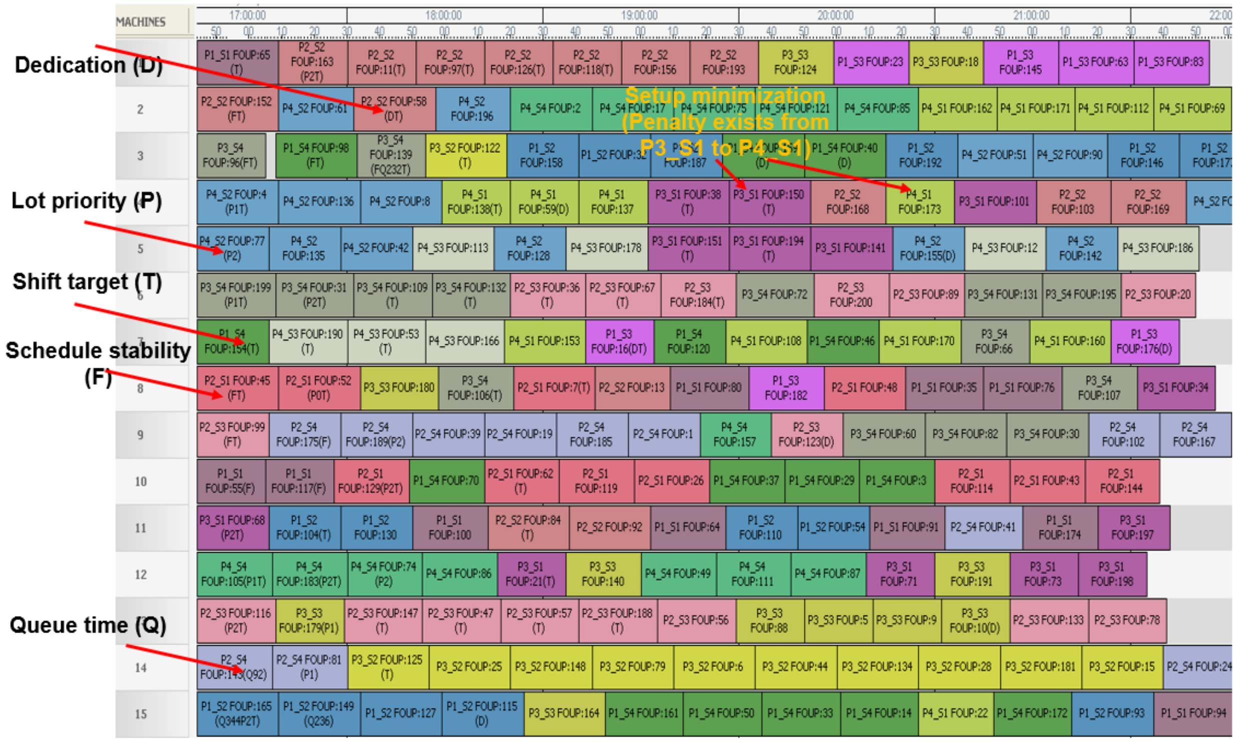


Figure 2: Self-Expository Gantt-Chart.

3.3 Model Performance

The model is tested against a datasets with 10 to 20 machines, 100 to 300 foups, and 10 to 60 products. In order to boost up the engine performance of the scheduler, a two-stage modeling technique was devised. At the first stage, a slot for each foup on a machine is generated without modeling each foup. It simply prescribes an optimal type of product and recipe for each slot by observing all constraints and objectives. At the second stage, the specified sequencing module allocates each foup into the matching slot. By taking this hierarchical modeling approach, we dramatically reduce a run-time while ensuring near-optimality of the solution. We found the optimal solution within a couple of seconds for the most of the datasets as shown on Figure 3.

```
Root node processing (before b&c):
  Real time             =    0.62
Parallel b&c, 2 threads:
  Real time             =    0.00
  Sync time (average)   =    0.00
  Wait time (average)   =    0.00
                          -------
Total (root+branch&cut) =    0.62 sec.
```

Figure 3: Cplex log for the problem set with 15 machines, 200 jobs, and 16 different recipes.

All experimentations were conducted on a normal personal computer with Windows 7, Intel i-7 processor, and 4 GB of memory. The runtime would be even faster in a real production server. We now call this scheduler "real-time scheduler (RTS)" since it truly generates the schedule within a few seconds.

4 CONCLUSION

Efficiently managing the production speed of multiple competing products in semiconductor manufacturing facilities is extremely important from the line management standpoint so industries set the daily production target and try to minimize the amount of underachieve while maximizing total throughput.

To cope with this challenge, we provide the real-time scheduling system based on mixed integer programming (MIP) capturing the salient characteristics such as shift target, machine dedication, sequence-dependent setup, foup queue time, foup priority, schedule stability, etc. However, MIP based detailed production scheduling-systems have not been successful in semiconductor industries due to the complex C++/SQL coding requirement and the difficulty of maintaining and changing the system as goals change.

We implement the MIP model in a new software, which is built upon the foundation of the flexible data manipulation environment found in AMAT/RTD. Then, we added a reason code of foup sequencing decision to the self-expository Gantt-Chart. The engine generates the optimal schedule within a couple of seconds in most of the datasets with 10 to 20 machines, 100 to 300 foups, and 10 to 60 products, owing to the proprietary two-stage modeling technique.

REFERENCES

Dabbas, R., Chen, H., Fowler, J. and Shunk, D. 2001. "A combined dispatching criteria approach to scheduling semiconductor manufacturing systems." *Computers and Industrial Engineering*, vol. 39, pp. 307–324.

Govind, N., Bullock, E., Linling, Iyer, H. B., Krishna, M., and Lockwood, C. S. 2008. "Operations management in automated semiconductor manufacturing with integrated targeting, near real-time scheduling, and dispatching," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21(3), pp. 363–370, 2008.

Bixby, R., R. Burda, and D. Miller. 2006. Short-interval detailed production scheduling in 300mm semiconductor manufacturing using mixed integer and constraint programming, *In Proceedings of the Advanced Semiconductor Manufacturing Conference*, 148–154.

CSPI/ezDFS. 2014 "CSPI/ezDFS". http://www.cspi.co.kr/en_02_01_product_ezdfs.htm.

AMAT/RTD. 2014 "AMAT/RTD" http://www.appliedmaterials.com/global-services/automation-software/apf-rtd-and-reporte.

IBM/OPL. 2014 "IBM/OPL." http://www-01.ibm.com/software/commerce/optimization/modeling/.

GAMS. 2014 "GAMS" http://www.gams.com/.

SAS/OR. 2014. "SAS/OR" http://www.sas.com/en_us/software/analytics/sas-or.html.

AUTHOR BIOGRAPHIES

MYOUNGSOO (ANDY) HAM is program director and associate professor in Industrial & Systems Engineering at Liberty University in Lynchburg, Virginia, USA. He holds an M.S. in Operations Research from the University of Texas at Austin and a Ph.D. in Industrial Engineering from Arizona State University. He worked for Samsung Austin Semiconductor, Samsung Electronics, Texas Instruments, Globalfoundries, and ILOG. His email address is mham@liberty.edu.

SIYOUNG CHOI is senior director of research and development in CSPI. After working for Samsung Electronics in the semiconductor division over 10 years, he joined CSPI as a founding member and created the real-time dispatching/scheduling solution, which is now adapted by many companies including Samsung Electronics, Samsung Display, Hynix, BOE, and Hydis. His email address is fasyc@cspi.co.kr.