Scrapping Small Lots in a Low-Yield and High-Price Scenario

Muh-Cherng Wu, Chie-Wun Chiou, and Hsi-Mei Hsu

***Abstract*—Some wafers in a lot may become spoiled after they are processed at a workstation; such a lot is called a small lot. In a low yield and high price scenario, scrapping small lots may increase revenue and profit; yet, this notion has seldom been examined. This study presents a model for formulating the decision-making problem of scrapping small lots. A genetic algorithm is used to solve the problem when the solution space is large. An exhaustive search method is used when the solution space is small. Some numerical examples are used to evaluate the outcome of scrapping small lots. The profit obtained by the proposed scrapping method may be up to 23% higher than that obtained without scrapping.**

***Index Terms*—Bottleneck, high price, low yield, product introduction, small lots.**

I. Introduction

IN SEMICONDUCTOR wafer fabrication, a factory (commonly called a *fab*) includes around 100 workstations, each of which has many functionally identical machines. A wafer normally requires 300–600 operations before it is completed, and each operation is performed in a workstation. The wafer fabrication process is called a reentry system because a wafer may visit a workstation several times. In practice, the hundreds of operations are often grouped into several tens of *layers*. A layer includes a sequence of operations, and one of the operations is processed at a *photo* workstation. To reduce management complexity, some decision-making problems are solved at the granularity of layers rather than operations.

A wafer in a semiconductor fab is transported in a fixed-size batch. Such a batch is called a *wafer lot* (or a *lot*, for short) that normally includes 25 wafers. Due to yield problems, some wafers in a lot during processing may become spoiled and cannot be processed further. The number of good wafers in a lot is then less than 25; such a lot is called a *small lot*. A lot that includes 25 good wafers is called a *full lot*. For reasons of quality, wafers in different small lots usually cannot be merged into a single lot. Such merging may further reduce the yield of small lots in their remaining operations. Our interviews with experienced process engineers reveal that the merge of small lots will greatly reduce the yield, particular in steppers and chemical mechanical polishing (CMP) machines. Therefore, in practice, small lots are usually not merged.

Manuscript received January 31, 2003; revised November 25, 2003. This work was supported under Contract NSC91-2213-E-009-134.

The authors are with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsin-Chu 300, Taiwan, R.O.C. (e-mail: mcwu@cc.nctu.edu.tw; cw.chiou@msa.hinet.net; hsimei@nctu.edu.tw).

Digital Object Identifier 10.1109/TSM.2003.822732

The manufacturing cost per wafer for a small lot, in some workstations, is higher than that of a full lot. Machines in a semiconductor fab are generally classified into two types, *series type* and *batch type*. The processing unit of a series-type machine is a wafer, while that of a batch-type machine is a batch of lots. A batch-type machine in one run (including one operation) may simultaneously process a batch up to six lots, over which the running costs are equally distributed. The running cost per wafer for a small lot on a batch-type machine is therefore higher than that for a full lot. In contrast, a series-type machine in one run processes a single wafer. The running cost per wafer on a series-type machine is, therefore, independent of lot size.

A semiconductor fab may face a decision problem about the scrapping of small lots. For example, given a small lot of 12 good wafers and with ten layers remaining to be processed, should the fab keep the lot for further processing or scrap it? Keeping the small lot until its completion will create revenue, while scrapping the lot provides an opportunity for processing new full lots. As stated, the manufacturing cost per wafer for a small lot exceeds that for a full lot. These cost/revenue factors should all be included when making the decision to scrap.

Making an effective decision to scrap small lots is very important, particularly at a low-yield and high-price scenario. The wafer yield of a fab may be quite low due to the introduction of advanced processes (e.g., low-k or copper processes) or new technologies (e.g., 0.13 or 0.09 μm). The wafer yield for a newly launched product might be very low due to the use of advanced processes or technologies. Our interviews with some wafer fabs in industry reveal that the wafer yield for 90-nm process technology was in the range of 30%–50% in 2002. Semiconductor companies generally would prefer to introduce the low-yield but new products to the market for two reasons. First, an early introduction tends to attract future customers and increase market share. Second, the price at this stage is often high enough that a low-yield production may still be profitable. The decision problem for scrapping small lots is, therefore, very important in a low-yield and high-price scenario.

Much literature on semiconductor yield modeling and its applications has been published [5], [14], [18]. Yet, very few study the decision for scrapping small lots in semiconductor manufacturing. Daigle and Powell propose a formalized management procedure to reduce wafer scraps [4]. Wu *et al.* analyzed the tradeoff factors of the scrap decision problem [22].

Based on the cost of yield, Maynard *et al.* proposed a heuristic method in IBM for the scrap decision of wafers [16], [17]. The heuristic method only suggests the layers at which small lots should be scrapped and cannot not determine the *threshold* for

scrapping a small lot. That is, they cannot determine how small a lot should be scrapped. Moreover, the opportunity cost for releasing new wafers is not modeled in their studies.

Interviews from industrial workers reveal that the decision of scrapping small lots is often made using the following two methods.

The first method is to set a universal threshold at each layer for scrapping small lots. For example, if the universal threshold is set to five, then any small lots with five or fewer wafers should be scrapped. This practice has two drawbacks. First, the determination of the threshold is heuristic. A cost and benefit analysis for such a heuristic rule is not available. Second, the threshold for various layers is fixed, but it should be different for each layer to possibly yield a better solution.

The second method is called the *sunk cost* approach. For a small lot about which a decision to scrap is to be made, the costs incurred in processing are sunk costs [3], [15]. The sunk cost method states that the sunk costs of the lot are *past* costs and therefore nothing to do with the scrapping decision; only *future revenues and costs* of the lot should be considered. That is, a small lot should be kept if its expected final revenue exceeds the remaining processing costs. Otherwise, the small lot should be scrapped. The sunk cost method sounds reasonable. However, the scrapping of a small lot provides an opportunity for manufacturing new lots, given limited capacity. A small lot may produce profit, but scrapping it and releasing the capacity to new lots might create more profit. The sunk cost approach is, therefore, deficient in neglecting the cost/benefit analysis and the opportunity provided by scrapping small lots. The benefit of releasing more new lots by scrapping small lots is called *opportunity cost* [3], [15].

This paper develops a mathematical model for the decision problem of scrapping small lots in a semiconductor fab. Based on the model, a genetic algorithm is proposed for making the scrapping decisions at each layer. However, when the number of low-yield layers is few, the exhaustive search method is used to determine the associated scrapping rules. Simulation experiments show that scrapping small lots as proposed may considerably increase profit.

The remainder of this paper is organized as follows. Section II describes the mathematical model of the problem. Section III explains the genetic algorithm. Section IV presents the assumptions and the cost data of the semiconductor fab used in the test examples. Section V compares the solutions of the proposed approach with those of other decision-making methods. Concluding remarks are presented in Section VI.

II. Model

This section presents a model for formulating the decision problem of scrapping small lots in a semiconductor fab. Assumptions of the semiconductor fab are first discussed. The notation of the model is introduced and subsequently explained with reference to a simplified fab. Finally, the decision problem is formulated and the complexity of the solution space is analyzed to explain the proposed use of a genetic algorithm to solve the problem.

A. Assumptions Concerning the Semiconductor Fab

The semiconductor fab of interest produces only one product and involves two types of workstations, the series type and the batch type. Let BT_s represent the bottleneck of the series-type workstations and BT_b represent the bottleneck of the batch-type workstations. Here, the bottleneck represents the most highly utilized one in a specified group of workstations. BT_s and BT_b of the fab are assumed to have been identified. According to the theory of constraints (TOC) [10]–[13], the capacity of the fab is limited either by BT_s or BT_b, according to which is more highly utilized. Section V will demonstrate that in a low-yield environment, the bottleneck of a fab may switch between BT_s and BT_b, primarily depending upon where the low-yield layers are. The decision problem is considered for a specified time horizon so that the available run time (or *capacity*) of each workstation is known.

