http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (6): 918-923, 2008
ISSN 1812-5638
© 2008 Asian Network for Scientific Information

Mathematical Modeling of the IC Final Testing Order Selection Problem

C.Y. Huang
Department of Business Administration, Ching Yun University, No. 229,
Chien Hsin Road, Jung-Li 320, Taiwan, R.O.C.

Abstract: After the depression of the semiconductor industry in 2000 to 2003, IC packaging and IC final testing companies, the back end of the supply chain in semiconductor manufacturing, have changed their capacity expansion strategy from follower strategy to lag strategy, in compared with wafer fabrication. As a result, when the market is in the upturn, the IC final testing practitioner may have insufficient capacity and a selection of candidate orders is necessary. This study thus proposes a Mixed Integer Linear Programming (MILP) model to solve the IC Final Test Order Selection Problem (ICFTOSP). With the objective of maximizing a firm's profit, the model takes into account the characteristics of limited capacity of testers, different profit for different orders and sequence dependent setup time. To increase the applicability of this MILP, a best estimate search strategy incorporating a branch based on pseudo reduced costs rule was adopted to increase the solution efficiency and the numerical result showed that an acceptable solution can be obtained in a reasonable computational time.

Key words: IC final testing, order selection, setup time, mixed integer linear programming

INTRODUCTION

Semiconductor manufacturing process consists of four major process stages: wafer fabrication, wafer probe, IC package and IC final test. The semiconductor manufacturing process is as shown in Fig. 1. Based on the appearance of products, the first two stages, which process on wafers, are referred to as front-end and the last two stages, which process on ICs, are referred to as back-end. Both wafer probe and IC final test are testing activities on products and the differences between the two are that the product appearances and the testing objectives are different. For wafer probing, the testing target is the dies on a wafer and the defective dies are inked so that they will not be further assembled and the manufacturing cost can be reduced as a result. On the other hand, IC final testing is to ensure the functionality of IC after the assembly process. Packaged ICs may be defective due to an improper assembly process, for example, an uneven lead frame that makes wire bonding incorrectly connects it with the die. In addition, final IC products may be used in a variety of environments and IC final test is to ensure that these ICs can meet the predetermined functions and criteria (such as high temperature) so that the required quality of final IC products can be met.

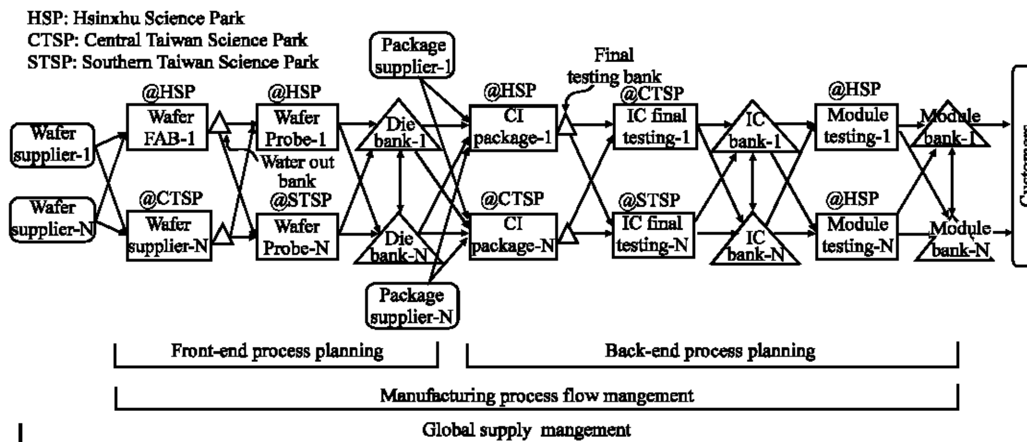

Fig. 1: Semiconductor manufacturing process (Chung *et al*., 2007)

When a tester machine is available, an operator needs to verify the setting of the tester. The procedure of the setup is as follows:

- Obtain the appropriate handler, hi-fix, change kit according the device type and package type of the IC product and bring them to the tester
- Connect the handler, hi-fix and change kit to the tester
- Bring the handler to the required temperature
- Download the required software

