行政院國家科學委員會專題研究計畫　期中進度報告

序的最佳化方法的改進(1/2)

計畫類別：個別型計畫
計畫編號：NSC91-2213-E-009-070-
執行期間：91年08月01日至92年07月31日
執行單位：國立交通大學電機與控制工程研究所

計畫主持人：林心宇

計畫參與人員：洪士程，黃榮壽，張紹興，林啟新，張文賢，陳亮元

報告類型：精簡報告

處理方式：本計畫涉及專利或其他智慧財產權，2年後可公開查詢

中　華　民　國 92年5月27日

計畫中文摘要

哈佛大學何毓琦教授於91年元月初在交大開設為期三天的短期課程，其中一個最重要的主題Ordinal Optimization引起參與此課程的各校教授熱烈的迴響，主因是 Ordinal Optimization 是一個應用範圍非常廣的研究主題，而且它對一些以往被認為不能解的最佳化問題提供了一個相當不錯的解答。

鑑於 Ordinal Optimization 應用範圍的廣大，本研究計劃針對現有的Ordinal Optimization 的方法做更進一步的改進。我們所擬定的改進目標有二：(i)如何選取較好的 N 個樣本解。(ii)提供一個新的解 Constrained Ordinal Optimization 問題的方法。

對此二大問題，我們將以二年的時間提出完整的解答。第一年我們將針對第(i)個主題提出解決的方法。第(i)個主題基本上可以分成兩個層次。即有結構資訊的系統及無結構資訊的系統的困難最佳化問題。我們將在第一年計劃針對有結構資訊的例子系統的困難最佳化問題提出一個找尋較好的 N個樣本解的方法。並進而求得較目前的Ordinal Optimization方法更好的解。而在第二年計劃中針對無結構資訊的系統的困難最佳化問題提出一個結合智慧型計算與 Ordinal Optimization 的演算法來找尋較好的 N 個樣本解並求得更好的解。同時我們也將針對 Constrained Ordinal Optimization 的問題提出一個具體可行的解決方法及其理論上的支持。

Abstract

In this project, we intend to propose methods for the following two significant subjects in Ordinal Optimization Theory: (i) Finding the better N samples and (ii) Constrained Ordinal Optimization Problems.

In the first subject, we consider two types of hard optimization problems. The first type is for systems with structural information and the second one is for the lack of structural information system. Thus, we intend to use two years to deal with these two subjects. In the first year, we will consider an example system that we are familiar with and propose a heuristic method to find the better N samples and then to find a good enough solution, which will be better than that can be found in the current Ordinal Optimization method. In the second year, we will combine the Ordinal Optimization method with an intelligent computing method to find the better N samples for a lack of structural information system's hard optimization problems and then to find a better good enough solution. In the meantimes, we will also propose a new method to solve constrained ordinal optimization problem and provide rigorous theoretical support for the proposed method.

一、前言

序的最佳化方法（Ordinal Optimization）是近年來由哈佛大學何毓琦教授所提出的一個解困難最佳化問題 （hard optimization problem）的快速且有效的方法。由於這個方法的提出使得一些向來被認為無法解的 NP hard 的問題可以重新檢視其可解性。本人於2001年5月至8月期間承蒙國科會資助至哈佛大學進行三個月的訪問，並在此期間與何教授共同研究序的最佳化相關問題，且共同發表了一篇文章[1]。由於這段期間的研習，本人不但

對序的最佳化方法有深入的瞭解，且有感於此方法無窮的應用潛力，乃提出此改進序的最佳化方法的計畫。

二、研究目的

由於目前序的最佳化方法仍有一些可以改進的地方，例如：1.它所謂的 top 5% 的解在具有龐大的樣本解空間的問題中，這樣求得的解在實際最佳化技巧運用者的眼光中稱不上是很好的解。以及 2.它還不能解有限制式的困難最佳化問題，尤其是具有等式限制式的。所以我們這個計畫的研究目的，便是要克服序的最佳化方法的這兩個缺點。

我們第一年的研究計劃及是針對第 1 個缺點來尋求改進的方法，亦即提出一個尋找較好的 N 個樣本解的方法。而且我們將對兩種類型的問題，即有結構資訊及無結構資訊的系統的兩種類型來提出方法。

針對有結構資訊的系統，我們擬以電力系統中的電容裝置問題為例，因為它便是具有等式與不等式限制的有離散控制變數（discrete control variables）的困難最佳化問題，針對無結構資訊的系統，我們擬以半導體製程的晶圓測試程序中如何訂定 lot， wafer，及 bin 的門檻值（threshold value）以減少誤宰（overkills）及重測（retests）的問題為例，因其所有可能的門檻值的組合數目超過 10^{30}，所以是一個困難最佳化問題。

