Int J Adv Manuf Technol (2006) 29: 746–752
DOI 10.1007/s00170-005-2568-2

ORIGINAL ARTICLE

Muh-Cherng Wu · Wen-Jung Chang · Chie-Wun Chiou

Product-mix decision in a mixed-yield wafer fabrication scenario

Received: 26 August 2004 / Accepted: 3 November 2004 / Published online: 24 March 2006

Abstract The product mix decision problem for semiconductor manufacturing has been extensively studied in literature. However, most of them are based on a high-yield scenario. Yet, in a low-yield manufacturing environment, some research claims that scrap low-yield lots in an early stage may produce more profit. Considering the early scrapping characteristics, this paper aims to solve the product mix decision problem for a mixed-yield scenario, which involves the simultaneous production of high-yield and low-yield products. A nonlinear mathematical program is developed to model the decision problem. Two methods for solving the nonlinear program are proposed. Method 1 converts the nonlinear program into a linear program by setting some variables as parameters. The method provides an optimal solution by exhaustively searching these parameterized variables and solving the LP models iteratively. Method 2 aims to reduce the computation complexity while providing a near optimal solution. Experiment results show that method 2 is better than method 1, when aggregately considering solution quality and computation efforts.

Keywords Mixed-yield scenario · Product mix planning · Scrapping

1 Introduction

Semiconductor manufacturing is a capital-intensive industry. Building a semiconductor fab (factory) usually costs over 1 billion dollars. How to select market demands to effectively utilize the production resources is therefore very important. Such a decision, often called *product mix planning problem*, is to determine the production quantity for each type of product, given that the future demand of each product has been known.

For a semiconductor fab, different product mix decisions would generate different profits [1]. A fab, which was optimally designed for a particular product mix, produces a maximum throughput in quantity for that product mix. However, the particular product mix may not be the most profitable due to the change of market. Another product mix might become more profitable, yet producing less in throughput. Therefore, the product mix planning problem is a trade-off decision between choosing the most profitable products and maximizing throughput.

Much literature on product mix decision has been published. Kasilingam [2] proposed a nonlinear programming model for solving the product mix decision in the presence of stochastic demand and alternate process plans. Hsu and Chung [3] applied the theory of constraints (TOC) to solve the product mix decision problem. Malik and Sullivan [4] use activity-based costing (ABC) to make product-mix and costing decisions. Kee and Schmidt, in their studies [5, 6], compared the differences of using ABC and TOC in making product mix decisions. Morgan and Daniels proposed methods for integrating product mix and technology adoption decisions [7]. The previous studies, though having established significant milestones, rarely addressed the impact of low production yield in developing their methods.

Low production yield is not unusual for a semiconductor fab, in particular at the stage of developing a new process or manufacturing a new product [8]. The production yield for some new products at their launching stages, from our interview with industry, may be as low as 30%. Wu et al. claimed that scrapping a low-yield wafer lot earlier may increase the profit of a fab [9]. That is, a wafer lot has to be scrapped if its yield after a process is lower than a predefined threshold. They developed a method for identifying the threshold of each process in order to maximize the total profit of a fab.

A competitive semiconductor fab always has to face the problem of developing new processes and new products. Therefore, a fab may be situated in a scenario, which involves the production of both low-yield new products and high-yield mature products. Such a scenario is called a *mixed-yield scenario*. Low-yield new products usually have a higher profit margin, due to their advances in technology or functions. High-yield mature products on the other hand usually have lower profit margin. The product mix decision for such a mixed-yield scenario is very important; however,

M.-C. Wu (✉) · W.-J. Chang · C.-W. Chiou
Department of Industrial Engineering and Management,
National Chiao Tung University,
Hsin-Chu, Taiwan, ROC
E-mail: mcwu@cc.nctu.edu.tw
Fax: +886-35-731913

it has been rarely concerned in literature. Chou and Hong [10] considered the factor of low production yield in solving the product mix decision for a semiconductor fab; yet their study did not consider the strategy of scrapping low-yield wafer lots.

This paper presents a method for making the product mix decision for a mixed-yield semiconductor fab, which adopts a low-yield-scrapping strategy. The product mix decision is formulated by a nonlinear programming model. This research proposes two methods to solve the nonlinear program. The first method obtains the optimal solution at the expense of requiring extensive computation. The second method provides a good or near-optimal solution and requires much less computation.

The remainder of the paper is organized as follows. Sect. 2 reviews the concept of scrapping low-yield lots. Sect. 3 discusses the proposed nonlinear programming model. Sect. 4 presents the two solution methods. Sect. 5 compares the solutions of two methods in various numerical experiments. Some concluding remarks are given in Sect. 6.