In the IC final testing factory that we investigated in this study, they will receive many orders from different customers (i.e., the IC packaging factories). The information of each candidate order includes its device type, package type, lot size, unit testing time and the required test environment of the order. When processing two consecutive orders, a sequence dependent setup time may be incurred due to the different of their setup requirement. If the testing condition is the same one when testing two different orders consecutively on a tester, the operator only needs to download the required program and set the testing environment. However, if the testing condition is different, the operator needs to obtain the required hi-fix and change kit and connect it with the tester first and then the required program needs to be downloaded and the testing environment set. In fact, different orders usually require different testing criteria. Such a production characteristic leads to a production scheduling problem with sequence dependent setup time. That is, with a different processing sequence of orders, the required setup time may be different. An example is as shown in Table 1.

Additionally, the unit profit of each order may not equal since the depreciation of the required tooling (i.e., handler, hi-fix and change kit) and the hourly rate of the tester machine. The tester machines, used in the shop floor to perform testing operations in the IC final testing factory, are the most expensive machine and usually have a highest utilization rate. The production planning mechanism is executed every month to select orders from candidate orders. According to the fundamental concept of the Theory of Constraints (TOC) (Goldratt and Cox, 1992), the performance of a system is determined by the bottleneck resource in that system. Therefore, the aim of the production planning mechanism is deciding which orders should be accepted for maximizing the received profits.

From the open literature review, the authors found that most works on IC final testing factories are related to shop floor production management. Uzsoy *et al.* (1991) applied the decomposition method to divide the production system of a semiconductor test facility into a number of workstations and developed an algorithm to solve the scheduling problem in each workstation for minimizing number of tardy jobs. Ovacik and Uzsoy (1992, 1994, 1996) extending the study of Uzsoy *et al.* (1991), proposed shifting bottleneck algorithm, rolling horizon algorithms and decomposition method to solve such a problem, respectively. Perry and Uzsoy (1993) combined a decomposition method for the static problem with a discrete event driven rescheduling approach to minimize maximum lateness in the testing shop floor. Yoo and Martin-Vega (1997) applied a decomposition method incorporated with shifting bottleneck algorithm for minimizing the number of tardy jobs in semiconductor test facility. Pearn *et al.* (2002a, b) considered a parallel machine version of the scheduling problem of Uzsoy *et al.* (1991). They proposed a Mixed Integer Linear Programming (MILP) formulation and a modified saving algorithm to solve the problem, respectively. Lin *et al.* (2004) developed a Drum-Buffer-Rope (DBR) based dispatching rule to schedule the shop floor for minimizing the setup time and maximizing the committed volume performance. Lin *et al.* (2005) developed a Parameterized Dispatching Rule (PDR) using the response surface method for optimizing the desirability function that is the combination of priority of products, on-time delivery performance, processing time, setup time, waiting time and machine flexibility. Song *et al.* (2007) considered a similar problem that stated in Pearn *et al.* (2002a, b) and proposed an Ant Colony Optimization (ACO) method to minimize the machine conversion time, i.e., machine setup time. To the best knowledge of authors, there is no research that focuses on the profit maximization in the IC final test factory. Therefore, this study tackles an IC Final Test Order Selection Problem (ICFTOSP) under a single bottleneck resource environment. With the consideration of limited capacity of testers, price and delivery of orders and sequence dependence setup time, a Mixed Integer Linear Programming (MILP) model is proposed to achieve the objective of increasing a firm's profit. A demonstrative is also given to illustrate the usability of the proposed model. In the last section, some concluding remarks are made.

Table 1: An example of sequence dependent setup

	To			
From	Idle	A	B	C
Idle	-	20	25	30
A	15	-	30	25
B	25	15	-	25
C	45	20	15	-

NOTATIONS

Parameters

- i : Index of candidate order, $i = 0, 1, 2, \ldots, I$, where 0 is a dummy order to represent the initial status of bottleneck resource (tester) at the beginning of the planning horizon
- n_i : Lot size of order i
- p_i : Unit processing time of order i
- pf_i : Unit profit of order i
- $s_{ii'}$: Setup time for changing setup from processing order i to order i'
- Q : A very large positive number
- C : Available capacity of bottleneck resource in planning horizon