B. Notation

Parameters

L total number of layers;
M total number of wafers in a full lot;
AT_s capacity (available run time) of BT_s;
AT_b capacity (available run time) of BT_b;
ts_i required run time of an operation processed by BT_s at layer i; $0 \le i \le L$;
tb_i required run time of an operation processed by BT_b at layer i; $0 \le i \le L$;
n number of lots simultaneously processed by BT_b;
P price of the product;
FC fixed cost of the fab;
$\overline{C_i}$ $= [c_k^i]^T$ processing cost per lot at layer i, $0 \le k \le M$; $0 \le i \le L$;
c_k^i processing cost for a lot with k wafers at layer i;
A_i $= [a_{jk}^i]$ yield matrix at layer i, $0 \le j \le M$; $0 \le k \le M$; $0 \le i \le L$;
a_{jk}^i probability that a lot with j wafers becomes one with k wafers, after completing the operations at layer i;

$$\text{if } j \ge k, \text{ then } 1 \ge a_{jk}^i \ge 0$$
$$\text{if } j < k, \text{ then } a_{jk}^i = 0$$
$$\sum_{k=0}^{M} a_{jk}^i = 1.$$

Variables

$\overline{U_i}$ $= [u_k^i]$ distribution of output lots at layer i when only one lot is released to the fab, $0 \le k \le M$, $0 \le i \le L$; $\overline{U_0} = [1, 0, \ldots, 0]$;
u_k^i number of output lots that carry k wafers at layer i when only one lot is released to the fab;
$\overline{W_i}$ $= [w_k^i]$ distribution of output lots at layer i, $0 \le k \le M$, $0 \le i \le L$;
w_k^i number of output lots that carry k wafers at layer i;
$S(\overline{W_i})$ $= \sum_{k=1}^{M} k \cdot w_k^i$ total number of output wafers at layer i;
$L(\overline{W_i})$ $= \sum_{k=1}^{M} w_k^i$ total number of output lots at layer i;

$\overline{h}$ $= [h_i]$ decision vector for scrapping small lots;
h_i threshold for scrapping small lots at layer i, $1 \le i \le L$ and $1 \le h_i \le M-1$;
$R(h_i)$ $= [r^i_{jk}]$scrapping matrix at layer i, $0 \le j \le M$; $0 \le k \le M$; $1 \le i \le L$;
r^i_{jk} a binary variable (0 or 1)

$$\begin{aligned}
&\text{if } j > h_i \text{ and } k = j, \text{ then } r^i_{jk} = 1;\\
&\text{if } j > h_i \text{ and } k \neq j, \text{ then } r^i_{jk} = 0;\\
&\text{if } j \le h_i \text{ and } k = 0, \text{ then } r^i_{jk} = 1;\\
&\text{if } j \le h_i \text{ and } k \neq 0, \text{ then } r^i_{jk} = 0;\\
&\sum_{k=0}^{M} r^i_{jk} = 1;
\end{aligned}$$

$\lambda(\overline{h})$ number of input lots that can fully utilize the bottleneck of the fab, when $\overline{h}$ is applied in the fab.

C. Example to Explain Notation and Model

Without loss of generality, a simplified semiconductor fab is considered to clarify the above notation. The process comprises three layers $L = 3$ (Fig. 1) and a full lot carries three wafers $M = 3$. Suppose that within the time horizon, α wafer lots are released to the fab (e.g., $\alpha = 1000$). Let $\overline{U_0}$ represent [0 0 0 1]. The distribution of wafer lots $(\overline{W_0})$, the number of total good wafers $S(\overline{W_0})$, and the total number of lots $L(\overline{W_0})$ at the wafer start station (layer 0) can be expressed as follows:

$$\begin{aligned}
\overline{W_0} &= \alpha\overline{U_0} = 1000 \cdot [0 \quad 0 \quad 0 \quad 1] = [0 \quad 0 \quad 0 \quad 1000]\\
S(\overline{W_0}) &= \sum_{k=1}^{3} k \cdot w^0_k = 3000\\
L(\overline{W_0}) &= \sum_{k=1}^{3} w^0_k = 1000.
\end{aligned}$$

In these expressions, $w^0_0 = w^0_1 = w^0_2 = 0$ implies that no lot carries fewer than three wafers at layer 0. $w^0_3 = 1000$ denotes that each of the 1000 lots has three wafers. The total number of wafers $S(\overline{W_0})$ is 3000 and the total number of lots $L(\overline{W_0})$ is 1000.

Due to the yield problem, some wafers in a full lot may become *spoiled* after undergoing a layer of operations. The resulting lot may consequently become a small lot, which carries zero, one, or two wafers. Suppose that the following matrix A_1 describes the yield distribution after layer 1 is passed. For a full lot that has just passed layer 1, the probability that it becomes a lot with 0, 1, 2, or 3 wafers is $a^1_{30} = 0.01$, $a^1_{31} = 0.02$, $a^1_{32} = 0.03$, or $a^1_{33} = 0.94$, respectively. For $0 \le k \le 2$, $a^1_{kk} = 1$ and $a^1_{kj} = 0$ when $j \neq k$ denotes that the lots with fewer than three wafers are unchanged, though no such lot exists at layer 0

$$A_1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.01 & 0.02 & 0.03 & 0.94 \end{pmatrix} \end{array}.$$

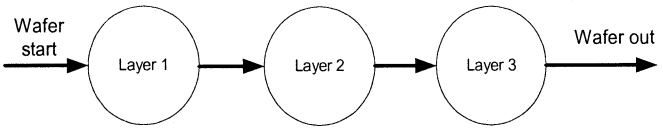


Fig. 1. Simplified process route for manufacturing a semiconductor product.

After the operations in layer 1 are finished *without applying any scrapping rule*, the distribution of wafer lots $(\overline{W_1})$ and the associated $S(\overline{W_1})$ and $L(\overline{W_1})$ can be computed as

$$\begin{aligned}
\overline{W_1} &= \overline{W_0} \times A_1 = [10 \quad 20 \quad 30 \quad 940]\\
S(\overline{W_1}) &= \sum_{k=1}^{3} \left(k \times w^1_k\right) = 2900\\
L(\overline{W_1}) &= \sum_{k=1}^{3} w^1_k = 990.
\end{aligned}$$

These expressions reveal that 60 full lots now become small lots, of which ten lots have zero wafers, 20 have one wafer, and 30 have two wafers. The remaining 940 lots still have three wafers. The total number of wafers $S(\overline{W_1})$ is 2900 and the total number of lots $L(\overline{W_1})$ is only 990.

In the simplified fab, the threshold for scrapping a small lot (h_i) must be either 1 or 2 because a full lot includes three wafers. Let the threshold for scrapping small lots at layer 1 be one wafer, $(h_1 = 1)$. The scrapping matrix at layer 1, $R(h_1)$, can be expressed as follows. $r^1_{00} = r^1_{10} = 1$ denotes that lots with zero or one wafer should be scrapped, while $r^1_{22} = r^1_{33} = 1$ denotes that lots with two or more wafers should be kept for further processing

$$R(h_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

After the scrapping rule h_1 is implemented, the expression of $\overline{W_1}$ can be modified as follows, where A_1 represents the yield matrix and $R(h_1)$ denotes the scrap matrix. Notice that the wafer lots that carry a single wafer are now scrapped, that is, $w^1_1 = 0$. The total number of lots $L(\overline{W_1})$ falls to 970 because 20 lots with one wafer have now been scrapped. The total number of wafers $S(\overline{W_1})$ is reduced to 2880

$$\begin{aligned}
\overline{W_1} &= \overline{W_0} \times A_1 \times R(h_1) = [0 \quad 0 \quad 30 \quad 940]\\
S(\overline{W_1}) &= \sum_{k=1}^{3} \left(k \times w^1_k\right) = 2880\\
L(\overline{W_1}) &= \sum_{k=1}^{3} w^1_k = 970.
\end{aligned}$$

For a product with L layers, let the yield matrix (A_i) and the scrapping rule (h_i) at each layer i be known. Suppose α wafer lots are released to the fab, that is, $\overline{W_0} = \alpha\overline{U_0}$ where

$\overline{U_0} = [0 \ldots 0 \ldots 01]$. Following the above procedure, $\overline{W_i}$, the distribution of output wafer lots at layer i can be expressed as

$$\overline{W_i} = \overline{W_0} \times \prod_1^i (A_i \times R(h_i)) = \alpha \overline{U_0} \times \prod_1^i (A_i \times R(h_i)).$$

The distribution of the final output lots is $\overline{W_L}$, the total number of final output wafers is $S(\overline{W_L})$. The scrapping decision at each layer i forms a decision vector $\overline{h} = [h_1 \ldots h_i \ldots h_L]$, where h_i is a decision variable. Changing the scrapping decision vector $(\overline{h})$ will change the number of output wafers $S(\overline{W_L})$. Notably, $h_L = 0$ because no scrapping is required at the last layer L.

As stated, BT_s represents the bottleneck of series-type workstations. Suppose each output wafer must visit BT_s once at each layer, and the processing time at layer i is ts_i. When α wafer lots $(\alpha = 1)$ are released to the fab and a scrapping decision vector $(\overline{h})$ is predefined, the required run time (or *used capacity*) of BT_s can be expressed by $\mathrm{Cap\_}S(\alpha, \overline{h})$ formulated below. Notably, the subscript i runs from 0 to $L-1$ because the input lots to layer i is $\overline{W_{i-1}}$ and the number of wafers to be processed at layer i is $S(\overline{W_{i-1}})$. Also, α and $\overline{h}$ are variables, that is, the utilized capacity of BT_s depends upon how α and $\overline{h}$ are

$$\mathrm{Cap\_}S(\alpha, \overline{h}) = \sum_{i=0}^{L-1} ts_{i+1} \cdot S(\overline{W_i}).$$

The used capacity of the bottleneck $\mathrm{Cap\_}S(\alpha, \overline{h})$ should equal the available capacity AT_s. Let $\lambda_s(\overline{h})$ be the number of input lots required to utilize fully BT_s. Then, $\lambda_s(\overline{h})$ can be determined as

$$\lambda_s(\overline{h}) = \frac{\mathrm{AT}_s}{\sum_{i=0}^{L-1} ts_{i+1} \cdot S(\overline{W_i})}.$$