2 Scrapping low-yield lots

Scrapping low-yield wafer lots have three main impacts [9]. First, the unit cost of processing a low-yield lot is lower than that of a high-yield lot. Second, the bottleneck of a fab might change when the fab yield greatly decreases. Third, the throughput of a fab might increase or decrease due to the scrapping of low-yield lots. Each impact is explained below.

2.1 Cost behavior of processing low-yield lots

In semiconductor manufacturing, wafers are transported in a cassette (called a lot), which normally carry 25 wafers. Due to yield problems in processing, some wafers in a lot may become out-of-specification and are removed from the lot. Such a lot, less than 25 wafers in number, is called a *small lot*. A lot carrying 25 wafers is herein called a *full lot*. To ensure the production quality, small lots usually cannot be merged into a full lot in their remaining processing.

Workstations in a semiconductor fab can be classified into two types: *batch* and *series*. A series machine processes one wafer at a time. The series-processing cost of each wafer, whether it is carried in a small lot or in a full lot, is equal. A batch machine processes several lots at a time; for example, a furnace machine may process up to six lots at a time. Each lot equally shares the batch-processing cost. Therefore, the *processing cost per wafer* for a small lot is higher than that of a full lot.

2.2 Change of fab bottleneck

In a semiconductor fab, the stepper is usually the most expensive machine and thus relatively few in number. The stepper workstation, being of series-type, is generally assumed to be the bottleneck of the fab in previous literature [3, 5, 6]. Such a bottleneck hypothesis is valid in a high-yield environment. However, the bottleneck may change to a batch workstation in a low-yield environment [9].

The following example illustrates the change of the fab bottleneck due to yield variations. Let a fab be equipped as follows. Among all the *series workstations*, the bottleneck series-workstation is 10 000 wafer/month in capacity. Among all the *batch workstations*, the bottleneck batch-workstation is 800 lots/month in capacity, which is equivalent to 20 000 wafers/month if all lots are full lots (100% yield environment). The bottleneck series-workstation, in a high-yield environment, is therefore the bottleneck of the fab according to the theory of constraints (TOC) [11].

Now suppose the fab is in a low-yield situation, and the average wafer number per lot is ten wafers; namely, with 40% average yield. The bottleneck batch-workstation at most can produce 800 lots/month, which is now equivalent to produces only 8000 wafers/month. Remember that the bottleneck series-workstation can at most produce 10 000 wafers/month. The batch-workstation therefore becomes the bottleneck of the fab in the low-yield environment. That is, in the low-yield environment (40% yield), the fab bottleneck is a batch workstation, while in a high-yield environment (100% yield), the fab bottleneck is a series workstation.

2.3 Impacts on throughput due to scrapping small lots

Setting an appropriate *threshold* for scrapping small lots could increase the profit of a fab [9]. In a typical low-yield fab, only a few *critical operations* accounts for the low yield characteristic. After each critical operation, a threshold is defined for scrapping small lots. For example, a threshold value of five implies that a small lot, less than five wafers in number, should be scrapped and cannot be processed for its remaining operations.

Scrapping small lots may impose positive or negative impacts on the fab throughput. First, the throughput of the fab might increase because scrapping small lots increases the average wafer number per lot. This subsequently increases the output wafers of the bottleneck batch-workstation. If the fab bottleneck is now of batch type, then the throughput of the fab will increase by scrapping small lots. Conversely, the throughput of the fab might decrease because scrapping a small lot make its utilized capacity wasted. That is, the capacity, which has been utilized to produce the small lot, cannot produce any fab throughput.

3 Mathematical model

This section formulates a nonlinear programming model for the product mix decision in a *mixed-yield* scenario adopting the scrapping-small-lot policy. We first model the input/output relationship for a fab of interest. The input denotes the number of wafers released to the fab, and the output denotes the number of finally produced wafers. The input/output relationship model is subsequently used to formulate the nonlinear program.

3.1 Input/output relationship

The process route of manufacturing a wafer is quite long, typically involving hundreds of operations. These operations are usu-

ally segmented into several tens of groups (also called *layers*). In each layer of operations, an inspection has to be performed on each wafer to determine whether the wafer can be processed further. Therefore, each layer can be modeled as a small production system with a particular yield. A full wafer lot after passing a layer of operations, due to a yield problem, might reduce the number of its good wafers and become a small lot.