Decision variables

- t_i : Starting time of order i to be processed in planning horizon
- x_i : If order i is selected to process in planning horizon, then $x_i = 1$; otherwise, $x_i = 0$
- $y_{ii'}$: Relative precedence variable of order i and order i'. If order i is processed before order i', then $y_{ii'} = 1$; otherwise, $y_{ii'} = 0$
- $Z_{ii'}$: Direct precedence variable of order i and order i'. If order I is processed directly before order i', then $z_{ii'} = 1$; otherwise, $z_{ii'} = 0$

A MIXED INTEGER LINEAR PROGRAMMING FORMULATION

In an IC final testing factory, tester is the bottleneck resource and the objective function of the ICFTOSP is to maximize the system profit. In implementing the final testing task, the characteristic of sequence dependent setup time is present. The MILP model for the ICFTOSP can be represented by Eq. 1-15.

Objective function: The objective function tends to maximize the profit received in the planning horizon. It is the sum of the profit of the selected orders from the candidate orders.

$$\text{Maximize} \sum_{i=1}^{I} pf_i\, n_i\, x_i \qquad (1)$$

Machine capacity constraints: Constraint (2) ensures that the selected orders to process in planning horizon must not violate the available capacity of bottleneck resource. The total load in the planning horizon is calculated as the summing up the sum of the processing time of the selected orders and the sum of the setup time between orders.

$$\sum_{i=0}^{I} x_i\, n_i\, p_i + \sum_{i=0}^{I}\sum_{i'=0}^{I} z_{ii'}\, s_{ii'} \le C, \quad i \ne i' \qquad (2)$$

Processing precedence constraints: Constraints (3) and (4) ensure that the sequence between two orders must be hold ($y_{ii'} = 1$ or $y_{i'i} = 1$) when order i ($x_{ii'} = 1$) and order i' are selected ($x_i = x_{i'} = 1$) to process in planning horizon. Constraint (5), (6) and (7) describe the relationship between relative precedence variable $y_{ii'}$ and decision variable x_i. If both of order i and i' are not selected in planning horizon ($x_i = x_{i'} = 0$), then $y_{ii'}$ and $y_{i'i}$ must be zero, respectively (constraint (5) is satisfied). If one of order i or i' is selected in planning horizon ($x_i = 1$ or $x_{i'} = 1$), then both of precedence variables $y_{ii'}$ and $y_{i'i}$ must be zero, respectively (the value of $y_{ii'}$ and $y_{i'i}$ is restricted to be zero by either constraint (6) or (7)). Constraint (8) describes the relationship between relative precedence variable and direct precedence variable. Order i could precede order i' directly ($z_{ii'} = 1$) only when precedence variable $y_{ii'} = 1$ and order i could not precede order i' directly ($z_{ii'} = 0$) only when precedence variable $y_{ii'} = 0$. Constraint (9) ensures that when n orders are selected to process in planning horizon, there must be (n-1) direct precedence variables with value of 1. Constraint (10) states the initial status at the beginning of the planning horizon. We note that the initial status is the required test environment of the latest process order in prior planning horizon.

$$(y_{ii'} + y_{i'i}) - Q(x_i + x_{i'} - 2) \ge 1, \quad i \ne i' \qquad (3)$$

$$(y_{ii'} + y_{i'i}) + Q(x_i + x_{i'} - 2) \le 1, \quad i \ne i' \qquad (4)$$

$$(y_{ii'} + y_{i'i}) - Q(x_i + x_{i'}) \le 0, \quad i \ne i' \qquad (5)$$

$$(y_{ii'} + y_{i'i}) - Q(x_{i'} - x_i + 1) \le 0, \quad i \ne i' \qquad (6)$$

$$(y_{ii'} + y_{i'i}) - Q(x_i - x_{i'} + 1) \le 0, \quad i \ne i' \qquad (7)$$

$$y_{ii'} \ge z_{ii'} \quad i \ne i' \qquad (8)$$

$$\sum_{i=0}^{I} x_i - \sum_{i=0}^{I}\sum_{i'=0}^{I} z_{ii'} = 1, \quad i \ne i' \qquad (9)$$