As stated, BT_b represents the bottleneck of batch-type workstations. Suppose that each lot must visit BT_b once at each layer. The processing time for a batch of lots at layer i is tb_i. The batch-type workstation BT_b, during each operation, must process a batch of n lots simultaneously (for example, $n = 6$). When α wafer lots are released to the fab and a scrapping decision vector $(\overline{h})$ is given, the required run time (or used capacity) of BT_b is

$$\mathrm{Cap\_}B(\alpha, \overline{h}) = \sum_{i=0}^{L-1} tb_{i+1} \cdot \frac{L(\overline{W_i})}{n}.$$

To utilize fully the capacity of BT_b, the used capacity $\mathrm{Cap\_}B(\alpha, \overline{h})$ should equal the available capacity AT_b. Let $\lambda_b(\overline{h})$ represent the number of wafer lots that should be released to the fab to utilize BT_b fully. Then, $\lambda_b(\overline{h})$ can be determined as

$$\lambda_b(\overline{h}) = \frac{\mathrm{AT}_b}{\sum_{i=0}^{L-1} tb_{i+1} \cdot \frac{L(\overline{W_i})}{n}}.$$

$\lambda(\overline{h})$, the maximum number of wafer lots that must be released to the fab to utilize the fab capacity fully, is then determined as

$$\lambda(\overline{h}) = \mathrm{Min}\left\{\lambda_s(\overline{h}), \lambda_b(\overline{h})\right\}.$$

The manufacturing costs of a wafer can be classified into two types: variable and fixed. A fixed cost item is a constant over the time horizon, independent of how many wafers are produced. The variable costs (here called *processing cost*) are unit based. That is, the processing cost per unit is a constant. For a *series-type* machine, the processing cost *per wafer* is a constant; while for a *batch-type* machine, the processing cost *per lot* is a constant. Therefore, the *processing cost per lot* is higher for a lot of more wafers. That is, c_k^i monotonically increases with k.

In the simplified fab, suppose the cost per raw wafer is \$100, and at layer 1 the processing cost *per lot* for the batch-type machines is \$50; the processing cost *per wafer* for the series-type machines is \$10. Then, $\overline{C_0}$ and $\overline{C_1}$ can be expressed as below, where $c_2^1 = \$70$ includes two components, \$50 for using the batch-type workstation to process a lot and \$20 for using the series workstation to process two wafers. $c_2^0 = \$200$ is the cost of two raw wafers

$$\overline{C_0} = \begin{bmatrix} \$0 \\ \$100 \\ \$200 \\ \$300 \end{bmatrix}, \quad \overline{C_1} = \begin{bmatrix} \$50 \\ \$60 \\ \$70 \\ \$80 \end{bmatrix}.$$

The total variable cost (TVC) of manufacturing and the total revenue (TR) associated with all the wafers released to the fab are expressed below. The first term in TVC represents the raw wafer cost, and the second item is the processing cost. Notably, the input to layer i is $\overline{W_{i-1}}$. In TR, $S(\overline{W_L})$ represents the total number of output wafers, and P represents the price

$$\mathrm{TVC} = \overline{W_0} \times \overline{C_0} + \sum_{i=1}^{L} \overline{W_{i-1}} \times \overline{C_i}$$
$$\mathrm{TR} = P \cdot S(\overline{W_L}).$$

D. Mathematical Formulation

Following the preceding discussion, the decision problem of scrapping small lots can be mathematically formulated as follows. The objective function is the profit to be maximized, of which the first item is the total revenue; the second term is the total variable cost, and the third item FC represents the total fixed cost. Notably, $\overline{h}$ represents the decision variables; $S(\overline{W_L})$ and $\overline{W_i}$ both depend on $\overline{h}$.

$$\mathrm{Max}\ P \cdot S(\overline{W_L}) - \left(\overline{W_0} \times \overline{C_0} + \sum_{i=1}^{L} \overline{W_{i-1}} \times \overline{C_i}\right) - \mathrm{FC}$$

Subject to

$$\overline{U_i} = \overline{U_0} \times \prod_1^i (A_i \times R(h_i)) \tag{1}$$

$$S(\overline{U_i}) = \sum_{k=1}^{M} k \cdot u_k^i \tag{2}$$

$$L(\overline{U_i}) = \sum_{k=1}^{M} u_k^i \tag{3}$$

$$\lambda(\overline{h}) = Min\left\{ \frac{\mathrm{AT_s}}{\sum_{\mathrm{i=0}}^{\mathrm{L-1}} \mathrm{ts_i} \cdot \mathrm{S}(\overline{\mathrm{U_i}})}, \frac{\mathrm{AT}_b}{\sum_{i=0}^{L-1} tb_i \cdot \frac{L(\overline{U_i})}{n}} \right\} \tag{4}$$

$$\overline{W_i} = \lambda(\overline{h}) \cdot \overline{U_i} \tag{5}$$

$$S(\overline{W_L}) = \sum_{k=1}^{M} k \cdot w_k^L \tag{6}$$

$$h_i \geq h_j, \quad \text{if } i < j. \tag{7}$$

In the above equations, $\overline{U_0}$ refers to the scenario in which only one lot is released to the fab. In this *one-lot-released* scenario, $\overline{U_i}$ represents the output at layer i after the yield problem and scrapping rules are addressed; $S(\overline{U_i})$ represents the number of output wafers at layer i and $L(\overline{U_i})$ indicates the number of output lots at layer i. Equation (4) gives the maximum number of lots $\lambda(\overline{h})$ that must be released to the fab to utilize the capacity fully. The first term in (4) is the maximum number of lots that must be released to utilize BT_s fully, and the second term is the maximum number of lots that must be released to fully utilize BT_b. Equation (5) determines $\overline{W}_i$, the output at layer i when $\lambda(\overline{h})$ lots are released to the fab. Equation (6) determines $S(\overline{W_L})$, the total number of wafers output by the fab.

Equation (7) denotes that the scrapping threshold at an upstream layer should not be smaller than that at a downstream layer. Otherwise, it is an irrational decision. For example, a solution with $h_1 = 2$ and $h_2 = 4$ is invalid. Based on such scrapping rules, a small lot with three wafers will pass layer 1 but will be scrapped at layer 2. This implies that the processing of this lot at layer 2 is useless. That is, even if the yield of this lot at layer 2 is 100%, the lot should still be scrapped. Therefore, $h_i \geq h_j$ for $i < j$ is an dispensible constraint in the formulated problem.

E. Analysis of the Problem

The problem of interest is formulated as a nonlinear mathematical model in the above. The objective function is a quite complex nonlinear function, by carefully examining (1) and (4). For the cases with four critical layers, Section V will show that the objective function in the *discrete solution space* is *multimodal*. Moreover, the number of local maximum points of this objective function is *unpredictable*, ranging from 9 to 31 in five testing cases where each includes four low-yield layers. Observing these complex properties, we proposed the use of a genetic algorithm to solve the problem when the solution space is large. However, when the solution space is small (for example, including three critical layers or less), an exhaustive search is performed to solve the problem.

III. Genetic Algorithm

The techniques of the genetic-based algorithm (GA), first proposed in the 1970s [1], [6], [12], have been widely applied to various areas [8], [9]. Much literature has demonstrated that GAs represent powerful techniques for solving problems that involve a large space. GA techniques have been applied to semiconductor manufacturing applications, such as developing operation recipes [11], [19], scheduling [2], identifying potential deadlock set [20], and improving cluster tool performance [7]. However, GA techniques have not been used in solving the formulated scrap decision problem.

A GA is an iterative procedure that maintains a constant-sized population $P(t)$ of candidate solutions (*chromosomes*). During each iteration step t, called a *generation*, new chromosomes are generated by applying *genetic operators* to current chromosomes. Each existing and newly generated chromosome is evaluated for its *fitness value*, which denotes the quality of the solution that it represents. Based on these evaluations, a set of chromosomes is *selected* to form the new population $P(t+1)$. The procedure is repeated until the *termination conditions* are met.

To design a genetic algorithm, a method for representing a chromosome, the genetic operators, a fitness function, a selection strategy for generating a new population, and termination conditions must all be defined.

A. Representing a Chromosome and Generating Initial Population

A chromosome in the proposed GA is a decision vector for scrapping small lots ($\overline{h}$), which is represented by the following string of C positive integers, where h_i represents the threshold for scrapping small lots at the critical layer i and C represents the number of critical layers

$$\overline{h} = [h_1, h_2, \ldots, h_C].$$

The vector $\overline{h}$ is called a chromosome; h_i is called a *gene*. By considering the constraint in (7), a *valid* solution (chromosome) should be: $h_i \geq h_j$, if $i < j$. Let N_p be the total number of chromosomes in the population $P(t)$. The initial population $P(0)$ is created by randomly generating N_p *valid* chromosomes. That is, an integer representing h_1 is randomly selected from the interval $[0, M-1]$, and h_{i+1} $(1 \leq i \leq C-1)$ is randomly selected from the interval $[0, h_i]$.

B. Fitness Function

The purpose of defining a fitness function is to evaluate the quality of a solution. In the proposed GA, the objective function models the profit for each scrap decision and is chosen as the fitness function.

C. Crossover and Mutation Operators

Our GA defines two genetic operators, *crossover* and *mutation*, for creating new chromosomes. For the chromosomes in $P(t)$, we randomly select a pair of chromosomes (parents) and generate a random number r_1. If $r_1 < P_{\mathrm{cr}}$, then the crossover operation is performed on the pair, where P_{cr} is a predefined crossover probability.

