國立交通大學

統計學研究所
博士論文

利用統計方法自動依工程忍受度判斷機台差異及其在半導體製程改善之應用

A New Statistical Method for Automatic Partitioning Tools According to Engineers' Tolerance Control in Process Improvement

研 究 生：涂凱文

指導教授：盧鴻興　教授

中華民國九十七年八月

利用統計方法自動依工程忍受度判斷機台差異及其在半導體製程改善之應用

A New Statistical Method for Automatic Partitioning Tools According to Engineers' Tolerance Control in Process Improvement

研 究 生：涂凱文　　Student：Kai-Wen Tu

指導教授：盧鴻興　　Advisor：Dr. Henry Horng-Shing Lu

國 立 交 通 大 學
統 計 學 研 究 所
博 士 論 文

A Thesis

Submitted to Institute of Statistics College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Ph. D

in

Institute of Statistics
August 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

A New Statistical Method for Automatic Partitioning Tools According to Engineers' Tolerance Control in Process Improvement

Student：Kai-Wen Tu Advisor：Dr. Henry Horng-Shing Lu

Institute of Statistics

National Chiao Tung University

ABSTRACT

In the semiconductor industry, tool comparison is a key task in the yield and the product quality enhancements. We developed a new method, called tolerance control partitioning (TCP), to automatically partition tools into several homogenous groups based on the related metrology results. This methodology is based on a hierarchical normal model and the implementation is carried out using a Bayesian approach. There are several advantages of using the TCP method. First, it takes into account the unbalanced usage of the tools in the manufacturing processes. Moreover, the "engineer's tolerance control" can be incorporated into the TCP method via the specification of the priors in the Bayesian analysis, which justifies the significant difference between groups according to the experts' knowledge. This specification not only has the advantage of adjusting the number of partition groups but also avoids the problem of having too many partition groups with small differences which is often encountered in the conventional approaches. Some simulation results illustrate the advantages of the TCP method compared to the method of classification and regression trees (CART). Moreover, the TCP method is applied to two real examples for the yield and Cp/Cpk enhancement in the semiconductor industry. Both results confirm the practical usefulness of the proposed method. For general applications, the TCP method is also useful for other similar problems such as the comparisons between several experimental recipes or the comparisons between different materials.

Acknowledgements

謹寫下這段致謝文感謝所有人，也作爲自己人生的反省與珍貴的回憶。

決定重回已經離開十多年的學生生涯，其實不只是想再充實自己統計的專長，也是完成自己年少的夢想。只是時光飛逝一去不回，當時年少的我，如今已是身兼公司員工，兩個小孩的爸爸等多重身份的中年男子。

早上上班，晚上唸書的日子，讓日子每天都過得很快，常常累到想像自己是一位精疲力竭的劍客，用劍撐著疲痛的身軀，但卻又得目光銳利地看著下一個目標，準備前去。身體是疲憊的但卻鬥志高昂。如今博士生涯將近尾聲，其中的感觸非筆墨所能形容。我想千言萬語還是感謝兩字。

要感謝的人太多了！

感謝清華統計所的趙蓮菊老師，專業又高水準的統計推論課程，讓我快速進入〝博士生〞的狀況。更感謝她體諒我因家人患病那段日子分心無法兼顧學業，要不是她的一念之仁，我想我的博士生涯可能會有完全不同的結果。

感謝我碩士班指導教授陳鄰安老師，從我準備入學到畢業，總是給我很多鼓勵和幫助，並時常分享一些做學問的心得和指引我一些未來的方向，讓我受用不盡。我相信老師會是我一輩子的好朋友。

感謝陳志榮老師提供的高等機率論課程，他的教導方式，讓我對一直最害怕的科目，找到全新的學習方式，這對我通過資格考幫助很大。

感謝我的良師益友清華統計所的徐南蓉老師，雖然是大學同學，但卻是我博士生涯的學習導師，總是在我學習遇到困境時成為最佳的諮詢對象，也省下很多徒勞蹉跎的時光。在課堂上聽她精采的時間序列課程也是博士生涯中一大享受。