$$\sum_{i=1}^{I} z_{0i} = 1 \qquad (10)$$

Starting and finishing process time constraints: Constraint (11) ensures the starting time of order i must be zero ($t_i = 0$) if order i is not selected ($x_i = 0$). Constraint (12) ensures the starting time of order i' ($t_{i'}$) to be processed must be greater than the finishing time of order i ($t_i + n_i p_i + s_{ii'}$) if order i is scheduled prior to order i' ($y_{ii'} = 1$). Constraint (13) ensures that the starting time of order i' equals to the finishing time of order

i ($t_i + n_i p_i + s_{ii'} = t_{i'}$) if order i and i' are the two consecutive orders ($y_{ii'} = 1$ and $z_{ii'} = 1$). Constraint (14) ensures that the finishing time of order i can not surplus the length of the planning horizon.

$$t_i - Q x_i \le 0 \text{, for all i} \tag{11}$$

$$t_i + n_i p_i + s_{ii'} - t_{i'} + Q(y_{ii'} - 1) \le 0, \quad \text{for all i, } i \ne i' \tag{12}$$

$$t_i + n_i p_i + s_{ii'} - t_{i'} - Q(y_{ii'} + z_{ii'} - 2) \ge 0, \quad \text{for all i, } i \ne i' \tag{13}$$

$$t_i + n_i p_i \le C, \text{ for all i, } i \ne i' \tag{14}$$

Decision variables

$$t_i \ge 0, \text{ for all i} \tag{15}$$

$$x_i \in \{0, 1\}, \text{ for all i} \tag{16}$$

$$y_{ii'} \in \{0, 1\}, \text{ for all i} \tag{17}$$

$$z_{ii'} \in \{0, 1\}, \text{ for all i} \tag{18}$$

For an order selection problem with I orders, the MILP model contains I variables of x_i, I variables of t_i, I(I-1) variables of $y_{ii'}$, I(I-1) variables of $z_{ii'}$. In addition, there are a total of I(I-1) equations in constraint (2), I(I-1)/2 equations in constraint (3), (4), (5), (6) and (7) respectively, I(I-1) equations in constraint (8) and (9) respectively and I equations in constraint (10) and I equations in constraint (11) and I(I-1)/2 equations in constraint (12) and (13), respectively and I equations in constraint (14). Thus, the total number of variables is $2I^2$ and the total number of equations is $(15/2)I^2 - (9/2)I$.

RESULTS AND DISCUSSION

A simple example is presented in this section to explain the solving process and the application of the proposed model. Consider a production environment with one tester machine in bottleneck workstation. In the beginning of planning horizon, there have 15 candidate orders waiting for testing and the information of test type, processing time (min), lot size (die) and profit of each candidate order as shown in Table 2. The capacity limit of bottleneck resource is 120 min. When processing two consecutive orders, there may have setup time for preparing the testing environment in machine and the required setup time between different test types are shown in Table 3.

Table 2: The required test type, process time, lot size and profit for the 15 candidate orders

Order No.	Test type	Process time	Lot size	Profit
1	1*	2	1	12
2	1	3	1	18
3	1	2	1	12
4	2	4	1	28
5	2	3	1	21
6	3	3	1	24
7	3	3	1	24
8	3	4	1	32
9	4	1	1	9
10	4	1	1	9
11	4	1	1	9
12	5	2	1	18
13	5	3	1	27
14	5	2	1	18
15	5	3	1	27

*Is an index to represent the required testing environment condition

Table 3: Setup time required for switching one test type to another in example

	To					
From	0	1	2	3	4	5
0	-	22	22	25	25	22
1	22	-	16	24	25	7
2	22	16	-	25	24	22
3	22	21	22	-	16	22
4	22	22	21	16	-	22
5	22	7	22	25	25	-

Next, the proposed MILP formulation is implemented by using the famous commercial software ILOG OPL Studio 3.5. The ILOG OPL 3.5 adopted the CPLEX 7.5 as the computational core to solve kinds of Linear Programming (LP), integer programming (IP) and Mixed Integer Linear Programming (MILP) model. In order to solve the problem more efficiently, we adopt the best estimate search as the node selection strategy and branch based on pseudo reduced costs as the variable selection strategy. The best estimate search strategy estimates a given node's progress toward integer feasibility relative to its degradation of the objective function. This setting can be useful in cases where there is difficulty in finding feasible solutions or in cases where a proof of optimality is not crucial. The branch based on pseudo reduced costs rule is a computationally less-intensive version of pseudo costs in which pseudo costs are used to derive an estimate about the effect of each proposed branch from duality information. Such an implementation allows us to set various limits on the number of memory nodes and to obtain a better feasible solution within a reasonable computational time. There are a total of 2180 constraints equations and 544 decision variables.