A crossover operation proceeds as follows. A gene position (called the *crossover point*) is randomly selected. The segments to the right of the crossover point of a chromosome are exchanged with those in the other chromosome of the pair. Let the pair of chromosomes to be crossed be $X_1 = [3\,3 : 3\,2]$ and

$X_2 = [5\ 4 : 4\ 1]$. Suppose that a crossover point (:) has been chosen as indicated. The resulting two offspring chromosomes would be $Y_1 = [5\ 4 : 3\ 2]$ and $Y_2 = [3\ 3 : 4\ 1]$. Notice that an offspring chromosome that is invalid (e.g., Y_2) is discarded, because it does not meet the constraint in (7).

Let P represent the set including the pair of parent chromosomes, and S represent the set that includes the valid offspring chromosomes. We then generate a random number r_2. If $r_2 < P_{\mathrm{mu}}$ (a predefined mutation probability), a mutation operation is to be performed on a chromosome X selected as follows. If $S \neq \phi$, then X is the one randomly chosen from S. If $S = \phi$, then X is the one randomly chosen from P.

The mutation operation on the selected chromosome X proceeds as follows. A gene position (h_i) is randomly selected, and its value is replaced by an integer randomly chosen from the interval $[h_{i+1}, h_{i-1}]$. The rationale for choosing random mutation is to prevent the solution from being trapped to local optimal points. The crossover and mutation operations are repeatedly carried out until N_p new and valid chromosomes are created.

D. Selection Strategy

The chromosomes in population $P(t)$ and the new chromosomes created by crossover and mutation, totally $2N_p$ in number, are placed in a pool, which are sorted in descending order according to their fitness values. Let the sorted chromosomes be represented by $\overline{h_1}, \overline{h_2}, \ldots, \overline{h_{2N_p}}$. Then, N_p chromosomes are selected to form population $P(t+1)$ by the following procedure, where P_{su} denotes a predefined survival rate.

Step 1) Select the first ranking chromosome
$P(t+1) \leftarrow \overline{h_1}$
$n = 1$; /*counting the number of selected chromosomes*/
$k = 1$; /* the currently concerned chromosome*/
Step 2) Select the other $N_p - 1$ chromosomes
$k = k + 1$; /for the next concerned chromosome */
Generate a random number r
If $r < P_{\mathrm{su}}$ then $P(t+1) \leftarrow \overline{h_k}$ and $n = n + 1$, else continue.
If $n = N_p$ then stop, else go to Step 2).

E. Terminating Conditions

Population $P(t)$ is iteratively updated until the termination conditions are met. There are two termination conditions. First, the GA terminates when the best solution in $P(t)$ keeps the same for over N generations. Second, the GA is forcedly terminated when $t = Y$, where Y is a positive large integer.

IV. Assumptions Concerning Test Examples

Solutions of the decision problem for a semiconductor fab in various scenarios are analyzed. This section presents the assumptions made about the semiconductor fab. These assumptions are based on the data of a real fab in industry, with reasonable simplifications. Some important characteristics are analyzed using numerical examples. The method for estimating the yield matrix is also discussed.

TABLE I
Variable Processing Cost Per Lot at Layer i, $\overline{C_i} = [c_k^i]$

c_1^i = \$287	c_6^i = \$324	c_{11}^i = \$361	c_{16}^i = \$399	c_{21}^i = \$436
c_2^i = \$294	c_7^i = \$332	c_{12}^i = \$369	c_{17}^i = \$406	c_{22}^i = \$443
c_3^i = \$302	c_8^i = \$339	c_{13}^i = \$ 376	c_{18}^i = \$414	c_{23}^i = \$451
c_4^i = \$309	c_9^i = \$347	c_{14}^i = \$384	c_{19}^i = \$421	c_{24}^i = \$458
c_5^i = \$317	c_{10}^i = \$354	c_{15}^i = \$391	c_{20}^i = \$429	c_{25}^i = \$466

A. Assumptions and Analysis of Test Examples

In the examples, the semiconductor fab produces only one product. The time horizon for decision making is one month. The process route has 20 layers $(L = 20)$, and a full wafer lot includes 25 wafers $(M = 25)$. The monthly fixed cost is $\mathrm{FC} = \$24\mathrm{M}$, the largest fraction of which is machine depreciation. The raw wafer cost is \$81.6 per wafer, that is, $\overline{C_0} = [c_k^0]$ and $c_k^0 = \$81.6 \times k$. The variable processing cost per wafer is the same for each layer i, $\overline{C_i} = \overline{C_{i+1}}$ $(1 \le i \le L)$. Table I gives $\overline{C_i}$; the cost item (c_k^i) monotonically increases with the number of wafers (k) in a lot. The cost of processing a lot without a wafer is zero, $c_0^i = 0$.

In the fab, ts_i and tb_i are constants for each layer i. As stated, ts_i is the run time per wafer layer on BT_s. The unit of capacity of BT_s is a wafer layer because each of its operations is performed on individual wafers at each layer. In the following expression, the capacity of BT_s is 900 000 wafer layers (that is, $(\mathrm{AT}_s/ts_i) = 900\,000$), where $\sum_{i=0}^{L-1} S(\overline{U_i})$ denotes the total number of wafer layers processed by BT_s when only one lot is released to the fab; $S(\overline{U_i})$ represents the number of input wafers at layer $i+1$

$$\frac{\mathrm{AT_s}}{\sum_{\mathrm{i=0}}^{\mathrm{L-1}} \mathrm{ts_i} \cdot \mathrm{S(\overline{U_i})}} = \frac{\mathrm{AT}_s}{ts_i \sum_{i=0}^{L-1} S(\overline{U_i})} = \left(\frac{\mathrm{AT_s}}{\mathrm{ts_i}}\right) \cdot \frac{1}{\sum_{i=0}^{L-1} S(\overline{U_i})} = \frac{900\,000\ \mathrm{wafer-layers}}{\sum_{i=0}^{L-1} S(\overline{U_i})}.$$

In a very high-yield environment, if $\sum_{i=0}^{L-1} S(\overline{U_i}) = 500$ wafer layers (25 wafers $\times$ 20 layers), then 1800 lots (900 000/500) must be released to the fab to utilize BT_s fully. In contrast, if $\sum_{i=0}^{L-1} S(\overline{U_i}) = 300$ wafer layers in a low-yield environment, then 3000 (900 000/300) lots should be released to utilize BT_s fully.

The capacity unit of BT_b is a *batch layer* because each operation is performed on individual batches at each layer. In the example, a batch is assumed always to have six lots $(n = 6)$,

so the unit of capacity of BT_b is then a *lot layer*. In the following expression, the capacity of BT_b is 42 000 lot layers ($(n \cdot \mathrm{AT}_b)/ts_b = 42\,000$), where $\sum_{i=0}^{L-1} L(\overline{U_i})$ denotes the number of lot layers when only one lot is released to the fab; $L(\overline{U_i})$ represents the number of input lots at layer $i+1$

$$\frac{\mathrm{AT}_b}{\sum_{i=0}^{L-1} \frac{ts_b}{n} \cdot L(\overline{U_i})} = \left(\frac{n \cdot \mathrm{AT}_b}{ts_b}\right) \cdot \frac{1}{\sum_{i=0}^{L-1} L(\overline{U_i})} = \frac{42\,000 \text{ lot} - \text{layers}}{\sum_{i=0}^{L-1} L(\overline{U_i})}.$$

In the high-yield environment considered above, if $\sum_{i=0}^{L-1} L(\overline{U_i}) = 20$ lot layers (1 lot $\times$ 20 layers), then 2100 lots (42 000/20) must be released to utilize BT_b fully. In contrast, in the low-yield environment, if $\sum_{i=0}^{L-1} L(\overline{U_i}) = 15$ lot layers, then 2800 (42 000/15) lots must be released to utilize BT_b fully.

In the high-yield environment, BT_s can accommodate 1800 input lots while BT_b can accommodate 2100 lots. The fab can, therefore, accommodate up to 1800 lots and BT_s is the bottleneck of the fab. In the low-yield environment, BT_s can accommodate 3000 input lots while BT_b can only accommodate 2800 lots. Therefore, the fab can accommodate up to 2800 lots and BT_b is the bottleneck of the fab. In summary, the bottleneck of the fab is either BT_s or BT_b according to the distribution of small lot sizes in the fab, which is determined by the yield matrices and the scrapping rules.

B. Yield Matrix

Of the 20 layers of the process route, some have 100% yield and are called *noncritical layers*. The others, with low yield, are called *critical layers*. In the real world, the yield distribution at each layer can be obtained by collecting the data from the shop floor. In this example, the binomial distribution is assumed to estimate the yield matrix associated with a critical layer.