感謝也懷念我已故的恩師李昭勝老師，我從他身上學到了如何對後生晚輩提攜，鼓勵和支持。在我的博士路上，每當遇到困難，他總是提供了最有效的幫忙，他的鼓勵常常讓心情低落的我注入強心劑。心情安頓好了，就有再衝刺的力量。我想他給了我很多的信心，有時我甚至感覺老師比我對自己還有信心。而這是對一位離開十多年學生生涯的在職生最需要的。我更感謝他對我在博士學習規劃上的尊重。讓我有更多的發揮空間。而他對學術的熱忱，也是身為後生晚輩的我最佳的典範。我想我這〝問題〞學生可能讓他多操勞了！他的離開讓我有無限的感傷與懷念。「哲人日已遠，典型再夙昔」，他對我的師恩，將永生難忘。他對我的提點，會是我不斷努力下去的動力。

感謝我的指導教授盧鴻興老師，感謝他願意伸出援手，協助我，指導我完成博士求學的最後一段路，老師對學術的熱忱也是值得我再三學習的。

感謝所上的秘書郭姐，對我這個常常不在所上的特殊學生，提供特別的服務，讓我這〝忙碌的〞在職生方便許多，我在博士路上最後的小插曲也必須感謝她盡力的幫忙，很抱歉造成她的困擾。

也感謝所上其他授課老師的指導。

感謝交大統計所，開了一扇門，讓一位〝工業界〞的在職生有機會再接觸、再學習統計，我相信這將會鼓勵更多充滿熱情的統計工作者，再成長的機會。也會成為在業界打拼的統計人最佳的後盾。

感謝我公司的長官潘文森副總、古延輝協理與彭誠湧資深處長的支持，他們用行動表現出工業界對統計的重視，並提供了統計工作者再進步，再成長的機會。同時也感謝我公司的工作夥伴們，因為他們的優異表現，讓我在博士求學過程無後顧之憂。我也必須感謝在公司的一些好朋友們時常關心我的近況，雖然只是兩句關心的話語卻讓我再度充滿信心。

感謝我的口試委員洪志真老師、曾勝滄老師、黃榮臣老師及黃信誠老師，

對我的建議和指導，讓我對論文撰寫與學術嚴謹的態度永生難忘。也讓我對教育有新一層的體會。想起已故李昭勝老師常提醒我的一句話 “ Good Luck is always with well preparation” 我想一切還是因為自己做的不夠好，我會繼續努力。

感謝國際期刊 *IEEE Transactions on Semiconductor Manufacturing* 的 editor 及 reviewer 協助審核並 Accept 我的博士論文。可以發表文章在國際期刊一直是我在博士學習過程設定最大的目標。很高興我做到了，它真的給了我很大的信心與鼓勵。從自己在業界找題目、學統計技術、腸枯思竭想方法、挑燈夜戰做 simulation、做出成果、學習寫論文、學習打字到發表。心理上經歷了被國際期刊拒絕的失望感及苦苦等待國際期刊回覆的苦悶感到最後被國際期刊接受那不可思議的驚喜。很高興我在博士生的階段走完全程。這些點點滴滴將會是我人生中一段重要的回憶也是我下階段再出發最有幫助的經驗。

最後，我必須送上我最深的感謝給我的家人，畢竟身為人父、人夫、人子，做任何決定都將影響一家人的生活。感謝我的父、母、岳父、岳母對我的支持。我拿到博士學位，他們甚至比我還高興。感謝我兩位可愛的兒子光遠、光庭，他們不但乖巧懂事讓我不用分心，還會偶而做卡片鼓勵爸爸〝要保重，要加油〞。爸爸的博士之旅相信也讓他們感受到爸爸對讀書的熱情，希望這對他們的求學過程會有幫助。感謝我的賢內助李柏青小姐，雖然她工作家庭兩頭忙十分辛苦，但卻願意成就一位愛唸書老公的夢想。因為我的博士路讓她少了很多戶外郊遊、逛街及休息的機會也少了很多老公的寒喧問暖。我會十分珍惜人生路上老婆給我這圓夢的機會，我應該也為她戴上博士的方帽，以感謝她為我與家庭的犧牲與付出。〝老婆謝謝您〞。