Consider a fab adopting the strategy of scrapping small lots. Let λ_i represent the number lots of product i released to the fab. Suppose a full lot carry m wafers, then $m \cdot \lambda_i$ wafers are released to the fab. After passing each layer, due to the yield problem, the number of good wafers in a lot might reduce. The number of small lots at the output of each layer is thus a distribution. Let $W_{ij} = [w_k^{ij}]$ represent such a distribution, in which w_k^{ij} represents the number of small lots carrying k $(0 \le k \le m)$ good wafers when product i passes layer j. Thus, $W_{i,0} = [0, 0, \cdots, \lambda_i]$ represents the distribution of small lots at the input of layer 1, which denotes that λ_i lots or $m \cdot \lambda_i$ wafers have been released to the fab.

Let $A_{i,j} = [a_{s,t}^{ij}]$ represent the yield distribution of product i passing layer j. A_{ij} is an $(m+1) \times (m+1)$ matrix, in which $a_{s,t}^{ij}$ $(0 \le s, t \le m)$ represent the probability of a lot with s good wafers becoming a lot with t good wafers. As stated, scrapping small lots based on appropriate thresholds might produce more profit. Let h_{ij} represent the scrapping threshold of product i passing layer j. That is, a small lot with less than h_{ij} good wafers shall be scrapped. Such a scrapping strategy can be modeled by an $(m+1) \times (m+1)$ matrix, $R(h_{i,j}) = [r_{s,t}^{ij}]$. Here, $r_{s,t}^{ij}$ is a binary value (0 or 1), $r_{s,t}^{ij} = 1$ denotes that a lot with s good wafers will become a lot with t good wafers after applying the scrapping strategy; conversely, $r_{s,t}^{ij} = 0$ denotes that there is no such change. The values of $r_{s,t}^{ij}$ can be determined as below.

if $s > h_{i,j}$ and $s = t$, then $r_{s,t}^{ij} = 1$
if $s > h_{i,j}$ and $s \neq t$, then $r_{s,t}^{ij} = 0$
if $s \le h_{i,j}$ and $t = 0$, then $r_{s,t}^{ij} = 1$
if $s \le h_{i,j}$ and $t \neq 0$, then $r_{s,t}^{ij} = 0$

Figure 1 shows the input/output relationship of a layer, which involves three modules: processing, inspection, and scrapping. The input/output relationship for each layer can therefore be modeled as follows.

$$W_{i,j} = W_{i,(j-1)} \times A_{i,j} \times R(h_{i,j})$$

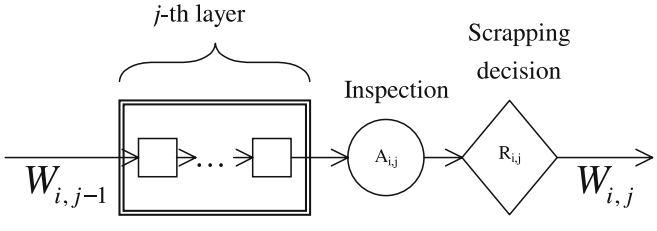

Fig. 1. The input/output relationship of wafer lots distribution between layer $j-1$ and layer j

Suppose $A_{i,j}$, $h_{i,j}$ at each layer are known, for a given wafer release $W_{i,0}$, we can determine the small lot distribution at each layer. Let $S(W_{i,j}) = \sum_{k=1}^{m} k \cdot w_k^{ij}$ represent the total number of good wafers and $L(W_{i,j}) = \sum_{k=1}^{m} w_k^{ij}$ represent the total number of lots, for product i at layer j.

As stated, a semiconductor fab typically involves two types of workstations: batch and series. Let BN_s represent the bottleneck among the series workstations, and BN_b represent the bottleneck among the batch workstations. According to TOC [11], BN_s and BN_b are the two critical factors in constraining the fab throughput.

Let $ts_{i,j}$ represent the processing time of a wafer by BN_s for product i at layer j, $1 \le j \le L_i$, where L_i denotes the number of layers for product i. Let $\bar{h}_i = [h_{i,j}]$ represent the threshold at each layer j for product i. The utilized capacity of BN_s can be computed as follows.

$$Cap\_S(\lambda_i, \bar{h}_i) = \sum_{i=1}^{n} \sum_{j=1}^{L_i} ts_{i,j} \cdot S(W_{i,j-1})$$

Suppose a machine in the batch workstation BN_b can process r lot per operation run. Let $tb_{i,,j}$ represent the operation time per run on BN_b for product i at layer j $(1 \le j \le L_i)$. The utilized capacity of BN_b can be computed as follows.