Let the average yield at a critical layer i be p. The method for estimating the yield matrix ($A_i = [a^i_{jk}]$) can be interpreted as tossing a coin. Processing a lot with j wafers at the critical layer is like tossing a coin j times, where the probability of obtaining a *head* in one toss is p. The probability of transforming a lot with j wafers into one with k wafers equals that of obtaining k heads in j tosses

$$a^i_{jk} = C^j_k p^k (1-p)^{j-k} \quad \text{for } j \geq k; \qquad = 0 \quad \text{for } j < k.$$

The yield matrix for the simplified fab described in Section II, in which a full lot includes three wafers, is described as follows, where the probability (a^i_{32}) of obtaining two quality wafers out of processing three is $3p^2(1-p)$:

$$A_i = \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1-p & p & 0 & 0 \\ (1-p)^2 & 2p(1-p) & p^2 & 0 \\ (1-p)^3 & 3p(1-p)^2 & 3p^2(1-p) & p^3 \end{bmatrix}.$$

V. Analysis and Comparison of Solutions

The solutions in three scenarios are analyzed and compared. Scenario *S1* concerns the fab with only one critical layer. Scenario *S2* concerns the fab with two critical layers. Scenario *S3* concerns the fab with four critical layers. In scenarios *S1* and *S2*, the exhaustive search method is implemented in finding $\overline{h}$, while in scenario S3, the proposed GA is used. The selling price per wafer in each scenario is $P = \$2898$.

Some interesting findings from the following solution analysis are first summarized here.

1) The proposed scrapping method outperforms both the sunk-cost method and the no-scrapping method when the critical layers are in the upstream. Yet, there may be no difference when the critical layers are in the downstream.
2) The fab bottleneck shifts between BT_b and BT_s, depending upon the location of critical layers (*CLs*). The fab bottleneck is BT_b when *CLs* are in the upstream and shifts to BT_s when *CLs* are in the downstream.
3) The revenue and profit tends to increase as the critical layers moves toward upstream.

A. Scenario S1

Three cases in scenario *S1* will be considered. In the first, no scrapping rules are applied such that $\overline{h} = \overline{0}$. In the second, the *sunk-cost method* is applied to scrap small lots. In the third, the scrap rule proposed herein is applied. Let CL refer to the critical layer in *S1*. The average yield at CL is $p = 40\%$. Tables II–IV show that the selling price per wafer in scenario *S1* is so high that the company still makes profit even with 40% average wafer yield.

Case 1: Applying No Scrap Rule in S1: Table II summarizes the results for scenario *S1* when no scrap rule is applied. For various CL_s, the fab bottleneck, the number of input lots, the revenue, the variable costs, and the profit are analyzed below.

First, the shift of the fab bottleneck is analyzed. Table II indicates that the fab bottleneck may be either BT_b or BT_s depending upon the location of CL. For $1 \leq CL \leq 15$, the batch-type workstation BT_b is the fab bottleneck because $\lambda(\overline{h}) = \lambda_b(\overline{h})$. For $16 \leq CL \leq 20$, the series-type workstation BT_s is the fab bottleneck because $\lambda(\overline{h}) = \lambda_s(\overline{h})$.

The shift of the fab bottleneck can be understood by comparing an example for which $\mathrm{CL} = 1$ with one for which $\mathrm{CL} = 20$. When $\mathrm{CL} = 1$, small lots are generated at the output of layer 1. The average number of wafers per lot at each layer is $S(\overline{U_i}) = 25$ for $i = 0$, and $S(\overline{U_i}) = 10$ (25 wafers $\times$ 40% yield) for $1 \leq i \leq 19$. Consequently, $\sum_{i=0}^{L-1} S(\overline{U_i}) = 215$ and $\lambda_s(\overline{h}) = 4186$ lots $= 900\,000/215$. Since each input lot at layer 1 is a full lot, $L(\overline{U_i}) = 1$ for $i = 0$. For $1 \leq i \leq 19$, $L(\overline{U_i}) = 1 - a^i_{25,0} = 1 - C^{25}_0 \cdot (0.4)^0 \cdot (0.6)^{25} = 1 - 2.84 \times 10^{-6} \cong 1$ Therefore, $\sum_{i=0}^{L-1} L(\overline{U_i}) = 20$ and $\lambda_b(\overline{h}) = 42\,000/20 = 2100$; $\lambda(\overline{h}) = \min\{\lambda_b(\overline{h}), \lambda_s(\overline{h})\} = 2100$ lots. The batch-type workstation BT_b is thus the bottleneck of the fab when $\mathrm{CL} = 1$.

When $\mathrm{CL} = 20$, small lots are generated at the output of layer 20. The average number of wafers per lot at each layer is $S(\overline{U_i}) = 10$ for $i = 20$ and $S(\overline{U_i}) = 25$ for $0 \leq i \leq 19$. Therefore, $\sum_{i=0}^{L-1} S(\overline{U_i}) = 500$ and

TABLE II
RESULTS OF SCENARIO S1 WITHOUT APPLYING SCRAP RULE

Critical Layer (CL)	$\lambda_s(\overline{h})$ (lots)	$\lambda_b(\overline{h})$ (lots)	$\lambda(\overline{h})$ (lots)	Input wafer	h_{CL+1}	Output Wafer	Revenue ($M)	Variable Cost ($M)	Fixed Cost ($M)	Profit ($M)
1	4186	2100	2100	52500	0	21000	60.9	19.4	24.0	17.5
2	3913	2100	2100	52500	0	21000	60.9	19.6	24.0	17.3
3	3673	2100	2100	52500	0	21000	60.9	19.8	24.0	17.0
4	3462	2100	2100	52500	0	21000	60.9	20.1	24.0	16.8
5	3273	2100	2100	52500	0	21000	60.9	20.3	24.0	16.6
6	3103	2100	2100	52500	0	21000	60.9	20.5	24.0	16.3
7	2951	2100	2100	52500	0	21000	60.9	20.8	24.0	16.1
8	2813	2100	2100	52500	0	21000	60.9	21.0	24.0	15.9
9	2687	2100	2100	52500	0	21000	60.9	21.2	24.0	15.6
10	2571	2100	2100	52500	0	21000	60.9	21.5	24.0	15.4
11	2466	2100	2100	52500	0	21000	60.9	21.7	24.0	15.2
12	2368	2100	2100	52500	0	21000	60.9	21.9	24.0	14.9
13	2278	2100	2100	52500	0	21000	60.9	22.2	24.0	14.7
14	2195	2100	2100	52500	0	21000	60.9	22.4	24.0	14.5
15	2118	2100	2100	52500	0	21000	60.9	22.7	24.0	14.2
16	2045	2100	2045	51136	0	20450	59.3	22.3	24.0	13.0
17	1978	2100	1978	49451	0	19780	57.3	21.8	24.0	11.6
18	1915	2100	1915	47872	0	19140	55.5	21.3	24.0	10.2
19	1856	2100	1856	46392	0	18550	53.8	20.8	24.0	8.9
20	1800	2100	1800	45000	0	18000	52.2	21.2	24.0	7.8

$\lambda_s(\overline{h}) = 1800$ lots (900 000/500). Now each lot is a full lot before layer 20 is reached, $L(\overline{U_i}) = 1$ for $0 \leq i \leq 19$. Therefore, $\sum_{i=0}^{L-1} L(\overline{U_i}) = 20$ and $\lambda_b(\overline{h}) = 2100$ lots; $\lambda(\overline{h}) = \min\{\lambda_b(\overline{h}), \lambda_s(\overline{h})\} = 1800$ lots. The series-type workstation BT_s is, therefore, the bottleneck of the fab.

Second, the number of input lots and the revenue are analyzed. Table II reveals that $\lambda_s(\overline{h})$ decreases as CL moves downstream. For example, $\lambda_s(\overline{h}) = 4186$ lots when $\mathrm{CL} = 1$ and $\lambda_s(\overline{h}) = 1800$ lots when $\mathrm{CL} = 20$. Each input lot on average produces ten output wafers for both $\mathrm{CL} = 1$ and $\mathrm{CL} = 20$. When $\mathrm{CL} = 1$, each input lot at layers 2–20 carries an average of only ten wafers. Consequently, more lots must be released to utilize fully the capacity of BT_s. Increasing the number of input lots increases revenue. Therefore, when the critical layer is an earlier one, the revenue is higher.

Third, the variable costs and profit are analyzed. Table II reveals that when BT_b is the bottleneck of the fab ($1 \leq CL \leq 15$), the variable costs increases as CL moves downstream. This property is analyzed as follows. The total number of input lots is the same for $1 \leq CL \leq 15$. When $\mathrm{CL} = 1$, the input lots at layer 1 are full lots; when $\mathrm{CL} = 2$, the input lots at layers 1 and 2 are full lots. The distributions of input lots ($\overline{W_{i-1}}$) for $3 \leq i \leq 20$ are the same for both $\mathrm{CL} = 1$ and $\mathrm{CL} = 2$. The variable costs when $\mathrm{CL} = 2$ are, therefore, higher than that when $\mathrm{CL} = 1$.

When BT_s is the bottleneck of the fab, the variable costs for $16 \leq CL \leq 20$ generally decline as CL moves downstream except when $\mathrm{CL} = 20$ for two reasons. First, the number of input lots decreases as CL moves downstream. Therefore, fewer lots are processed so variable costs decrease, essentially explaining why the variable costs at $\mathrm{CL} = 18$ exceeds those at $\mathrm{CL} = 19$. Second, as in the above analysis, the variable costs per lot increase as CL moves downstream, essentially explaining why the variable costs at $\mathrm{CL} = 20$ exceeds those at $\mathrm{CL} = 19$. Table II indicates that the profit falls as CL moves downstream, implying that revenue is the primary cause of the change in profit.