Contents

1. Introduction

The importance of semiconductor technology in today's world can hardly be exaggerated. Semiconductor devices are absolutely essential components for almost all electronic products. Without semiconductors, most of the electronic products and the systems cannot be made or operated and their influences on human society are beyond belief. The global semiconductor industry had around US$230 billion worth of revenue in 2005 and keeps creating new opportunities, socio-economic advancements and new human developments to nations and societies around the world [1]. Since building a modern wafer fabrication facility needs around US$3 billion, enhancing the yield rapidly to volume the production becomes an extremely important source of the competitive advantage in the hyper-competitive world on semiconductor manufacturing. The sooner a potentially lucrative circuit yields, the better the manufacturer generates a revenue stream. On the other hand, rapidly identifying the cause of yield loss can restore a revenue stream and prevent the destruction of the materials in process [2] [3].

In the following, the semiconductor manufacturing is briefly introduced. It follows a very complex process flow which is quite different from the traditional manufacturing industries. It takes about 30-60 days to complete the processes of making bare silicon wafers into integrated circuits, such as the microprocessors or the memory chips. In general, 25 wafers are processed together in a group called a lot, and the size of each wafer ranges from 3 to 12 inches in diameter. Each wafer could contain thousands of dies depending on the size of the die being produced. During the manufacturing process, the lots are manufactured through lots of process steps (more than 150 process steps). Each process step involves several tools for production. After completing each process step, the metrology systems collect the physical data and the electrical data, such as the film thickness, film uniformity, critical dimension, overlay, defect particle count, voltage, current and wafer sort, etc. At the end of the all process steps, the Wafer Acceptance Test (WAT) with 100–500 electrical test items and the Wafer Sort Test (WST) with 50-100 test items are performed sequentially to each wafer. The objectives of WAT and WST are to perform the device characteristic analysis and the die functionality sorting, respectively. Since these testing data also characterize the quality of the manufacturing and the performance of the products, therefore how to use these data to improve the process itself becomes an interesting issue. However these data are huge with lots of variables (about 100M–1G for each lot), it is very time-consuming for engineers to analyze the data and find out the sources of variations in the production processes. Among various analyses, tool comparison is one key task for engineers in the yield improvements and therefore an effective and time-efficient method for comparing tools is critical for rapidly improving the yields [4]. In the following, we briefly review the conventional approaches in this area, including the multiple comparison methods and the clustering methods.

For multiple comparisons, the analysis of variance (ANOVA) for normal data and the Kruskal-Wallis test [5] for non-normal data are the two most popular statistical methods for testing the significant differences between population means among groups. For our tool partition problem, in order to compare the performances among different tools (i.e., groups) at each process step, the engineers perform these two statistical tests regarding the distribution of the considered metrology data associated with each tool. By quickly reviewing the testing result for each individual step, tool differences might be detected at certain process steps and an alarm will be triggered for further checking or investigation. In general, the main purpose of this kind of testing procedures is to find the variation sources (i.e., which process steps) and identify the possibility of abnormal tools. After finding the significant differences among tools at certain steps, the engineers will partition all the relevant tools into several homogenous groups and further identify the best groups or the problematic groups of tools in order to enhance the product quality or to exclude the worst tools, see for examples in [6]-[8]. This partition problem is an important and practical issue for engineers but cannot be handled by the ANOVA or the Kruskal-Wallis test simultaneously. Some multiple pairwise comparison procedures, such as the methods suggested by Fisher [9], Tukey [10], Keuls [11], Duncan [12], Scheffe [13], and Dunnett [14] [15], provide useful information about the ranking or ordering structures of the group means but these methods cannot directly partition different tools (or treatments) into homogenous and non-overlapping groups. For example, there are three tools to be partitioned and their sample means satisfy $\overline{Y}_1 \le \overline{Y}_2 \le \overline{Y}_3$. Suppose that a multiple comparison procedure finds that the differences $|\overline{Y}_1 - \overline{Y}_2|$ and $|\overline{Y}_2 - \overline{Y}_3|$ are not significant but the difference $|\overline{Y}_1 - \overline{Y}_3|$ is significant. It is not clear how to partition these three tools into homogenous but non-overlapping groups since both $(\overline{Y}_1, \overline{Y}_2)$ and $(\overline{Y}_2, \overline{Y}_3)$ are reasonable homogenous groups.