Table 4 shows the feasible solution and solving time under different node limits. When the node limit is set to 1,000, a relatively good solution can be obtained in 2.7810 sec and the total profit is 267. Table 5 is the feasible

Table 4: Objective value and required solving time under different number of nodes limits

No. of nodes limit	Objective value	Solving time (sec)
1000	267	2.7810
5000	267	17.2030
10000	267	37.7500
50000	276	219.9380
100000	276	538.0310

Table 5: MILP solution under the node limit set as 1000

Name	Value	Name	Value	Name	Value	Name	Value
t_{01}	95	y_{0008}	1	y_{0715}	1	y_{1205}	1
t_{02}	92	y_{0010}	1	y_{0801}	1	y_{1213}	1
t_{04}	116	y_{0011}	1	y_{0802}	1	y_{1214}	1
t_{05}	113	y_{0012}	1	y_{0804}	1	y_{1215}	1
t_{06}	32	y_{0013}	1	y_{0805}	1	y_{1301}	1
t_{07}	29	y_{0014}	1	y_{0806}	1	y_{1302}	1
t_{08}	25	y_{0015}	1	y_{0807}	1	y_{1304}	1
t_{10}	51	y_{0104}	1	y_{0810}	1	y_{1305}	1
t_{11}	52	y_{0105}	1	y_{0811}	1	y_{1314}	1
t_{12}	75	y_{0201}	1	y_{0812}	1	y_{1401}	1
t_{13}	80	y_{0204}	1	y_{0813}	1	y_{1402}	1
t_{14}	83	y_{0205}	1	y_{0814}	1	y_{1404}	1
t_{15}	77	y_{0504}	1	y_{0815}	1	y_{1405}	1
x_{00}	1	y_{0601}	1	y_{1001}	1	y_{1501}	1
x_{01}	1	y_{0602}	1	y_{1002}	1	y_{1502}	1
x_{02}	1	y_{0604}	1	y_{1004}	1	y_{1504}	1
x_{04}	1	y_{0605}	1	y_{1005}	1	y_{1505}	1
x_{05}	1	y_{0610}	1	y_{1011}	1	y_{1513}	1
x_{06}	1	y_{0611}	1	y_{1012}	1	y_{1514}	1
x_{07}	1	y_{0612}	1	y_{1013}	1	z_{0008}	1
x_{08}	1	y_{0613}	1	y_{1014}	1	z_{0105}	1
x_{10}	1	y_{0614}	1	y_{1015}	1	z_{0201}	1
x_{11}	1	y_{0615}	1	y_{1101}	1	z_{0504}	1
x_{12}	1	y_{0701}	1	y_{1102}	1	z_{0610}	1
x_{13}	1	y_{0702}	1	y_{1104}	1	z_{0706}	1
x_{14}	1	y_{0704}	1	y_{1105}	1	z_{0807}	1
x_{15}	1	y_{0705}	1	y_{1112}	1	z_{1011}	1
y_{0001}	1	y_{0706}	1	y_{1113}	1	z_{1112}	1
y_{0002}	1	y_{0710}	1	y_{1114}	1	z_{1215}	1
y_{0004}	1	y_{0711}	1	y_{1115}	1	z_{1314}	1
y_{0005}	1	y_{0712}	1	y_{1201}	1	z_{1402}	1
y_{0006}	1	y_{0713}	1	y_{1202}	1	z_{1513}	1
y_{0007}	1	y_{0714}	1	y_{1204}	1		

The objective value and the required solving time, Node limit, integer feasible: Objective = 267, Solution time = 2.7810 sec, Iterations = 24,557, Nodes = 1,000. All other variable are zero

solution under the node limit set as 1000. The selected orders are No. 1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14 and 15 ($x_{01} = x_{02} = x_{04} = x_{05} = x_{06} = x_{07} = x_{08} = x_{10} = x_{11} = x_{12} = x_{13} = x_{14} = x_{15} = 1$) and processing sequences is no. 8, 7, 6, 10, 11, 12, 15, 13, 14, 2, 1, 5 and 4 ($z_{0008} = z_{0807} = z_{0706} = z_{0610} = z_{1011} = z_{1112} = z_{1215} = z_{1513} = z_{1314} = z_{1402} = z_{0201} = z_{0105} = z_{0504} = 1$) and the total profit is 267 and the total computational time for solving the example is 2.7810 sec.