$$Cap\_B(\lambda_i, \bar{h}_i) = \sum_{i=1}^{n} \sum_{j=1}^{L_i} tb_{i,j} \cdot \frac{L(W_{i,j-1})}{r}$$

To generalize the input/output relationship model, we assume only one wafer lot is released to the fab and define a unit vector $U_{i,j} = \frac{1}{\lambda_i} \cdot W_{i,j}$. That is, $U_{i,j} = [u_k^{ij}]$ represents the distribution of small lots at each layer for each product, when only one lot is released to the fab. Since $W_{i,j} = W_{i,(j-1)} \times A_{i,j} \times R(h_{i,j})$ and $U_{i,j} = \frac{1}{\lambda_i} \cdot W_{i,j}$, the formula $U_{i,j} = U_{i,(j-1)} \times A_{i,j} \times R(h_{i,j})$ therefore holds.

3.2 Product mix decision

The above input/output relationship is used to formulate a nonlinear program for the product mix decision under a mixed-yield scenario, which applies a strategy of scrapping low-yield lots. Notations of the model are given below. To facilitate referencing, some notations, which have been explained above, are also listed here.

A. Notations.

n	Total number of product type
m	The number of wafers in a full lot
L_i	Total number of layers for product i
P_i	Price of product i
FC	Fixed cost of the fab in the concerned time horizon
$C_{i,j} = [c_k^{ij}]$	a $(m+1) \times 1$ matrix, in which c_k^{ij} represents the processing cost of a lot carrying k good wafers for product i at layer j; $C_{i,j}$ denotes the raw wafer cost per lot for $k = 0$.

d_i	Minimum demand of product i
D_i	Maximum demand of product i
AT_s	The available capacity of BN_s
AT_b	The available capacity of BN_b
$ts_{i,j}$	The operation time per wafer on BN_s for product i at layer j
$tb_{i,j}$	The operation time per run on BN_b for product i at layer j
r	The number of lots per run on BN_b
$U_{i,j} = [u_k^{ij}]$	The distribution of small lot when only one lot is released to the fab
$A_{i,j} = [a_{s,t}^{ij}]$	The yield matrix for product i at layer j
$\bar{h}_i = [h_{i,j}]$	The scrapping threshold for product i at each layer
λ_i	The number of lots released to the fab

B. Model

$$\text{Max} \sum_{i=1}^{n} \left[P_i \cdot S(U_{i,L_i}) \cdot \lambda_i - \left(U_{i,0} \times C_{i,0} + \sum_{j=1}^{m} \left(U_{i,j-1} \times C_{i,j} \right) \right) \cdot \lambda_i \right] - FC$$

Subject to

$$U_{i,j} = U_{i,0} \prod_{k=1}^{j} (A_{i,k} \times R(h_{i,k})) \quad 1 \le i \le n; 1 \le j \le L_i \tag{1}$$

$$S(U_{i,j}) = \sum_{k=1}^{m} k u_k^{ij} \quad 1 \le i \le n; 1 \le j \le L_i \tag{2}$$

$$L(U_{i,j}) = \sum_{k=1}^{m} u_k^{ij} \quad 1 \le i \le n; 1 \le j \le L_i \tag{3}$$

$$\sum_{i=1}^{n} \left[\sum_{j=1}^{L_i} ts_{i,j} \cdot S(U_{i,j-1}) \right] \cdot \lambda_i \le AT_s \tag{4}$$

$$\sum_{i=1}^{n} \left[\sum_{j=1}^{L_i} tb_{i,j} \cdot \frac{L(U_{i,j-1})}{r} \right] \cdot \lambda_i \le AT_b \tag{5}$$

$$S(U_{i,L_i}) \cdot \lambda_i \ge d_i \quad 1 \le i \le n \tag{6}$$

$$S(U_{i,L_i}) \cdot \lambda_i \le D_i \quad 1 \le i \le n \tag{7}$$

$$h_{i,j} \ge h_{i,k} \text{ if } j < k \quad 1 \le i \le n \tag{8}$$

In the above formulation, the objective function models the profit of the fab, in which the term

$$P_i \cdot S(U_{i,L_i}) \cdot \lambda_i$$

models the revenue, the term

$$\left(U_{i,0} \times C_{i,0} + \sum_{j=1}^{m} \left(U_{i,j-1} \times C_{i,j} \right) \right) \cdot \lambda_i$$

describes the variable costs, and FC is the fixed cost.