三、文獻探討

困難的最佳化問題在實際的世界中隨處可見，例如有離散變數的最佳化問題(optimization problem with discrete control variables)中的最佳化電力潮流(optimal power flow）問題[2-4]是最為電力工程界的研究人員所熟悉。又如隨機最佳化問題（stochastic optimization problem)[5] 更是在離散事件中動態系統(discrete event dynamic systems)[6] 中俯拾即是的問題。近年來，此類型最佳化問題，概皆以 simulated annealing 法[7]，基因演算法[8]，或 tabu search 法[9]來求解，但這些方法的計算皆非常耗時而不適合實際應用。不同於這些以全域搜尋技術（global searching technique) 來解困難最佳化問題的方法，序的最佳化方法提出了一個可迅速地求得 top n% 解的方法。何教授及其研究團隊提出了相當多的研究論文[10-11]來闡明此方法的優點：即它不像一般的 heuristic 法，它是一個可以提供 quantitative result 的方法。整個序的最佳化方法的演算步驟的理論基礎是在 Universal Alignment Probability [11]這篇文章中，它提供了如何評量所提出的 rough model 的 roughuess 以及如何選取 Selected subset 的樣本數，這些在實際的應用中非常地有用，有了這篇文章的基本概念後，對實際應用問題的序的最佳化方法的推導可說是事半功倍。

四、研究方法

我們針對我們所擬改進的序的最佳化方法的部分，挑選兩類具有實際應用價值的問題來研究並設計求解的方法。

第一類問題是具有結構資訊的有離散變數的最佳化問題，在這類問題中我們挑選了電力系統的電容配置問題來作為我們提出一個改進現行序的最佳化方法的例子問題。在解此電容配置問題的同時，我們先求解具有離散控制變數的最佳電力潮流問題。我們針對最佳電力潮流問題所提出的改進序的最佳化方法是設計一個較佳的選擇 N 個樣本解的方法。此方法可簡述如下：

Step 1: 先將所有離散控制變數視為連續控制變數後，求解連續性（continuous）的最佳電力潮流問題。

Step 2: 根據各個離散控制變數的敏感度分析（sensitivity analysis）將對目標函數較不敏感的離散控制變數固定在最靠近其連續值解

的離散值，僅讓最敏感的前 10 個離散控制變數可彈性設為最靠近其連續值解之左邊或右邊之離散值。

Step 3: 根據 Step 2 所選出之 10 個離散控制變數所形成的 2^{10}=1024 個組合中，根據敏感度分析，挑選出前 50 名可能具有較低之目標函數的組合。

Step 4: 將離散變數分別固定在 Step 3 所選出之組合，然後解最佳電力潮流問題。於是在這 50 個最電力潮流問題中具有最低目標函數所對應的離散控制變數的組合即是我們要找的解。

第二類問題是無結構資訊而具有龐大樣本空間的隨機最佳化問題，我們挑選了一個非常具有應用價值的晶圓測試程序中如何訂定關鍵值（threshold values）以減少晶粒誤宰 (overkills) 及重測 (retests) 次數的問題。在這個問題中，我們結合了類神經網路及基因演算法來做為我們在超過 10^{30} 個樣本解中挑出較好的 N (=1000) 個樣本解的方法，此方法可大略敘述於下：

Step 1: 均勻地從龐大的樣本解空間中挑選 300 個樣本解，將每個樣本解根據晶圓測試程序作隨機模擬並求出其對應之誤宰及重測次數。

Step 2: 根據此 300 個 I/O mappings 來訓練一個類神經網路的 weights。

Step 3: 利用基因演算法從龐大的樣本解空間中找最好的前 1000 個解，而此基因演算法中對每一個樣本解的 fitness values 的計算即是利用 Step 2 中所建立之類神經網路來求得。

Step 4: 利用較短的隨機模擬，從 Step 3 得到的 1000 個樣本解中選出較佳的前 50 個解。

Step 5: 利用完整的隨機模擬，將 Step 4 得到的 50 個解中選出的解即為我們最後要找的解。

在這個研究中，我們不但改進了序的最佳化方法中如何找到更好的 N 個樣本解，我們也同時解了兩個創新且具應用價值的問題。

五、結果與討論

雖然此計畫僅進行到一半，我們卻已經有相當豐碩的成果。首先我們已成功地提出了一個以序的最佳化方法為基礎的解具有離散變數的最佳電力潮流問題的演算法則，我們這個方法的提出也獲得何教授的協助並已撰寫成論文投稿至 IEEE Trans. on Power Systems，目前正在審核的階段。此結果可概述如下：

與傳統的解法比較，我們的方法在目標函數值上平均降低 28%；同時，傳統的方法在很多情況下，無法得到可行解，而我們的方法每一次皆可得到可行解。與全域搜尋技巧的 tabu search 法比較，當 tabu search 法花了較我們的方法將近 100 倍的 CPU times 時，其目標函數值仍高於我們所得的目標函數值 18%。由此可見，我們的方法不但快速且可有效地得到一個不錯的解。