Another popular approach for partitioning is using the cluster analysis. Scott and Knott [16] suggested a procedure which starts by dividing the *k* means into two groups and then performs a test to decide whether the partition is acceptable. This approach is equivalent to a hypothesis testing problem:

$$
\begin{aligned}
H_0 : \theta_1 = \theta_2 = \dots = \theta_\kappa \quad \text{v.s.} \quad H_1 : \theta_i\text{'s equal for } i \in P_1, \\
\theta_j\text{'s equal for } j \in P_2.
\end{aligned}
$$

where $\{P_1, P_2\}$ forms a partition of $\{1, 2, \dots, \kappa\}$ and P_1 and P_2 are two disjoint and nonempty sets with $P_1 \cup P_2, = \{1, 2, \dots, \kappa\}$. If the test is significant at some chosen level α, similar testing procedure is then applied to each individual P_i, i=1,2. The procedure is continued sequentially until all tests are not rejected (i.e., no further partition is necessary). Worsley [17] proposed a nonparametric version of Scott and Knott's method. Although this approach is intuitive and easy to implement, the Type I error of the entire test is difficult to control due to sequential testing procedures, in particular when the number of splitting gets larger. Moreover, the final partition result may not be unique which highly depends on the initial partition.

To overcome the difficulty of controlling the Type I error (the probability of erroneous grouping) for the sequential testing procedure, Calinski and Corsten [18] proposed two cluster methods to partition these tools (or treatments) in a balanced design by embedding the simultaneous testing procedures based on *F* test and Studentized range test, respectively. Although this approach solved the problem of controlling Type I error, it still has several other disadvantages, such as the partition groups are too many with small differences when the number of observations for each tool is large; the issue about unbalance data is not considered which generally loses the power when the usages are quite different among tools.

Jolliffe [19] proposed an alternative method to perform the cluster analysis. This approach used a particular dissimilarity measure which is defined by the *P*-value of the Studentized range test for testing the difference between two group means. A larger *P*-value indicates that two groups are more similar and a smaller *P*-value indicates that two groups are more distinguishable. One critical issue for this hierarchical clustering approach is to determine the number of clusters which is usually determined subjectively.

Data mining approaches [20] [21], such as the classification and regression trees (CART) [22] and the neural networks [23], have also been used for the partition problem. Recently, some commercial data analysis software (for example: Yield Dynamics, BI IBM, Odyssey YMS, and dataPOWER) in engineering use these approaches for the yield enhancements. However, these approaches involve supervised algorithms which rely on more complex initial parameter setups and the partition results are usually sensitive to these setups [22], and therefore it is somehow difficult for engineers to use them in practice. In Appendix A, the CART algorithm is briefly introduced which will be compared with the proposed method in the simulation study.

To sum up, each above-mentioned method has its own advantages and disadvantages compared to the other methods under different circumstances. But none of them is capable of incorporating the experts' specific opinions into the statistical analysis. For example, for our tool partition problem, the engineers' tolerance controls which quantify the tolerance (minimum value) of tool differences should be used in some way in the analysis on determining the grouping structure. Another challenge for our tool partition problem in the manufacturing processes is to account for unequal tool usages. Unequal usages for different tools at each process step make the numbers of processed lots for each tool varying which induces unbalance data issue for the statistical tests. Montgomery [24] addressed that the statistical tests for multiple comparison lose power for unbalance data. The goal of this thesis is to develop a method for tool partition which incorporates the experts' specific opinions on the tolerance of tool differences and considers the unbalance data issue simultaneously.