Constraint 1 describes the input/output relationship of each layer. Constraint 2 is used to compute the number of good wafer outputted at each layer. Constraint Eq. 3 intends to compute the number of lots outputted at each layer. Constraints 4–5 request that the utilized capacity of BN_s and BN_b cannot be greater than its available capacity. Constraint 6 denotes the minimum output quantity for each product, requested by the management. Constraint 7 describes the maximum market demand of each product.

Constraint 8 prohibits an incompatible setting of threshold values. That is, a downstream layer cannot define a threshold value higher than its upperstream layer. For example, suppose the threshold for layer 1 is $h_{i,1} = 6$ and that for layer 2 is $h_{i,2} = 8$. Based on such a scrapping strategy, a lot with seven good wafers will pass layer 1 and be processed at layer 2. Yet, whatever the yield of layer 2 is, the lot will be surely scrapped after layer 2 because $h_{i,2} = 8$. Such an incompatible threshold setting is undoubtedly prohibited.

4 Solution methods

This section first analyzes the complicated characteristics in solving the mathematical program formulated in the above section. Two proposed methods for solving the mathematical programs are then presented.

(A) Analysis of the mathematical program. The formulation in the above section is a nonlinear program, in which $h_{i,j}$ and λ_i are independent decision variables. Constraint 1, the multiplication of several matrices including $R(h_{i,j})$, essentially gives the nonlinear relationship. Solving such a nonlinear program appears to be quite complicated, and needs a method for making it simplified.

The nonlinear program can be simplified as a linear program if $h_{i,j}$ are given parameters. Namely, suppose $h_{i,j}$ are given parameters which meet Constraint 8, we can determine the vector of $U_{i,j}$ and subsequently the values of $S(U_{i,j})$ and $L(U_{i,j})$. Constraints 1–3 can therefore be removed. The above nonlinear program formulation now becomes a linear program, which involves only four constraints, 4–7. In the linear program, $h_{i,j}$, $U_{i,j}$, $S(U_{i,j})$ and $L(U_{i,j})$ are treated as parameters, and λ_i are the decision variables. The two proposed methods for solving the nonlinear program are both based on such a simplification, and become solving linear programs.

(B) Method 1. The first method starts with determining a solution space of $\bar{h}_i$ and then solves the linear program for each element in the solution space. Let S represent such a solution space. An element in S, $\{\bar{h}_i = [h_{i,j}] | 1 \le i \le n\}$ is a set of vectors, which denotes a combination of thresholds for each product i at each critical layer j. Let C represent the number of critical layers and $N(S)$ represent the number of elements in S. Then $N(S) = m^{nC}$, if Constraint 8 is not considered, where m denotes the number of wafers in a full lot. Namely, assume that $m = 25$, $C = 2$, $n = 2$, then $N(S) = 25^4 = 390,625$. This implies that we have to solve the linear program 390 625 times to obtain the solution of the

product mix decision problem. Such a computation requirement seems acceptable.

However, the method is in nature computationally extensive. For a case with $m = 25$, $C = 2$, $n = 10$, a huge number of computation, $N(S) = 25^{20}$, which appears to be infeasible to solve the problem in an acceptable time. Therefore, another approach to solve such a problem is by adopting some *meta-heuristics algorithms* such as genetic algorithms [12], Tabu search algorithms [13], or simulated annealing algorithms [14]. The basic ideas of these meta-heuristics algorithms are *randomly* and *wisely* selecting some good elements in S in an non-exhaustive search manner. Such approaches will greatly reduce the number of computations; however, the obtained solution is not an optimal solution. This research adopts the exhaustive search approach in the numerical experiments.

(C) Method 2. The other method solves the formulated nonlinear program by a two-stage approach. The first stage determines the threshold $\bar{h}_i$ for each product i, and the second stage solves the linear program by taking the obtained $\bar{h}_i$ as parameters.

In determining $\bar{h}_i$, we assume that the fab produces only one product i. The bottleneck of the fab may be either of series-type or of batch-type. If the fab bottleneck is of series-type, then Constraint 5 in the nonlinear program model can be removed. Likewise, if the fab bottleneck is of batch-type, then Constraint 4 in the nonlinear program model can be removed. Following the discussion about *method 1*, we need only solve a linear program m^C times to obtain $\bar{h}_i$ for a particular product i based on a particular assumption of fab bottleneck.