此外，我們在晶圓測試的訂定關鍵值以減少誤宰與重測問題上，也獲致很好的結果，以下為我們所得結果的大略描述：

我們以實際半導體製造公司中晶圓製造的數據來建立我們隨機模擬中 Poisson distribution 的模型，且以該公司之晶圓測試程序當作我們模擬的測試程序，根據我們在研究方法中所列出之步驟，我們求得減少誤宰及重測之關鍵值。我們同時將所求得之關鍵值與 1000 個隨機產生之關鍵值比較所對應之誤宰及重測，我們所得的值是所有的關鍵值中最好的。

此結果中所使用之數據係來自某大半導體製造公司及工研院朋友們的協助，我們已在今年 3 月 31 日在德國慕尼黑召開的先進半導體製造會議(ASMC 2003)中發表，論文如附件一所示，其中我們也註明了對國科會贊助本計劃研究的謝辭。

六、參考文獻

[1] Lin, S. and Ho, Y.C., ‘‘Universal alignment probability revisited,’’ JOTA, pp.299-407, May 2002.

[2] Liu, W., Papalexopoulos, A. and Tinney W., ‘‘Discrete shunt controls in a Newton optimal power flow,’’ *IEEE Trans. on Power Systems*, vol. 7, no. 4, pp. 1509-1520, Nov. 1992.

[3] Tinney, W., Bright, J., Demaree, K. and Hughes, B., ‘‘Some deficiencies in optimal power flow,’’ *IEEE PICA Conf. Proc.*, pp. 164-169, Montreal, May 1987.

[4] Bakirtzis, A. and Meliopoulos, A., ‘‘Incorporation of switching operation in power system corrective control computations,’’ *it IEEE Trans. on Power Systems*, vol. 2, no. 3, pp. 669-676, Aug.1987.

[5] Ho, Y.C., Cassandras, C.C., Chen, C.H., and Dai, L., ‘‘Ordinal optimization and simulation,’’ *Journal of Operation Research Society*, vol. 21, pp. 490-500, 2000.

[6] Cassandras, C. G. and Lafortuene, S., Introduction to discrete event systems, Kluner Academic Pull., Bostor, MA., 1999.

[7] Chen, L., Suzuki, H. and Katou, K., ‘‘Mean field theory for optimal power flow,’’ *IEEE Trans. on Power Systems*, vol. 12, no. 4, Nov. 1997.

[8] Ma, J. T. and Lai, L. L., ‘‘Evolutionary programming approach to reactive power planning,’’ *IEEE Proc., Generation, Transmission and Distribution*, vol. 143, no. 4, 1996.

[9] Kulworawanichpong, T. and Sujitjorn, S., ‘‘Optimal power flow using tabu search,’’ *IEEE Power Engineering Review*, pp. 37-40, June 2002.

[10] Ho, Y.C., ‘‘Soft optimization for hard problem,’’ Lecture notes, Harvard University, Cambridge, MA., 1996.

[11] Lau, T.W.E. and Ho, Y.C., ‘‘Universal alignment probability and subset selection for ordinal optimization,’’ *JOTA*, vol. 39, no. 3, June 1997.

七、計劃成果自評

本計劃已進行將近一年，所獲成果堪稱豐碩，總計已發表了一篇知名國際會議論文，以及一篇投稿知名期刊之論文，我們期望明年此時，能有更豐富的成果報告。

附件一

Reducing the Overkills and Retests in Wafer Testing Process

S. C. Horng [§] S. Y. Lin M. H. Cheng
Dept. of Elec. & Contr. Eng.
National Chiao Tung Univ.
[§] Dept. of Electronic Eng.
Chin Min College of Tech. & Comm.
1001, Ta Hsueh Rd., Hsinchu, Taiwan,
R.O.C.
sylin@cc.nctu.edu.tw

F. Y. Yang
Taiwan Semiconductor
Manufacturing Co.
121, Park Ave.3, Science-Based
Industrial Park, Hsinchu, Taiwan,
R.O.C.
fyyang@tsmc.com.tw

C. H. Liu W. Y. Lee C. H. Tsai
Industrial Technology Research Institute
Bldg.11,195-3 Chung Hsing Rd., Section
4, Chutung,Hsinchu,Taiwan,
R.O.C.
770690@itri.org.tw

Abstract

Reducing overkills is one of the main objectives in wafer testing process, however the major mean to prevent overkills is retest. In this paper, we formulate the problem of reducing overkills and retests as a stochastic optimization problem to determine optimal threshold values concerning the number of good dies and the number of bins in a lot and wafer to decide whether to go for a retest after a regular wafer probing.

The considered stochastic optimization problem is an NP hard problem. We propose an Ordinal Optimization theory based two-level method to solve the problem for good enough threshold values to achieve lesser overkills and retests within a reasonable computational time. Applying to a case based on the true mean of bins of a real semiconductor product, the threshold values we obtained are the best among 1000 sets of randomly generated threshold values in the sense of lesser overkills under a tolerable retest rate.

Keywords

Wafer testing, overkill, retest, stochastic optimization, ordinal optimization.