We formulate the tool partition problem as a hierarchical model [25] in which the metrology measurements from each tool follow a normal distribution with tool-specific mean. It is reasonable to expect that the tool-specific means for "similar tools" are related in some way, such as viewing these tool-specific means as realizations from the same distribution. Under this setup, the engineers' tolerance of tool differences can be naturally incorporated into the variance structure of the model imposed on the tool-specific means. For such hierarchical models, Bayesian analysis is the most popular method for inference in the literature. We will use the Bayesian analysis for searching the possible grouping structure for different tools in which the reversible jump Markov chain Monte Carlo (RJMCMC) algorithm, proposed by Green [26], will be used for the implementation. We called this proposed procedure “tolerance control partitioning” (TCP) which partitions tools into several homogenous groups subject to the engineer’s specific tolerance about the mean differences.

The remainder of this paper is structured as follows. In Chapter 2, the hierarchical model is introduced and the guidelines for choosing the priors for the parameters and hyper-parameters in the Bayesian analysis are also addressed. In Chapter 3, three simulation experiments are designed to illustrate the advantages of using the TCP method. For comparison, the partition results using the pruning scheme for the CART method (which also incorporates the tolerance of tool differences into account) are also considered in the simulations. In Chapter 4, two real examples in the semiconductor industry are analyzed for illustration. One example is related to the yield enhancement and the other example is about the Cp/Cpk enhancement. In addition, we propose two new ideas to integrate TCP with a statistical dashboard [4] for yield enhancement and automatic process control (APC) [27]-[29] for Cp/Cpk enhancement, respectively. In Chapter 5, some conclusions and discussions for the TCP method are given. In Chapter 6, possible extensions of applying the TCP method are addressed for future work.

2. Methodologies

In this chapter, we introduce the TCP methodology. In Section 2.1, we formulate our tool partition problem by a hierarchical Bayesian model. In Section 2.2, we briefly introduce the reversible jump Markov chain Monte Carlo (RJMCMC) method. In Section 2.3, we apply the RJMCMC to determine the best partition and estimate the model parameters under a Bayesian approach for the tool partition problem. In Section 2.4, we suggest some guidelines for the initial setups for the model parameters and the hyper-parameters in the RJMCMC algorithm for the TCP method. In Section 2.5, we describe the standard method for monitoring the convergence of the Markov chain in the Bayesian context.

2.1 A Hierarchical Bayesian Model for Partition Problems

For each observation, Y denotes the response variable (e.g., the yield), and x denotes the categorical predictor with J possible categorical levels (e.g., tools). The conditional distribution of Y given x is, $Y \mid x = j$ ~ Normal (θ_j, σ^2) for the j-th tool, and $\theta = (\theta_1, \theta_2, ..., \theta_J)'$ is the unknown parameter vector. The TCP method intends to partition different tools into several homogenous groups (that is, to partition

$\{\theta_j : j = 1, 2, ..., J\}$ into several homogenous groups) according to the values of the response variable.

Each partition for J tools can be represented as a collection g=$\{S_1, S_2, \ldots, S_\kappa\}$ with κ groups and each group S_k is a subset of {1, 2,…, J}, satisfying $\bigcup_{k=1}^{\kappa} S_k = \{1, 2, \ldots, J\}$ and $S_i \cap S_j = \phi$ for $i \neq j$. The number of groups for the partition g, κ, determines the degree of heterogeneity among different tools. According to the partition structure (or the grouping structure), we further assume that the mean parameters θ_j 's follow the same normal distribution with the hyper-parameters μ_k and τ when these θ_j 's belong to the same group S_k. That is, $\theta_j \sim$ Normal (μ_k , τ^2) if $j \in S_k$, j=1,2,…, J .

If the partition g is known in advance, all the parameters can be easily estimated by using the maximum likelihood. But, the partition g is unknown and a Bayesian method is used for estimation. In the following, the prior distributions for the mean and variance parameters and the related hyper-parameters are specified:

(1) the prior for μ_k: $\pi(\mu_k)\sim$ Uniform (a, b), for k=1,2,…, κ, and $\{\mu_k\}$ are mutually independent,

(2) the prior for τ^2: $\pi(\tau^2)$ is a scaled inverse chi-squared with degrees of freedom ν and the scale parameter $s^2 \equiv (tolerance / 6)^2$ in which the *tolerance* is a tuning parameter defined by the engineers to stand for the

acceptable difference between tools,

(3) the prior for σ^2: $\pi(\sigma^2)$ is distributed as a scaled inverse chi-squared with degrees of freedom ν_1 and the scale parameter s_1^2.