Case 2: Applying the Sunk-Cost Method In S1: The *sunk cost*of a small lot that faces a scrap decision at layer i is the associated processing costs that have already be spent, that is, the processing cost from layer 1 to $i - 1$. The sunk cost method has two important features. First, the sunk cost of a small lot is irrelevant to the decision of scrapping. Second, a small lot should not be scrapped if the revenue from its final output wafers exceeds the remaining variable costs. Otherwise, the lot should be scrapped.

This method is clarified by the following example in which $\mathrm{CL} = 1$. Suppose a lot K with only one wafer faces a scrapping decision at layer 2. Should lot K be scrapped or kept at layer 2? The number of remaining layers of lot K is 19 ($2 \leq i \leq 20$) and each has 100% yield. Lot K, if not scrapped, will carry only one wafer throughout the remaining layers. The final output will

TABLE III
RESULTS OF SCENARIO S1 BY APPLYING THE SUNK-COST METHOD

Critical Layer (CL)	$\lambda_s(\bar{h})$ (lots)	$\lambda_b(\bar{h})$ (lots)	$\lambda(\bar{h})$ (lot)	Input wafer	h_{CL+1}	Output wafer	Revenue ($M)	Variable Cost ($M)	Fixed Cost ($M)	Profit ($M)
1	4186	2100	2100	52500	1	21000	60.9	19.4	24	17.5
2	3913	2100	2100	52500	1	21000	60.9	19.4	24	17.3
3	3673	2100	2100	52500	1	21000	60.9	19.8	24	17.0
4	3462	2100	2100	52500	1	21000	60.9	20.1	24	16.8
5	3273	2100	2100	52500	1	21000	60.9	20.3	24	16.6
6	3103	2100	2100	52500	1	21000	60.9	20.5	24	16.3
7	2951	2100	2100	52500	1	21000	60.9	20.8	24	16.1
8	2813	2100	2100	52500	1	21000	60.9	21.0	24	15.9
9	2687	2100	2100	52500	1	21000	60.9	21.2	24	15.6
10	2571	2100	2100	52500	0	21000	60.9	21.5	24	15.4
11	2466	2100	2100	52500	0	21000	60.9	21.7	24	15.2
12	2368	2100	2100	52500	0	21000	60.9	21.9	24	14.9
13	2278	2100	2100	52500	0	21000	60.9	22.2	24	14.7
14	2195	2100	2100	52500	0	21000	60.9	22.4	24	14.5
15	2118	2100	2100	52500	0	21000	60.9	22.7	24	14.2
16	2045	2100	2045	51136	0	20450	59.3	22.3	24	13.0
17	1978	2100	1978	49451	0	19770	57.3	21.8	24	11.6
18	1915	2100	1915	47872	0	19140	55.5	21.3	24	10.2
19	1856	2100	1856	46392	0	18550	53.8	20.8	24	8.9
20	1800	2100	1800	45000	0	18000	52.2	20.4	24	7.8

TABLE IV
APPLY OPTIMUM SCRAP RULES TO SCENARIO S1

CL	$\lambda_s(\bar{h})$ (lots)	$\lambda_b(\bar{h})$ (lots)	$\lambda(\bar{h})$ (lots)	h_{CL+1}	Output wafer	TR ($M)	TVC ($M)	FC ($M)	Profit ($M)	Case 1 Profit ($M)	Profit Change (%)
1	5894	3520	3520	9	23650	68.6	22.9	24.0	21.7	17.5	23.8%
2	4605	2786	2786	8	22504	65.2	21.4	24.0	19.8	17.3	14.6%
3	3936	2415	2415	7	21833	63.3	20.8	24.0	18.5	17.0	8.8%
4	3676	2392	2394	7	21643	62.7	21.0	24.0	17.8	16.8	5.8%
5	3346	2223	2223	6	21332	61.8	20.7	24.0	17.1	16.6	3.4%
6	3165	2214	2214	6	21255	61.6	20.9	24.0	16.7	16.3	2.2%
7	2968	2141	2141	5	21112	61.2	20.9	24.0	16.3	16.1	1.1%
8	2827	2138	2138	5	21083	61.1	21.1	24.0	16.0	15.9	0.7%
9	2690	2111	2111	4	21026	60.9	21.3	24.0	15.7	15.6	0.3%
10	2574	2110	2110	4	21016	60.9	21.5	24.0	15.4	15.4	0.1%
11	2466	2102	2102	3	21006	60.9	21.7	24.0	15.2	15.2	0.1%
12	2368	2100	2100	0	21000	60.9	21.9	24.0	14.9	14.9	0.0%
13	2278	2100	2100	0	21000	60.9	22.2	24.0	14.7	14.7	0.0%
14	2195	2100	2100	0	21000	60.9	22.4	24.0	14.5	14.5	0.0%
15	2118	2100	2100	0	21000	60.9	22.7	24.0	14.2	14.2	0.0%
16	2045	2100	2045	0	20450	59.3	22.3	24.0	13.0	13.0	0.0%
17	1978	2100	1978	0	19780	57.3	21.8	24.0	11.6	11.6	0.0%
18	1915	2100	1915	0	19140	55.5	21.3	24.0	10.2	10.2	0.0%
19	1856	2100	1856	0	18550	53.8	20.8	24.0	8.9	8.9	0.0%
20	1800	2100	1800	0	18000	52.2	21.2	24.0	7.0	7.0	0.0%

be one wafer and the revenue will be \$2898. The remaining variable costs are $c_1^i \times 19 = \$287 \times 19 = \$5453$. Lot K should be scrapped because its final revenue is less than the remaining variable costs. Alternatively, suppose a lot H carrying two wafers also faces a scrapping decision at layer 2. The final output will be two wafers with an associated revenue of $\$2898 \times 2 = \$5796$. The remaining variable costs are $c_2^i \times 19 = \$294 \times 19 = \$5586$. Lot H, therefore, should not be scrapped because its final revenue exceeds the remaining variable costs.

Scenario *S1* involves only one critical layer. Therefore, only a threshold at layer $\mathrm{CL}+1$ needs to be defined to establish the decision vector $\overline{h} = [h_i]$. That is, $h_i = 0$ for $i \neq \mathrm{CL}+1$. Table III indicates that the threshold $h_{\mathrm{CL}+1} = 1$ for $1 \leq \mathrm{CL} \leq 8$ and $h_{\mathrm{CL}+1} = 0$ for $9 \leq \mathrm{CL} \leq 19$. Since $h_{\mathrm{CL}+1}$ is either 1 or 0, the results of applying the sunk cost method (Table III) and those obtained by applying no scrapping rule (Table II) are quite similar. In a very high price scenario, the sunk-cost method tends to yield results close to those obtained by the no-scrapping method because the revenue of a lot with one wafer always exceeds the remaining costs. Tables III and II reveal exactly the same results because the number of small lots with only one wafer is very low, only around 0.01 lots $(2100 \times a_{25,1}^i = 2100 \times C_1^{25} \cdot (0.4)^1 \cdot (0.6)^{24} = 0.01)$.

Case 3: Applying the Optimum Scrapping Rule in S1: In Case 3, an exhaustive search is performed to identify the optimum threshold $(h_{\mathrm{CL}+1})$ to be applied in *S1* for $1 \leq \mathrm{CL} \leq 18$. With reference to Table IV, when $\mathrm{CL} = 1$, the bottleneck of the fab is BT_b. The optimum scrapping threshold is $h_{\mathrm{CL}+1} = h_2 = 9$ wafers; that is, a lot with nine or fewer wafers input to layer 2 should be scrapped. The number of input lots is 3513, around $67\% = (3513 - 2100)/2100$ higher than that in Case 1 (Table II). Each input lot on average produces $6.72 = 23\,650/3520$ wafers, about 33% fewer than in Case 1. With more input lots, the total variable cost in Case 3 is a little higher than in Case 1. Consequently, the profit in Case 3 greatly exceeds that in Case 1, by about $24\% = (21.7 - 17.5)/17.5$. The last column in Table IV shows the percentage difference between the profit in Case 3 and that in Case 1.

Scrapping small lots enables more new wafer lots to be fabricated and has two effects. First, it might increase the output of wafers and, therefore, revenue. Second, it might increase the total variable cost because more wafers are processed. The scrapping threshold is selected to optimize the trade off between the two effects.

When the critical layer is a downstream layer, the optimum decision tends to be not to scrap. The scrap threshold in Tables II–IV is 0 for $12 \leq \mathrm{CL} \leq 20$. When $\mathrm{CL} = 19$, each lot processed between layer 1 and 19 is a full lot; a decision to scrap is made at layer 20. Scrapping a small lot at layer 20 provides very little space to release new lots. Suppose a lot K carries one wafer and faces a decision to scrap at layer 20. If lot K is scrapped, BT_s releases capacity of 1.0 *wafer layer* to fabricate new lots. This released capacity provides space for around $0.002 = 1/(19 layers \times 25 \text{ wafers})$ new lots. With a 40% yield at layer 19, the newly released lots will produce approximately 0.002 $(0.002 \times 40\% \times 25$ wafers) output wafers. Furthermore, the new release increases the variable costs by $\$17(0.002 \times \$466 \times 19)$. Conversely, when not scrapped, lot K produces 1.0 output wafer. Lot K should, therefore, not be scrapped.