1. Introduction

The wafer fab process is a sequence of hundreds of different process steps, which results in an unavoidable variability accumulated from the small variations of each process step. Thus, to avoid incurring the significant expense of assembling and packaging chips that do not meet specifications, the wafer probing in the manufacturing process becomes an essential step to identify flaws early.

Wafer probing establishes a temporary electrical contact between test equipment and each individual chip on a wafer to determine the goodness of a chip. Although there exist techniques such as the SPC [1] for monitoring the operations of the wafer probes, the probing errors may still occur in many aspects and cause some good dies being over killed; consequently, the profit is diminished. Thus, reducing the number of overkills is always one of the main objectives in wafer testing process. The major mean for preventing overkills is retest. However, retest is an operation of high engineering cost and a major factor for decreasing the throughput. Thus, the overkill and the retest possess inherent conflicting factors, because reducing the former can gain more profit while increasing the latter will degrade the throughput and increase the cost. What implies is that drawing a fine line for deciding whether to go for a retest is an important research issue in the wafer testing process.

There may be different testing procedures in different chip manufacturers. But, no matter what testing procedures are used, the decision for carrying out the retest should be based on whether the number of good dies and the number of bins in a lot and wafer exceed the corresponding threshold values. Thus, determining these threshold values so as to minimize the overkills and retests is the main theme of our problem. Furthermore, since the goodness of a chip and the probing errors are of stochastic nature, our problem becomes a stochastic optimization problem, which in general is a simulation oriented NP hard problem. It is well-known that to obtain an optimal solution of an NP hard optimization problem is computationally intractable. To deal with this hard stochastic optimization problem, we propose in this paper an Ordinal Optimization (OO) theory [2] based two-level approach to solve for a good enough solution of the threshold values in the aspect of reducing overkills and retests.

2. Problem Statement and Mathematical Formulation

In this paper, we employ a typical testing procedures used in a semiconductor manufacturing company in Taiwan, which is briefly described in the following.

For every wafer in a lot, a wafer probing is performed twice. A die is considered to be good if it is good in either test. We let g_i and g_{ij} denote the number of good dies in lot i and wafer j of lot i, respectively, and let B_{ijk} denote the number of bin k in wafer j of lot i. Then, a three-stage checking on the number of good dies is performed to determine the necessity of carrying out a retest. We let $g_{L\min}$ and $g_{W\min}$ denote the threshold values of the number of good dies to pass or hold the lot and wafer, respectively; we let $n_{k\max}$, $k=1,...,K$, denote the threshold values of bin k in the hold wafer to determine whether to perform a retest, where K denotes the number of the types of bin. The mechanism of the three-stage checking can be summarized below. If $g_i \geq g_{L\min}$, we pass the whole lot; otherwise, we will check the number of good dies in each individual wafer of this lot. If $g_{ij} \geq g_{W\min}$, we pass wafer j; otherwise, we will hold this wafer and check its bins. For those hold wafers, if $B_{ijk} \geq n_{k\max}$, we will perform retests for bin k to check whether there are probing errors.

In the above testing procedures, although we may pass the lot or wafer when the threshold-value test is a success, there may be overkills. In general, the percentage of overkills is proportional to the number of probed bad dies, that is, for smaller number of probed bad dies, there will be less overkills. The relationship between them can be found empirically from the real manufacturing process. We let $p_L(B_i)$, $p_W(B_{ij})$ and $p_{bk}(B_{ijk})$ denote the functions of the percentage of the overkills in probed bad dies in lot i, wafer j and bin k, denoted by B_i, B_{ij} and B_{ijk}, respectively. Defining v_i, v_{ij} and v_{ijk} as the number of overkills in lot i, wafer j and bin k, respectively, we have $v_i = p_L(B_i) \times B_i$, $v_{ij} = p_W(B_{ij}) \times B_{ij}$ and $v_{ijk} = p_{bk}(B_{ijk}) \times B_{ijk}, k=1,...,K$. However, we assume that for any retested bin, there will be no overkill because the dies had been probed three times. Thus, a flow chart of the employed testing procedures after the initial two times of wafer probing is shown in Figure 1.

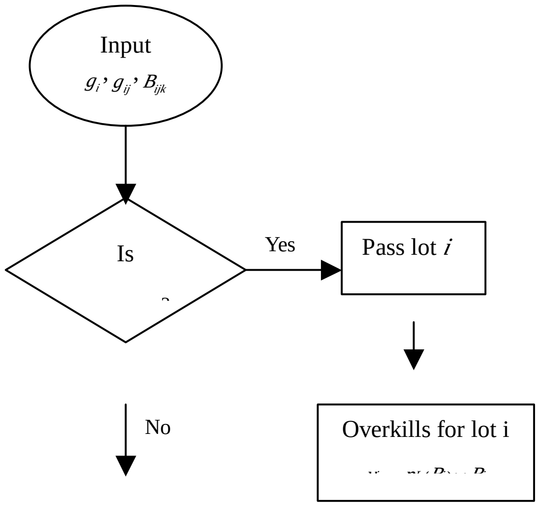
Based on these procedures, we see that if we increase $g_{L\min}$ and $g_{W\min}$ while decreasing $n_{k\max}$, the number of overkills will decrease, however the number of retests will increase. Thus, to reduce both overkills and retests, we will set minimizing the overkills as our objective function while keeping retest rate under a satisfactory level. Furthermore, since the defects of bins in a wafer occurred randomly with a Poisson probability distribution, the testing procedures are of stochastic nature. Based on the above analysis, our problem can be formulated as the following stochastic optimization problem:

$$\min_{x \in X} E[V]$$

subject to {stochastic wafer testing procedures},

$$E[R] \leq r_T, \qquad (1)$$

where $X \equiv [g_{L\min}, g_{W\min}, n_{k\max}, k=1,...,K]$ denotes the vector of threshold values; X denotes the sample space of X; the random variables V and R denote the number of overkills and retests per wafer, respectively; $E[\bullet]$ denotes the expected value of $[\bullet]$; the stochastic wafer testing procedures is described in Figure 1, which will be used to compute the values of V and R for each wafer with randomly generated bins; r_T denotes the tolerable retest rate in units of number of retests per wafer.



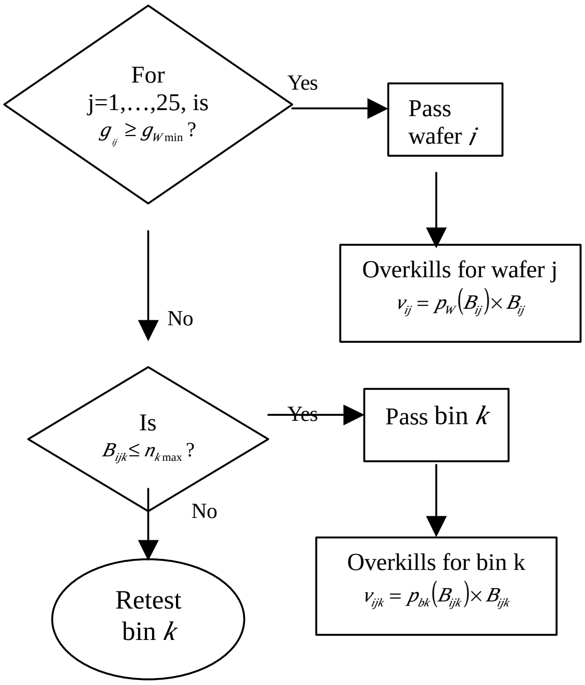


Figure 1: Flow chart of the wafer testing procedures.

It should be noted that the value of r_T is determined based on the economic environment. When the chip demand is weak, the throughput, in general, is not a critical problem in the manufacturing process. We can allow a higher retest rate, that is larger r_T, so as to reduce more overkills to gain more profit; on the other hand, if the chip demand is strong, the throughput is more important, and we should set the value of r_T smaller. Thus, our stochastic optimization problem (1) is to find the optimal vector of threshold values, x^* to minimize $E[V]$, the expected number of overkills per wafer, subject to the employed testing procedures and the constraint on $E[R]$, that is the expected number of retests per wafer should be kept under a tolerable level r_T.

3. Ordinal Optimization Theory Based Two-Level Algorithm

3.1 Preliminaries

The stochastic optimization problem (1) is clearly an NP hard problem in two aspects. The first one is the immense size of the sample space X, which is explained in the following. Considering a lot containing 25 wafers, and each wafer consisting of, say, 2438 dies, the possible ranges of the integer values $g_{L\min}$, $g_{W\min,}$ and $n_{k\max}$ are [0, 60950], [0, 2438] and [0, 2438], respectively. Consequently for $K=12$, the size of X will be more than 10^{30}. The second one is to compute the accurate $E[V]$ and $E[R]$ for a given $X(=[g_{L\min}, g_{W\min}, n_{k\max}, k=1,...,K])$, we need to complete a stochastic simulation. That is to compute the values of V and R for more than 10000 wafers with randomly generated bins based on Figure 1 then take their average values. This implies that we have to perform at least 10^{30} lengthy stochastic simulations to obtain the optimal solution x^* of (1). This is computationally intractable.

Thus, to deal with this NP hard optimization problem (1), we will employ a recently developed optimization technique called Ordinal Optimization (OO) [2] to solve for a good enough solution with high probability instead of searching the best solution x^* for sure.