For the partition *g*, following the specification in Consonni and Veronese [30], the prior distribution is defined as

$\pi(g) \propto \dfrac{\kappa^{-1}}{\{\# of \text{ partitions whose degree} = \kappa\}}$, where κ is the degree of partition *g*.

Under the above prior setup, the posterior distribution for the parameters satisfies

$$p(g,\boldsymbol{\theta},\boldsymbol{\mu},\sigma^2,\tau^2 \mid \boldsymbol{y}) \propto p(g,\boldsymbol{\theta},\boldsymbol{\mu},\sigma^2,\tau^2,\boldsymbol{y})$$

$$= p(\boldsymbol{y} \mid \boldsymbol{\theta},\sigma^2)\,p(\boldsymbol{\theta} \mid g,\boldsymbol{\mu},\tau^2)\,\pi(g)\,\pi(\boldsymbol{\mu})\,\pi(\sigma^2)\,\pi(\tau^2)$$

$$\begin{aligned}&\propto \left(\frac{1}{(2\pi\sigma^2)^{1/2}}\right)^{\sum_{j=1}^{J} n_j} e^{\frac{-\sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij}-\theta_j)^2}{2\sigma^2}} \left(\frac{1}{(2\pi\tau^2)^{1/2}}\right)^{J} e^{\frac{-\sum_{j=1}^{J}\left(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)}\right)^2}{2\tau^2}} \\ &\quad\times \frac{\kappa^{-1}}{\{\#\text{ of partitions whose degree} = \kappa\}} \left(\frac{1}{b-a}\right)^{\kappa} \left\{\prod_{k=1}^{\kappa} \mathrm{I}_{(\mu_k\in(a,b))}\right\} \\ &\quad\times \frac{(\nu_1/2)^{(\nu_1/2)}}{\Gamma(\nu_1/2)} s_1^{\nu_1} (\sigma^2)^{-(\nu_1/2+1)} e^{-\nu_1 s_1^2/(2\sigma^2)} \frac{(\nu/2)^{(\nu/2)}}{\Gamma(\nu/2)} s^{\nu} (\tau^2)^{-(\nu/2+1)} e^{-\nu s^2/(2\tau^2)}\end{aligned} \qquad (1)$$

where $\boldsymbol{y} = \{y_{ij} : j=1,2,\ldots,J; i=1,2,\ldots,n_j\}$, $\boldsymbol{\mu} = (\mu_1,\mu_2,\ldots,\mu_\kappa)'$, $g = \bigcup_{k=1}^{\kappa} S_k$, and $\boldsymbol{\theta} = (\theta_1,\theta_2,\ldots,\theta_J)'$. This posterior for the model parameters does not has a close form but is proportional to the joint distribution of the observed variables and the unknown parameters. For this kind of problem with unknown grouping structures, the

RJMCMC method is often used for the Bayesian implementation which is described in the next subsection.

2.2 Reversible Jump Markov Chain Monte Carlo

The RJMCMC algorithm was initially proposed for the Bayesian model determination problems [26]. But since then, it has been applied to many other problems such as the change point problems, the mixture problem, and the factorial experiments [26][31][32]. In the semiconductor manufacturing context, Bergeret and Gall [33] have applied this algorithm to detect the change point for the yield trend under the situation that the failure occurs at a problematic process stage and there are different yield performances before and after the failure time. Moreover, the RJMCMC algorithm has also been used with the CART method which leads to the Bayesian CART method [34]-[38].