For a particular assumption of the fab bottleneck, we need to solve the linear program $n \cdot m^C$ times, if Constraint 8 is not considered, to obtain the thresholds of the n products. There are two assumptions for the fab bottleneck. Therefore, we need to solve the linear program $2 \cdot n \cdot m^C$ times. For the above case with $m = 25$, $C = 2$, $n = 10$, method 2 only need to solve the linear program 12 500 times, greatly less than 25^{20} times required by *method 1*. The basic idea of method 2 essentially adopts a *problem-decomposition* approach, and cannot guarantee an optimal solution.

The required computation for method 2 surely can also be computationally extensive if C is a large number. However, in a typical fab, only a few layers are critical layers. That is, the number of critical layers (C) may only range from 1 to 3. Therefore, method 2 may not be so computationally extensive in real application.

5 Numerical examples

Without loss of generality, a hypothetic fab is first presented to illustrate the product mix decision problem. We then analyze whether the strategy of scrapping small lots is better than the no-scrapping strategy in a mixed–yield scenario. Subsequently, the scrapping thresholds determined in the first stage of method 2 are discussed. Finally the solution results of the two proposed methods for solving a product mix decision problem are compared.

(A) Fab data. The fab produces two products ($n = 2$), and each product has 20 layers of which layer 2 and 3 are the two critical layers. The yield at each non-critical layer is 100%. The yields of the two critical layers are equal. Let y represent the aggregated fab yield and p represent the yield of a critical layer. If $y = 40\%$, then $p = 63\%$ ($p = \sqrt{y}$). The yield matrix $A_{i,j} = [a^{ij}_{s,t}]$ is so designed based on the *binomial distribution* as follows:

$$a^{ij}_{s,t} = C^s_t p^t (1-p)^{s-t} \text{ for } s \geq t, \text{ and } a^{ij}_{s,t} = 0 \, for \, s < t.$$

A full lot has 25 wafers ($m = 25$). The raw wafer cost is $c^{ij}_0 =$ \$2000. The variable cost for processing at each layer, $[c^{ij}_k](1 \leq k \leq 20)$, is shown in Table 1. The fixed cost is $FC = \$1.1 \times 10^7$.

The parameters ts_{ij} and tb_{ij} are assumed to be constant, namely $ts_{ij} = ts$ and $tb_{ij} = tb$. The capacity of bottleneck series-workstation is so designed that $\frac{AT_s}{ts} = 900\ 000$ *wafer-layers*. Likewise, the capacity of the bottleneck batch-workstation is such designed, $\frac{AT_b \cdot r}{tb} = 42,000$ *lot-layers*, which is equivalent to $1\ 050\ 000 = 42\ 000 \times 25$ *wafer-layers* if all lots are full lots.

(B) Comparison between scrapping and no-scrapping. Suppose that the fab is in a *mixed-yield* scenario. The first product is with high yield, $y_1 = 90\%$; the second product is with low yield, $y_2 = 40\%$. Two scrapping alternatives are to be compared. Alternative 1 uses the strategy of scrapping small lots, and alternative 2 does not scrap any small lots. Let P_1 and P_2 respectively denote the unit price of the first and the second product; d_1, d_2 respectively represent the lower bound of the production quantity of product 1 and product 2. Table 2 shows the profit difference of the two alternatives for $d_1 = 0$ and $d_2 = 0$. From the table, the strategy of scrapping small lots will yield more profit than the no-scrapping strategy. The profit differences, in the example, range from 5.2% to 7.8%. This finding implies that scrapping small lots is also a good strategy in a mixed-yield scenario, just as that in a low-yield scenario [9].

Table 1. Variable processing cost at each layer

$c^{ij}_1 = \$283$	$c^{ij}_6 = \$320$	$c^{ij}_{11} = \$358$	$c^{ij}_{16} = \$395$	$c^{ij}_{21} = \$433$
$c^{ij}_2 = \$290$	$c^{ij}_7 = \$328$	$c^{ij}_{12} = \$365$	$c^{ij}_{17} = \$403$	$c^{ij}_{22} = \$440$
$c^{ij}_3 = \$298$	$c^{ij}_8 = \$335$	$c^{ij}_{13} = \$373$	$c^{ij}_{18} = \$410$	$c^{ij}_{23} = \$448$
$c^{ij}_4 = \$305$	$c^{ij}_9 = \$343$	$c^{ij}_{14} = \$380$	$c^{ij}_{19} = \$418$	$c^{ij}_{24} = \$455$
$c^{ij}_5 = \$313$	$c^{ij}_{10} = \$350$	$c^{ij}_{15} = \$388$	$c^{ij}_{20} = \$425$	$c^{ij}_{25} = \$463$