There are two basic tenets of the OO theory [2]. The first one is that of order versus value in decision making. Of course, determining whether $J(\overline{\ }_1) < J(\overline{\ }_2)$ is much more easier than determining $J(\overline{\ }_1) - J(\overline{\ }_2) = ?$: considering the intuitive example of determining which of the two melons in two hands is heavier versus identifying how heavier one is than the other. The second tenet is the goal softening. Instead of asking the best for sure in optimization, it settles for the good enough with high probability. What softening of the goal buys is on easing of the computational burden. It is much easier to get something in the top n% than it is to get the best. Thus what OO theory concluded is the following: Suppose we simultaneously evaluate a large set of alternatives very approximately and order them according to the approximate evaluation. Then there is high probability that we can find the actual good alternatives if we limit ourselves to the top n% of the observed good choices. Thus, firstly, we use only a very rough model to "order" the goodness of a solution relying on the robustness of ORDER against noise and model error to separate the good solutions from the bad solutions. Secondly, we soften the goal of the problem and look for a good enough solution, which is among the top n% of the search space with high probability. These two steps will greatly reduce the computational burden for the NP hard problem (1).

3.2 A Two-Level OO Approach

3.2.1 Motivation

The current OO theory based approach [2-5] is carried out in the following three steps: (i) Uniformly select N, say 1000, samples from the sample space X of the vectors of threshold values. (ii) Using a rough model of the considered problem to select the top s, say 35, samples from the N, that is the ESTIMATED top 3.5% samples among the N. (iii) Use the exact model of the considered problem to evaluate the s samples obtained in (ii); then the top k, say 1, sample should be the actual good enough, actual top 3.5% among the N, solution with high probability (≥ 0.95) as guaranteed by the OO theory [3].

However, according to [4], the top 3.5% of the uniformly selected N samples will be a top 5% sample of the sample space X with a very high probability (≥ 0.99). Thus, for X with size of 10^{30}, a top 5% sample is a sample among the top 5×10^{28} samples. This certainly not seems to be a good enough solution in the sense of practical optimization. The factor causing this non-satisfactory result is that the N samples are uniformly selected from X. Thus to overcome this defect, we propose a two-level OO approach. In the first level, we will use a rough but efficient and effective model instead of uniform selection to obtain excellent N samples from X to replace step (i) of the regular OO approach as indicated above. Then, in the second level, we will proceed with steps (ii) and (iii). Before the detailed description of our two-level approach, we need to convert (1) into an unconstrained problem first.

3.2.2 Converting (1) to the Unconstrained Problem

In general, the constrained OO problem is typically harder than the unconstrained one [5]. However, since our constraint on the retest rate shown in (1) is a soft-constraint in a sense. Therefore, we can use a penalty function to relax that constraint and transform (1) into the following unconstrained stochastic optimization problem:

$$\min_{x\in X} \; E[V] + P(E[R]-r_T)(E[R]-r_T)$$

subject to {the stochastic testing procedures} (2)

where $P(E[R]-r_T)$ denotes a continuous penalty function of $E[R]-r_T$ such that $P(E[R]-r_T)$ >0 if $E[R]>r_T$, and $P(E[R]-r_T)$ =0, otherwise.

3.2.3 The First-Level Approach

As indicated in the OO theory [2], "order" of the samples is likely preserved even with a rough model. Thus, to select N excellent samples from X without taking much computation time, we need to construct a rough but efficient and effective model to evaluate the objective value of (2) for a given sample x, i.e. a vector of threshold values, and use an efficient scheme to select excellent samples. Our model is constructed based on two Artificial Neural Networks (ANNs), and our selection scheme is the Genetic Algorithm (GA).

3.2.3.1 The Artificial Neural Network (ANN) Based Model

The ANN can be trained to implement a given mapping between the inputs and outputs. Considering the inputs as the samples $x\in X$, then we can use two ANNs to implement the mapping from the inputs to the outputs of $E[V]$ and $E[R]$, respectively. Once these two ANNs are trained by a given set of training data, we can input any sample x to the two ANNs to obtain the corresponding $E[V]$ and $E[R]$, which will be used to calculate the objective value of (2). This forms our effective and efficient model to calculate the objective value of (2) for a given sample x.

The ANN we employed in our approach is the two-layer feed-forward back propagation neural network. We obtain the set of training data by the following two steps. (a) Narrow down the sample space X by excluding the irrational threshold values and denote the reduced sample space by $\hat{X}$. (b) Uniformly select n samples from $\hat{X}$ and compute the corresponding $E[V]$ and $E[R]$ using a shorter stochastic simulation, that is to perform the simulations of the testing procedures shown in Figure 1 for 300 wafers with randomly generated bins and take the average of the values of V and R.

Denoting the n samples by $x_i, i=1,...,n$, the n corresponding $E[V]$ by $v_i, i=1,...,n$, and the n corresponding $E[R]$ by $r_i, i=1,...,n$. Then, the training problems for these two ANNs to determine their branch weights are:

$$\min_w \sum_{i=1}^{n}[v_i - f_1(x_i \mid w_1)]^2 \qquad (3)$$

and

$$\min_w \sum_{i=1}^{n}[r_i - f_2(x_i \mid w_2)]^2, \qquad (4)$$

where w_1 and w_2 denote the vectors of the branch weights of the two ANNs; $f_1(x_i \mid w_1)$ and $f_2(x_i \mid w_2)$ denote the actual outputs of the two ANNs for the $E[V]$ and $E[R]$ when the input is x_i and the vectors of branch weights are w_1 and w_2, respectively. To speed up the convergence of the training, (3) and (4) are best solved by the Levenberg-Marquardt algorithm [6,7] and Scaled Conjugate Gradient algorithm [8,9], respectively.