TABLE V
RESULTS OF SCENARIO 2

CL1	CL2	TR ($M)	TVC ($M)	Profit ($M)	Profit Diff ($M)	Profit Diff %	h_{CL1+1}	h_{CL2+1}
2	3	$63.6	$20.8	$18.8	$1.6	8.5%	12	7
2	6	$62.4	$20.7	$17.7	$0.8	4.5%	12	6
3	4	$62.8	$20.9	$18.0	$1.0	5.7%	11	7
3	6	$61.9	$20.6	$17.3	$0.5	3.0%	11	6
4	5	$61.9	$20.6	$17.3	$0.6	3.5%	10	6
4	6	$61.7	$20.7	$17.0	$0.4	2.4%	10	6
5	6	$61.6	$20.8	$16.8	$0.4	2.1%	6	6
6	7	$61.2	$20.8	$16.4	$0.2	1.2%	9	5
8	9	$61.0	$21.1	$15.8	$0.1	0.4%	6	4
10	11	$60.9	$21.6	$15.3	$0.0	0.1%	3	3
12	13	$60.9	$22.0	$14.8	$0.0	0.0%	0	0

B. Scenario S2

Scenario *S2* refers to the fab with two critical layers, *CL1* and *CL2*. The average yield is 63.2% at both *CL1* and *CL2* so the average yield of the fab is 40% $(0.4 = 0.632^2)$. The exhaustive search method is used to determine the optimum scrap rules.

Table V summarizes the results obtained under various combinations of *CL1* and *CL2* in scenario *S2*, when the proposed scrap decision is made. The profit in *S2* when the proposed scrapping rule applied is higher than that obtained when no scrapping rule is applied. Columns six and seven in Table V present the actual and fractional difference in profit between the two cases.

Let *Case(S2, CL1, CL2)* refer to a case in scenario *S2*. Table V shows that the profit declines as the two critical layers move farther apart. For example, the profit in *Case(S2, 2, 3)* exceeds that in *Case(S2, 2, 6)*. Tables IV and V reveal that the profit in *Case(S2, 2, 3)* is less than that in *S1* when $\mathrm{CL} = 2$, implying that the yield problem had better be identified as early as possible.

Notice that *Case(S2, CL3, CL4)* is a case which attempts to model *a product produced by a real fab in 2000*, at that time the price is pretty high and the yield is quit low due to introducing new process technology for the product. Table V reveals that the profit margin can increase about 5.7% by applying the proposed scrapping method.

C. Scenario S3

Scenario *S3* concerns the fab with four critical layers, *CL1*, *CL2*, *CL3*, and *CL4*. The average yield at each critical layer is 79.4% so the average yield of the fab is 40% (i.e., $0.4 = 0.794^4$). The proposed GA, coded in EXCEL with built-in VBA [21], is used to find the optimum scrap rules. A personal computer with 2.4-GHz CPU is used to run the GA program. Table VI shows the solutions obtained by the GA in five testing cases of scenario *S3*. The parameters of the GA are set as follows: $P_{\mathrm{cr}} = 0.7$, $P_{\mathrm{mu}} = 0.1$, $P_{\mathrm{su}} = 0.7$, $N = 30$, and $Y = 500$. The determination of these parameters is by experiment. Some other parameter settings have been tried and all produce the same final solution. Table VII shows the numbers of generations at the termination of the GA for these parameter settings.

TABLE VI
SOLUTIONS OBTAINED BY THE PROPOSED GA IN FIVE CASES OF SCENARIO *S3* AND THE NUMBER OF LOCAL MAXIMUM POINTS (*NUMBER OF LMP*) IN EACH SOLUTION SPACE

	CL1	CL2	CL3	CL4	TR ($M)	TVC ($M)	Profit ($M)	Profit Diff ($M)	h_{CL1+1}	h_{CL2+1}	h_{CL3+1}	h_{CL4+1}	*No. of L.M.P*
Case1	1	3	5	7	$62.6	$21.0	$17.6	$0.6796	17	12	8	5	9
Case2	2	4	6	8	$61.6	$20.7	$17.0	$0.3185	16	11	7	5	13
Case3	3	5	7	9	$61.2	$20.6	$16.6	$0.1374	14	10	7	4	12
Case4	4	6	8	10	$61.0	$20.7	$16.3	$0.0676	14	9	6	4	31
Case5	5	7	9	10	$61.0	$20.9	$16.1	$0.0434	11	8	5	4	29

TABLE VII
NUMBER OF GENERATIONS WHEN GA TERMINATED (N_t) IN DIFFERENT PARAMETER SETTINGS

P_{cr}	P_{mu}	P_{su}	N_t
0.4	0.1	0.7	43
0.6	0.1	0.7	38
0.7	0.1	0.7	38
0.8	0.1	0.7	43
0.7	0.05	0.7	44
0.7	0.1	0.7	38
0.7	0.4	0.7	39
0.7	0.6	0.7	42
0.7	0.1	0.6	68
0.7	0.1	0.65	52
0.7	0.1	0.75	48
0.7	0.1	0.8	48

For benchmarking, the exhaustive search method is also performed. Notice that the exhaustive search excludes in advance the chromosomes that do not meet the constraint set in (7). This exclusion greatly reduces the computation time for space search. In each case of Table VI, the solution obtained by the GA and that obtained by the exhaustive search method is exactly the same. As known, GAs inherently only ensure a near-optimum or local-optimum solution. However, the testing results show that the proposed GA seems to be a very good method in solving the formulated problem. In each case, the GA takes about 3 min in computation and the exhaustive search method takes about 30 min. To justify the reliability of the GA, we run each case 20 times with different random seeds for generating initial populations. The 20 obtained solutions are also exactly the same. Fig. 2 shows the evolution process of the 20 runs in testing case 1.

From the data obtained by the exhaustive search, we find that the objective function in the solution space is multimodal. The last column in Table VI shows the number and location of local maximum points in each case, which ranges from 9 to 31. Table VIII presents the distribution of the local maximum points in Case 1. The complex multimodal property of the objective function implies that GA is a good approach to solve the formulated problem. A sensitivity analysis for price changing is also

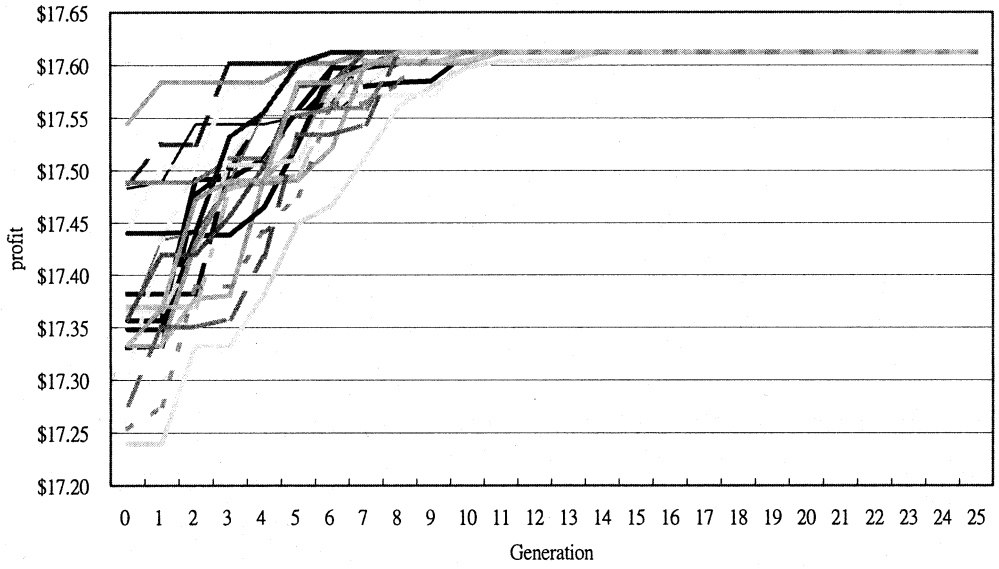


Fig. 2. Evolution process of 20 runs of case 1 in Scenario 3 (money unit: million U.S. dollars).

TABLE VIII
NINE LOCAL MAXIMUM POINTS IN CASE 1 OF SCENARIO S3

No.	h_{CL1+1}	h_{CL2+1}	h_{CL3+1}	h_{CL4+1}	Profit ($M)
1	11	11	8	5	$17.39
2	12	3	3	0	$16.94
3	12	11	8	0	$17.34
4	12	12	5	5	$17.36
5	17	8	8	5	$17.56
6	17	12	5	0	$17.49
7	17	12	5	5	$17.58
8	17	12	8	0	$17.55
9	17	12	8	5	$17.61

examined. Table IX shows that the scrapping threshold tends to increase when the price is getting higher. Moreover, the scrapping decision is relatively stable when the price does not substantially change. In the low-yield and high-price scenario, the impact of cost change is quite small. Its sensitivity analysis is, therefore, not presented here.