Table 2. Profit comparison between scrapping and no-scrapping strategies

	Profit of using scrapping strategy (\$M)	Profit diff. (\$M)	Profit diff (%)
$P_1 = \$500$ $P_2 = \$2,900$	32.2	1.6	5.2%
$P_1 = \$500$ $P_2 = \$1,800$	8.3	0.6	7.8%

Table 3. Scrapping thresholds for the low yield product determined in method 2

Yield	Bottleneck series		Bottleneck batch	
	h_2	h_3	h_2	h_3
40%	7	4	12	7
50%	8	5	12	9
60%	9	6	14	11
70%	10	7	16	13
80%	0	0	18	15
85%	0	0	18	16
90%	0	0	19	17

Table 5. Comparison of computation time (* denotes by estimation)

Problem size	Method	Computation time	Number of times in solving LP program
Fab 1	Method 2	2 min	1302
$n = 2$	Method 1	240 min	105 625
$m = 25$, $C = 2$			
Fab 2	Method 2	5 min	1952
$n = 3$	Method 1	54 days*	34 328 125
$m = 25$, $C = 2$			

(C) Determining thresholds of method 2. In method 2, the scrapping threshold for each product, based on each assumption of fab bottleneck, should be first determined. Table 3 shows such results in various fab yields. Let h_2, h_3 respectively represent the threshold of the two critical layers, layer 2 and layer 3, of a particular product. Notice that the scrapping thresholds (h_2 and h_3) may change when the fab bottleneck changes. Refer to the first row in Table 3, where the fab yield of a product is $y = 40\%$. The thresholds are $h_2 = 7$, $h_3 = 4$ when the fab bottleneck is of series-type, and $h_2 = 12$ and $h_3 = 7$ when the fab bottleneck is of batch-type.

Moreover, when the fab yield changes, the thresholds (h_2 and h_3) may also change. Refer to the first two rows in Table 3, which address two fab yields $y = 40\%$ and $y = 50\%$. When the fab yield increases, the threshold values in general will increase. However, when the fab yield is high enough ($y > 80\%$), the threshold value becomes zero; namely, it requires no scrapping. This phenomenon can be interpreted as follows. When the fab yield increases, the number of small lots, less than a certain threshold in the number of good wafers, may decrease. The scrapping threshold therefore may need to be raised to increase profit. Yet, when the fab yield is high ($y = 81\%$, or $p = 90\%$), the probability of producing small lots becomes quite low. Therefore, we may not need taking any scrapping strategy (referring to the last three rows of Table 3).

(D) Comparison of the two proposed methods. The two proposed methods for solving the nonlinear program are compared. Let h_{i2}, h_{i3} represent the scrapping threshold of layer 2 and 3 of product i ($i = 1, 2$). Table 4 shows the solution results of adopting method 1 and method 2 in four mixed-yield scenarios. Here, method 1 uses an exhaustive search for determining h_{i2} and h_{i3} and solve the linear program for each combination of (h_{12}, h_{13}, h_{22}, h_{23}). The solution obtained by such an exhaustive search is an optimal solution.

Table 4 shows the solution results of the two proposed methods. In the table, *method_2_batch* denotes that the fab bottleneck is assumed to be of batch workstation, and *method_2_series* denotes that the fab bottleneck is assumed to be of series workstation. The table shows that the fab bottleneck is of batch-type when method 2 is used to solve each of the four mixed-yield scenarios. For the four mixed-yield scenarios, the scrapping thresholds h_{ij} suggested by method 2 and method 1 only slightly differ. The solution quality of method 2 is very good, though not optimal, only about 0.007% less than the optimal solution provide by method 1.

Table 5 shows the computation efforts required by the two methods for two hypothetical fabs. Fab 1 produces two products (i.e., $n = 2$, $m = 25$, $C = 2$), and fab 2 produces three products (i.e., $n = 3$, $m = 25$, $C = 2$). In solving the product mix decision problem of fab 1, method 1 needs about 240 min, (4 hours), which solves 105 625 LP programs and method 2 needs only 2 min, which solves 1302 LP programs. In solving the problem of fab 2, method 2 takes about 5 min, which solves 1952 LP programs; method 1 needs a formidable computation time, about 54 days by estimation (34 328 125/105 625 *4/24). The computer program for the above experiments is coded in Matlab software,

Table 4. Solution comparisons between method 1 and method 2 ($P_1 = \$1440$, $P_2 = \$2880$)