CONCLUSIONS

After the depression of the semiconductor industry in 2000 to 2003, IC packaging and IC final testing companies, the back end of the semiconductor supply chain, have changed their capacity expansion strategy from follower strategy to lag strategy, in compared with wafer fabrication. Compared with the frond-end practitioners in the supply chain of semiconductor manufacturing, the capacity level of IC final testing companies is relatively lower than that of wafer fabrication and an appropriate selection of candidate orders for final testing is essential.

The IC final test order selection problem has the characteristics of different lot size, process time, order profit, sequence dependent setup time and limited capacity. Based on these, this research proposes a MILP model to solve the order selection problem. This study introduces a real-world example to describe the ICFTOSP and uses ILOG OPL 3.5 to construct the model. The results can be a reference for IC final testing companies for order selection.

To increase the applicability of this MILP, a best estimate search strategy incorporating a branch based on pseudo reduced costs rule was adopted to increase the solution efficiency and the numerical result showed that an acceptable solution can be obtained in a reasonable computational time.

ACKNOWLEDGMENTS

The authors would like to thank the National Science Council, Taiwan, R.O.C., for support under contract no. NSC 96-2416-H-231-001-MY2.

REFERENCES

Chung, S.H., I.P. Chung and A.H.I. Lee, 2007. Collaborative production-distribution planning for semiconductor production turnkey service. Lect Notes Comput. Sci., 4705: 860-870.

Goldratt, E.M. and J. Cox, 1992. The Goal: A Process of Ongoing Improvement, 2nd Edn. North River Press, New York, ISBN: 0884270610. .

Lin, J.T., F.K. Wang and W.T. Lee, 2004. Capacity-constrained scheduling for a logic IC final test facility. Int. J. Prod. Res., 42: 79-99.

Lin, J.T., F.K. Wang and P.C. Kuo, 2005. A parameterized-dispatching rule for a Logic IC sort in a wafer fabrication. Prod. Plan. Control, 16: 426-436.

Ovacik, I.M. and R. Uzsoy, 1992. A shifting bottleneck algorithm for scheduling semiconductor testing operations. J. Elect. Manuf., 2: 119-134.

Ovacik, I.M. and R. Uzsoy, 1994. Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup time. Int. J. Prod. Res., 32: 1243-1263.

Ovacik, I.M. and R. Uzsoy, 1996. Decomposition methods for scheduling semiconductor testing facilities. Int. J. Flex Manuf. Syst., 8: 357-388.

Pearn, W.L., S.H. Chung and M.H. Yang, 2002a. A case study on the wafer probing scheduling problem. Prod. Plan. Control, 13: 66-75.

Pearn, W.L., S.H. Chung and M.H. Yang, 2002b. Minimizing the total machine workload for the wafer probing scheduling problem. IIE Trans., 34: 211-220.

Perry, C.N. and R. Uzsoy, 1993. Reactive scheduling of a semiconductor testing facility. Proceeding of 1993. IEEE/CHMT International Electronic Manufacturing Technology Symposium, Oct. 4-6, pp: 191-194.

Song, Y., M.T. Zhang, J. Yi, L. Zhang and L. Zheng, 2007. Bottleneck station scheduling in semiconductor assembly and test manufacturing using ant colony optimization. IEEE T Autom. Sci. Eng., 4: 569-578.

Uzsoy, R., L.A. Martin-Vega, C.Y. Lee and P.A. Leonard, 1991. Production scheduling algorithm for a semiconductor test facility. IEEE. T Semiconduct M., 4: 270-280.

Yoo, W.S. and L.A. Martin-Vega, 1997. A decomposition methodology for scheduling semiconductor test operations for number of tardy job measures. J. Elect. Manuf., 17: 51-61.