3.2.3.2 The Genetic Algorithm (GA)

With the above effective and efficient objective value (or the so-called fitness value in GA terminology) evaluation model, we can then efficiently select the excellent N samples from X using GA, which is briefly described as follows. Assuming an initial random population produced and evaluated, genetic evolution takes place by means of three basic genetic operators: (a) parent selection; (b) crossover; (c) mutation. The population in GA terminology represents a sample x, i.e. a vector of threshold values, in our problem, and each population is encoded by a string of 0s and 1s. The string is called a chromosome. Parent selection is a simple procedure whereby two chromosomes are selected from the parent population based on their fitness values. Solutions with high fitness values have a high probability of contributing new offspring to the next generation. The selection rule we used in our approach is a simple roulette-wheel selection [10]. Crossover is an extremely important operator for the GA. It is responsible for the structure recombination (information exchange between mating chromosomes) and the convergence speed of the GA and is usually applied with high probability (0.7). The chromosomes of the two parents selected are combined to form new chromosomes that inherit segments of information stored in parent chromosomes. There are many crossover scheme, we employ the single-point crossover [10] in our approach. While crossover is the main genetic operator exploring the information included in the current generation, it does not produce new information. Mutation is the operator responsible for the injection of new information. With a small probability, random bits of the offspring chromosomes flip from 0 to 1 and vice versa and give new characteristics that do not exist in the parent population. In our approach, the mutation operator is applied with a relatively small probability (0.02) to every bit of the chromosome.

There are two criteria for the convergence of GA. One is when the fitness value of the best population does not improve from the previous generation, and the other is when evolving enough generations.

We start from 10000 randomly selected samples from X as our initial populations. After the applied GA converges, we rank the final generation of populations based on their fitness values and pick the top 1000 populations to serve as the N samples in the second-level OO approach.

3.2.4 The Second–Level Approach

In the second-level, starting from the N samples obtained in the first level, we will proceed with step (ii) by evaluating each sample using a rough model, which is a shorter stochastic simulation based on Figure 1 as described previously, for evaluating the objective value of (2). We will then order the N samples based on the obtained objective values and choose the top s (=35) samples. Then in step (iii), we will evaluate each of the s samples using an exact model. The exact model we employed here for each sample is to calculate the objective value of (2) based on a longer stochastic simulation. That is replacing the 300 wafers with randomly generated bins in shorter stochastic simulations by 10000 wafers. Then the sample associated with the least objective value of (2) is the solution that we are looking for.

3.3 The OO Theory Based Two-Level Algorithm

Now, our OO theory based two-level algorithm can be stated as follows.

Step 0: Narrow down the sample space X by excluding the irrational values of $g_{L\min}$, $g_{W\min}$ and $n_{k\max}, k=1,...,K$, and denote the reduced sample space by $\hat{X}$.

Step 1: Uniformly select 300 samples from $\hat{X}$ as inputs and perform a shorter stochastic simulation based on Figure 1 to obtain the corresponding approximate $E[V]$ and $E[R]$. Training two ANNs to implement the mapping between the inputs and the corresponding two sets of outputs.

Step 2: Randomly produce 10000 samples from $\hat{X}$ as the initial populations. Apply GA to these populations using the efficient and effective fitness-value evaluation model based on the two ANNs trained in Step 1. After the algorithm converges, we rank all the final populations based on their fitness values and select the top N (=1000) populations.

Step 3: Run a shorter stochastic simulation for each of the N samples obtained in Step 2 to evaluate the corresponding objective value of (2). Ranking the N samples based on their objective values and select the top s (=35) samples.

Step 4: Run a longer stochastic simulation for each of the s samples to evaluate the corresponding objective value of (2). The sample, i.e. the vector of threshold values, with the least objective value of (2) is the good enough solution that we are looking for.