Yield	Method	h_{12}	h_{13}	h_{22}	h_{23}	Profit ($)	Profit diff ($)
$p_1 = 90\%$	Method 1	19	17	11	7	34 216 809	–
$p_2 = 40\%$	Method_2_batch	19	17	12	7	34 214 340	−2469
	Method_2_ series	0	0	7	4	34 067 098	−149 711
$p_1 = 85\%$	Method 1	18	16	12	7	34 598 533	–
$p_2 = 40\%$	Method_2_batch	18	16	12	7	31 598 533	0
	Method_2_ series	0	0	7	4	30 649 497	−949 036
$p_1 = 70\%$	Method 1	16	13	12	7	31 318 551	–
$p_2 = 40\%$	Method_2_batch	16	13	12	7	31 318 551	0
	Method_2_ Series	10	7	7	4	29 972 939	−1 345 612
$p_1 = 60\%$	Method 1	15	11	12	7	31 059 049	–
$p_2 = 40\%$	Method_2_batch	14	11	12	7	31 058 649	−400
	Method_2_ series	9	6	7	4	29 710 377	−1 348 672

running on a personal computer equipped with CPU 3.0 GHZ and DRAM 512 MB.

From the numerical experiments, method 2 seems better than method 1 due to its much less computation requirements and its good solution quality. The computation efforts would be greatly saved in Method 2 when the problem size increases.

6 Concluding remarks

This paper presents a product mix decision problem in a mixed-yield semiconductor fab, which simultaneously produces high yield and low yield products. Scrapping low-yield wafer lots during the process may increase profit. Yet, this strategy has not been included in the previous literature on product mix decision. In the semiconductor industry, new products and new processes have been continuously developed and have low yields in its early stage. This research therefore includes the scrapping strategy in addressing the product mix decision problem.

A nonlinear program is developed to model such a product mix decision problem. The decision problem involves two sets of decision variables: scrapping thresholds (h_{ij}) and product release quantity (λ_i). Two solution methods are proposed to solve the nonlinear program, by setting the scrapping threshold variables as parameters and convert the nonlinear program into a linear program. Method 1 uses an exhaustive search to identify h_{ij} for all products and solve the associated linear program extensively. Method 2 individually determines h_{ij} for each product i, and then solve the associated linear program. Method 1 provides an optimal solution, but is computationally extensive. Though not providing an optimal solution, method 2 is very good in solution quality, only less than the optimal solution by 0–0.007% and requires much less computation than method 1. This research therefore suggests the use of method 2 in solving the product mix decision problem.

This research in modeling the capacity of bottleneck workstations adopts a static capacity model. That is, the factor of cycle time is not addressed. Future extension of this research includes the consideration of cycle time in dealing with the mixed-yield product mix decision problem. Another extension to this research is developing some meta-heuristics algorithms such as genetic algorithms, Tabu search algorithms, and simulated annealing algorithms to enhance method 1.

References

1. Li S, Tirupati D (1997) Impact of product mix flexibility and allocation policies on technology. Comput Oper Res 24(7):611–626
2. Kasilingam RG (1995) Product mix determination in the presence of alternate process plans and stochastic demand. Comput Ind Eng 29(1–4):249–253
3. Hsu TC, Chung SH (1998) The TOC-based algorithm for solving product mix problems. Prod Plan Control 9(1):36–46
4. Malik SA, Sullivan WG (1995) Impact of ABC information on product mix and costing decisions. IEEE Trans Eng Manage 42(2):171–176
5. Kee R (1995) Integrating activity-based costing with the theory of constraints to enhance production-related decision-making. Am Account Assoc Account Horizons 9(4):48–61
6. Kee R, Schmidt C (2000) A comparative analysis of utilizing activity-based costing and the theory of constraints for marking product-mix decisions. Int J Prod Econ 63:1–17
7. Morgan LO, Daniels RL (2001) Integrating product mix and technology adoption decisions: a portfolio approach for evaluating advanced technologies in the automobile industry. J Oper Manage 19:219–238
8. Maynard DN, Kerr DS (2003) Cost of yield. IEEE/SEMI Advanced Manufacturing Conference, pp 165–170
9. Wu MC, Chiou CW, Hsu HM (2004) Scrapping small lots in a low-yield and high-price scenario. IEEE Trans Semiconduct Manuf 17(1):55–67
10. Chou YC, Hong IH (2000) A methodology for product mix planning in semiconductor foundry manufacturing. IEEE Trans Semiconduct Manuf 13(3):278–285
11. Goldratt EM (2000) Necessary but not sufficient: a theory of constraints business novel. North River, Great Barrington, MA
12. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. Wiley, New York
13. Glover F (1990) Tabu search: a tutorial. Interfaces 20(4):74–94
14. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680