Our experiments show that the exhaustive search takes about 1 min of computation for a case including three critical layers. Therefore, we suggest the use of the exhaustive search method when the number of critical layers is less than or equal to 3; otherwise, the GA is suggested.

Notice that the profit in the first row of Table IV considerably exceeds that of Table VI. The first row in Table IV concerns a fab with layer 1 as the only one critical layer and its yield is 40%. The first row in Table VI concerns a fab with four critical layers, layers 1, 3, 5, and 7; each has a yield 79.4%, so the average yield of the fab is also 40%. This property implies that moving low yield layers upstream tend to increase the profit. This finding can also be confirmed by comparing the seventh row of Table IV (layer 7 is the only critical layer with 40% yield) and the first row of Table VI.

TABLE IX
SCRAPPING THRESHOLDS AND PROFITS IN DIFFERENT AVERAGE SELLING PRICES

Price ($)	h_{CL1+1}	h_{CL2+1}	h_{CL3+1}	h_{CL4+1}	Profit ($M)
$2,609	16	11	8	5	$11.38
$2,754	16	11	8	5	$14.48
$2,761	16	11	8	5	$14.64
$2,762	16	11	8	5	$14.67
$2,768	17	12	8	5	$14.79
$2,783	17	12	8	5	$15.11
$2,812	17	12	8	5	$15.73
$2,841	17	12	8	5	$16.36
$2,899	17	12	8	5	$17.61
$2,928	17	12	8	5	$18.24
$3,043	17	12	8	5	$20.74
$3,188	17	12	8	5	$23.87
$4,348	17	12	8	6	$48.95
$4,638	17	12	8	6	$55.23
$4,928	17	12	8	6	$61.50
$5,072	17	12	8	6	$64.63
$5,077	17	12	8	6	$64.73
$5,078	17	12	8	6	$64.75
$5,087	18	12	8	6	$64.95
$5,101	18	12	8	6	$65.26
$5,217	18	12	8	6	$67.79
$5,797	18	12	8	6	$80.41

VI. CONCLUDING REMARKS

This study formulates a model for solving the decision-making problem concerning the scrapping of small lots in semiconductor wafer fabs. This problem is very important, especially in a low-yield and high-price scenario. Such a scenario occurs quite often, especially when a new product

using advanced processes or technologies is just being introduced to the market. The demand is high, yet the yield of the product is very low. Scrapping small lots appropriately in such a scenario can increase profit, but this idea has been rarely considered in previous literature. This paper may be the first to mathematically formulate the problem.

When the number of low-yield layers is less than or equals three, the exhaustive search method is suggested to solve the formulated problem. Otherwise, the proposed GA is suggested. The GA inherently ensures only a near-optimum solution. However, in each of our five testing cases, the GA always yields the optimum solution. The proposed scrapping method considerably outperforms both the sunk-cost method and the no-scrapping method when the critical layers are in the upstream. Yet, there may be no difference when the critical layers are in the downstream.

Solutions of the numerical examples reveal the following two interesting phenomena concerning a low yield fab. First, given a single process route, the bottleneck of the fab may switch between a series-type workstation and a batch-type workstation. Suppose that the bottleneck of a fab is a series-type workstation. Low yield at a downstream layer provides very few opportunities for releasing new wafer lots; the series-type workstation thus remains the bottleneck. In contrast, low yield at an upstream layer provide the opportunity to release more wafer lots. The number of input lots can only be increased up to the capacity of the batch-type workstation. A batch-type workstation might consequently become the fab bottleneck.

Second, the difference between the profit obtained by scrapping and that obtained without scrapping is substantial when the low yield layers are upstream. The difference is less significant when the low yield layers are downstream. This finding implies that a low yield at downstream layers will cause most of the early-used capacity useless. Developing effective means of increasing the yield of the downstream layers is, therefore, crucial, even at the cost of reducing the yield upstream.

ACKNOWLEDGMENT

The authors acknowledge suggestions made by the reviewers.

REFERENCES

[1] A. D. Bethke, "Genetic algorithm as functions optimizers," Ph.D. dissertation, Dept. Comput. Commun. Sci., Univ. Michigan, Ann Arbor, 1981.

[2] J. H. Chen, L. C. Fu, M. H. Lin, and A. C. Hunag, "Petri-net and GA-based approach to modeling, scheduling, and performance evaluation for wafer fabrication," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 619–636, May 2001.

[3] R. Cooper and R. S. Kaplan, "Measure cost right: Make the right decision," *Harvard Bus. Rev.*, pp. 96–123, Sept./Oct. 1988.

[4] K. Daigle and R. Powell, "Manufacturing scrap reduction team," in *Proc. IEEE/SEMI Advanced Manufacturing Conf.*, 1996, pp. 230–231.

[5] D. Dance and R. Jarvis, "Using yield models to accelerate learning curve progress," *IEEE Trans. Semiconduct. Manuf.*, vol. 5, pp. 41–46, Feb. 1992.

[6] K. A. DeJong, "Analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Dept. Computer and Communication Sciences, Univ. Michigan, Ann Arbor, 1975.

[7] M. A. Dümmler, "Using simulation and genetic algorithms to improve cluster tool performance," in *Proc. 1999 Winter Simulation Conf.*, 1999, pp. 875–879.

[8] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*. New York: Wiley, 2000.

[9] D. E. Glodberg, *Genetic Algorithms in Search Optimization & Machine Learning*. New York: Addison Wesley, 1989.

[10] E. Goldratt and R. Fox, *The Race*. Croton-on-Hudson, NY: North River, 1986.

[11] S. S. Han and G. S. May, "Using neural network process models to perform PECVD silicon dioxide recipe synthesis vai genetic algorithms," *IEEE Trans. Semiconduct. Manuf.*, vol. 10, pp. 279–287, May 1997.

[12] J. H. Holland, *Adaptation in Neural and Artificial Systems*. Ann Arbor: Univ. Michigan Press, 1975.

[13] R. Kee and C. Schmidt, "A comparative analysis of utilizing activity-based costing and the theory of constraints for making product-mix decisions," *Int. J. Prod. Econ.*, pp. 1–17, 2000.

[14] W. Kuo and T. Kim, "An overview of manufacturing yield and reliabilty modeling for semiconductor products," *Proc. IEEE*, pp. 1329–1344, Aug. 1999.

[15] B. Lee and B. Bowhill, "Accounting for manufacturing: identifying the links between markets, production and costing systems," *Eng. Manage. J.*, pp. 182–188, 1997.

[16] D. N. Maynard, D. S. Kerr, and C. Whiteside, "Determining cost of yield to monitor fab manufacturing processes" (in http://www.micromagazine.com/archive/03/06/maynard.html), *Micromagazine.com*, pp. 63–69, June 2003 .

[17] ——, "Cost of yield," in *Proc. IEEE/SEMI Advanced Manufacturing Conf.*, 2003, pp. 165–170.

[18] R. N. Nurani, R. Akella, and A. Strojwas, "In-line defect sampling methodology in yield management: an integrated framework," *IEEE Trans. Semiconduct. Manuf.*, vol. 9, pp. 506–517, Nov. 1996.

[19] E. A. Rietman and R. C. Frye, "A genetic algorithm for low variance control in semiconductor device manufacturing: some early results," *IEEE Trans. Semiconduct. Manuf.*, vol. 9, pp. 223–229, May 1996.

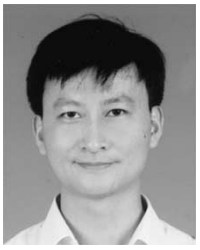
[20] H. Y. Yoon and D. Y. Lee, "Identification of potential deadlock set in semiconductor track systems," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2001, pp. 1820–1825.

[21] E. Wells, *Microsoft EXCEL 97 Developer's Handbook*. Redmond, WA : Microsoft Press, 1997.

[22] M. C. Wu, C. W. Chiou, and H. M. Hsu, "Scrap rules for small lots in wafer fabrication," in *Proc. 2002 Semiconductor Manufacturing Technology Workshop*, Hsin-Chu, Taiwan, R.O.C., 2002, pp. 181–184.

Muh-Cherng Wu received the B.S. degree from National Chiao Tung University, Hsin-Chu, Taiwan, R.O.C., in 1977 and the MBA degree from National Chen-Chi University, Taipei, Taiwan, in 1979. He received the M.S. and Ph.D. degrees in industrial engineering from Purdue University, West Lafayette, IN, in 1985 and 1988, respectively.

He is a Professor in the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan. His research interests include production management, supply chain management, and computer integrated manufacturing.



Chie-Wun Chiou received the B.S. degree in industrial engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C., and the M.S. degree in industrial engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C. He is working toward the Ph.D. degree at National Chiao Tung University.

He has been working in the semiconductor industry for over eight years and possesses several U.S. patents in the semiconductor field.

Hsi-Mei Hsu received the B.S. degree in industrial management from National Chen-Kung University, Tainan, Taiwan, R.O.C., in 1970, the MBA degree from Osaka University, Osaka, Japan, in 1975, and the Ph.D. degree in industrial engineering from National Tsing-Hua University, Hsinchu, Taiwan, in 1991.

She is a Professor in the Department of Industrial Engineering and Management, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. Her research interests include multiple criteria decision making, performance measurement, supply chain management, and production management.