# 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 $n_{k\max}$, $k = 1,\ldots,K$, 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 $g_i \ge g_{L\min}$, we pass the whole lot; otherwise, we will check the number of good dies in each individual wafer of this lot. If $g_{ij} \ge g_{W\min}$, we pass wafer $j$; otherwise, we will hold this wafer and check its bins. For those hold wafers, if $B_{ijk} \ge n_{k\max}$, 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 $v_{ijk} = p_{bk}(B_{ijk}) \times B_{ijk}$, $k = 1,\ldots,K$. 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:

$$\begin{aligned} &\min_{x\in X} E[V] \\ \text{subject to}\quad &\{\text{stochastic wafer testing procedures}\}, \\ &E[R] \le r_T, \qquad (1) \end{aligned}$$

where $X \equiv [g_{L\min}, g_{W\min}, n_{k\max}, k = 1,\ldots,K]$ 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.

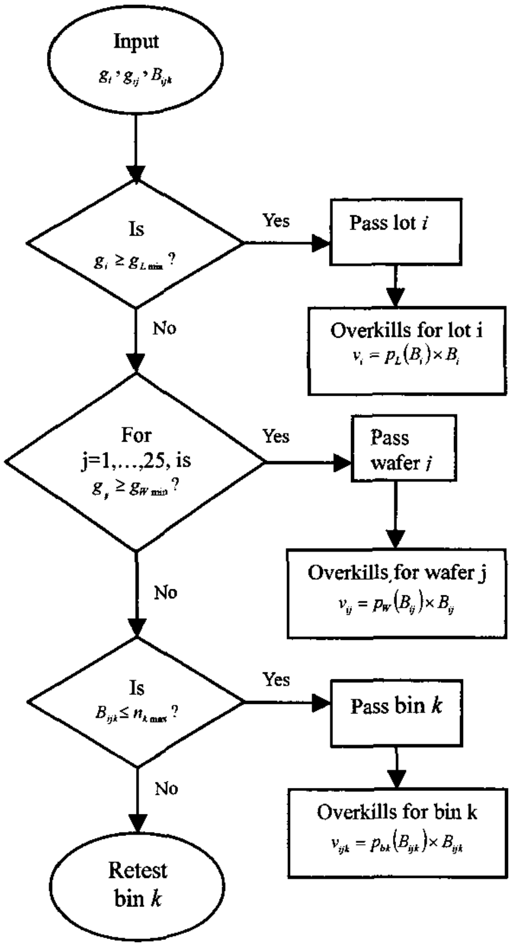


**Figure 1:** Flow chart of the wafer testing procedures.

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 $J(\theta_1) < J(\theta_2)$ is much more easier than determining $J(\theta_1) - J(\theta_2) = ?$: 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 ($\geq 0.95$) 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 ($\geq 0.99$). Thus, for $X$ with size of $10^{30}$, a top 5% sample is a sample among the top $5 \times 10^{28}$ 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:

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

$$\text{subject to } \{\text{the stochastic testing procedures}\} \qquad (2)$$

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:

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

and

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

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 meanś 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 $\mu_k, k=1,\ldots,12$, 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

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

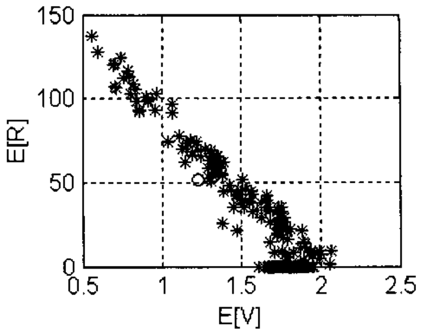
$$p_W(B_{ij}) = 0.01 \times (1.0 + 3.0 \times \frac{B_{ij} - \mu_W}{\mu_W}), \text{ and}$$

$$p_L(B_i) = 0.01 \times (1.0 + 3.0 \times \frac{B_i - \mu_L}{\mu_L}), \text{ where } \mu_W = \sum_{k=1}^{K} \mu_k$$

and $\mu_L = 25\mu_W$. 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$\mu_k$], $k$=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 $\mu_k, k=1,\ldots,12$. 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 $(n_{1\max},\ldots,n_{12\max}) = (31,21,10,2,8,3,4,63,12,20,1,3)$. 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 Industrial Technology Research Institute in Taiwan, R.O.C., under the project #C324K52AN0, and in part by the National Science Council in Taiwan, R.O.C., under grant NSC91-2213-E009-070.

### 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.