4. Simulation Results

Our simulation is based on the following data obtained from certain product of a foundry. Each lot contains 25 wafers, and the total number of dies in a wafer and a lot are 2438 and 60950, respectively. There are 12 bins, and their means $\text{«}_k, k=1,...,12,$ are 11.6,13.4,27.3,0.3,20.5,1.2,1.4,59.5,34.0,6.6,2.5,and 0.2. The yield rate is around 92.67%. The functions of the percentage of the overkills in probed bad dies, $p_L(B_i)$, $p_W(B_{ij})$ and $p_{bk}(B_{ijk})$ we employed here are

$p_{bk}(B_{ijk}) = 0.01\times(1.0+3.0\times\frac{B_{ijk}-\text{«}_k}{\text{«}_k})$,

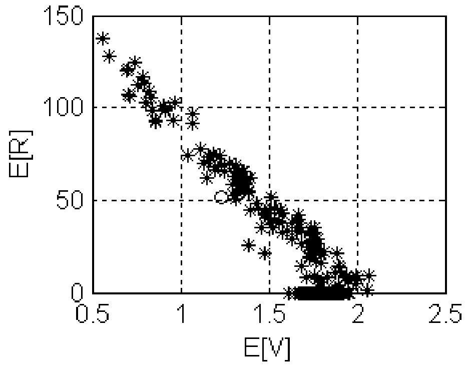
$p_W(B_{ij}) = 0.01\times(1.0+3.0\times\frac{B_{ij}-\text{«}_W}{\text{«}_W})$, and

$p_L(B_i) = 0.01\times(1.0+3.0\times\frac{B_i-\text{«}_L}{\text{«}_L})$, where $\text{«}_W=\sum_{k=1}^{K}\text{«}_k$

and $\text{«}_L = 25\text{«}_W$. We used the sigmoid-type function as our penalty function $P(E[R]-r_T)$ in (2). We set the tolerable retest rate r_T =50.

In Step 0 of our algorithm, the narrowed ranges we use for the $g_{L\min}$ and $g_{W\min}$ are [30000, 60950] and [1200, 2438], respectively, while the range for the $n_{k\max}$ is [1, 3 «_k], k=1,…,12. We uniformly select 300 samples from this reduced sample space $\hat{X}$. In Step 1, the shorter stochastic simulation for each vector of threshold values is performed by processing 300 wafers through the testing procedures with randomly generated bins based on a Poisson probability distribution with parameters of $\text{«}_k, k=1,...,12$. In Step 2, the convergence criteria we employed for our GA is when the evolving number of generations exceed 50. In Step 3, the shorter stochastic simulation for each vector of threshold values is carried out in the same way as in Step 1, so does the longer stochastic simulation in Step 4 except for replacing 300 by 10000 wafers.

The good enough vector of threshold values we obtained is $g_{L\min}=56525,$ $g_{W\min}=2261,$ and $(n_{1\max},...,n_{12\max})=(31,21,10,2,8,3,4,63,12,20,1,3)$. The $E[V]$ and $E[R]$ resulted from these good enough threshold values are shown in Figure 2 by the ○ point. In the same figure where the $E[V]$ and $E[R]$ shown by the * points are resulted from 1000 randomly selected vectors of threshold values. We see that for the retest rate under 50, the expected number of overkills per wafer resulted by our vector of threshold values is the best among all the randomly selected vectors of threshold values.



Legend:
o – the (E[V],E[R]) resulted from the good enough vector of threshold values determined by our algorithm.
* – the (E[V],E[R]) resulted from the 1000 randomly generated vectors of threshold value.

Figure 2: Performance comparison of the randomly generated threshold values and the threshold values determined by our algorithm.

5. Conclusions

In this paper, we have proposed a novel formulation to reduce the overkills and retests by determining a good enough vector of threshold values in a wafer testing process of three-stage check. Our formulation provides a flexibility for practical applications by taking various economic conditions into account. In addition, the presented OO theory based two-level algorithm will not only work successfully in the stochastic optimization problem considered in this paper but also be useful for other semiconductor manufacturing related optimization problems.

Acknowledgment

This research work is supported in part by the National Science Council in Taiwan, R.O.C., under grant NSC91-2213-E009-070, and in part by the Industrial Technology Research Institute in Taiwan, R.O.C., under the project #C324K52AN0.

References

[1] Muriel S., Garcia P., Marie-Richard O., Monleon M., and Recio M., “Statistical bin analysis on wafer probe’, ASMC 2001, pp.187-192.

[2] Ho, Y.C., “Soft optimization for hard problems,” Lecture notes, Harvard Univ., MA, 1996.

[3] Lau, T.W.E. and Ho, Y.C., “Universal alignment probability and subset selection for ordinal optimization,” JOTA, vol. 39, no. 3,June 1997.

[4] Lin, S.-Y. and Ho, Y.C., “Universal alignment probability revisited,” JOTA, May 2002.

[5] Ho, Y.C., Cassandras, C.C., Chen, C.H., and Dai, L., “Ordinal optimization and simulation,” Journal of Operation Research Society, v. 21, pp. 490-500, 2000.

[6] Marquardt, D., "An algorithm for least squares estimation of nonlinear parameters. SIAM J. Appl. Math. 11, 431-441.

[7] Hagan, M. T., and M. Menhaj, “Training feed-forward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[8] Martin Fodslette Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, 6:525--533, 1993.

[9] Hagan, M. T., H. B. Demuth, and M. H. Beale. *Neural Network Design*, Boston, MA: PWS Publishing, 1996.

[10] G. Lindfield and J. Penny. Numerical Methods using Matlab. 2nd edition, Prentice Hall, 2000.