We present the RJMCMC algorithm as introduced in Green [26]. Suppose that the competing models can be enumerable and are represented by the set of models $\mathcal{M} = \{M_1, M_2, \ldots\}$. Under the model M_k and given the data $\boldsymbol{y}$, the model parameter θ_k has the posterior distribution

$$p(\theta_k \mid \boldsymbol{y}, k) = p(\boldsymbol{y} \mid \theta_k, k)\, p(\theta_k \mid k)$$

where $p(\boldsymbol{y} \mid \theta_k, k)$ and $p(\theta_k \mid k)$ are the sampling distribution of $\boldsymbol{y}$ and the prior distribution of the parameter θ_k given the model M_k, respectively. The RJMCMC methods are an extension of the Metropolis-Hastings algorithm [39] that allows a Markov chain not only moves between different parameter values but also between models with different dimensions. This algorithm is designed to be reversible so as to maintain detailed balance of a irreducible and aperiodic chain that converges to the correct target posterior distribution. Please refer the reference [26] for the details of RJMCMC.

In the following, we only conceptually illustrate the jumping scheme between different models in the RJMCMC algorithm.

1. Propose a jump from the current Model M_k to the other model $M_{k'}$ with probability $J(M_k \to M_{k'})$.
2. Sample $\boldsymbol{v}$ from a proposal density $q(\boldsymbol{v} \mid \theta_k, k, k')$, where $\boldsymbol{v}$ is an augmented variable which plays the role of adjusting the unequal dimension problem between two models M_k and $M_{k'}$.
3. Set $(\theta_{k'}, \boldsymbol{v}') = g_{k,k'}(\theta_k, \boldsymbol{v})$, where $g_{k,k'}(\cdot)$ is called a bijection function between $(\theta_k, \boldsymbol{v})$ and $(\theta_{k'}, \boldsymbol{v}')$ which provides a one-to-one mapping between two sets of parameters $(\theta_{k'}, \boldsymbol{v}')$ and $(\theta_k, \boldsymbol{v})$ and two augmented variables ***v*** and ***v’*** play the roles of matching the dimensions of the

parameters for two model M_k and $M_{k'}$.

4. The RJMCMC moves from the current model M_k with parameter θ_k to another model $M_{k'}$ with parameter $\theta_{k'}$ according to the acceptance probability $r = \min\{1, A\}$ where

$$A = \frac{p(\boldsymbol{y} \mid \theta_{k'}, k')\, p(\theta_{k'})\, p(k')}{p(\boldsymbol{y} \mid \theta_k, k)\, p(\theta_k)\, p(k)} \frac{J(M_{k'} \to M_k)\, q(v' \mid \theta_{k'}, k', k)}{J(M_k \to M_{k'})\, q(v \mid \theta_k, k, k')} \left| \frac{\partial\, g_{k,k'}(\theta_k, v)}{\partial(\theta_k, v)} \right|.$$

Each iteration in the RJMCMC algorithm includes the above 4 steps and repeating such iterations constructs a Markov chain. Under some appropriate setting for the jumping scheme, this Markov chain will converge and its stationary distribution is identical to the target posterior distribution of the parameters for the selection problem among the model collection $\mathcal{M}$. The posterior distribution and posterior mode for each parameter can be estimated using the MCMC draws after the convergence is achieved. In particular, the model with the highest posterior distribution on $\mathcal{M}$ is considered as the best model based on the Bayesian approach [35] [36] [40].

For our tool partition problem, the set $\mathcal{M}$ consists of all possible partition models g with various degrees $\kappa = 1, 2, \ldots, J$. The RJMCMC algorithm helps us to determine the best partition among J tools and simultaneously estimate the model parameters under the hierarchical Bayesian model. Although we did not give the detail formulations for the functions involved in the jumping scheme (such as $J(M_k \to M_{k'})$, $g_{k,k'}(\cdot)$ and $q(\cdot\,|\,\theta_k, k, k')$ since they are usually problem-specific, the detail scheme for our tool partition problem will be described in Section 2.3.

Moreover, we suggest the usage of posterior distribution of possible partitions to realize the partitioning structure of tools instead of the usage of hierarchical tree structures

2.3 TCP Method for Tool Partitions

In Section 2.1, the posterior distribution of $(g,\theta,\mu,\sigma^2,\tau^2)$ given the data $\boldsymbol{y}$ has been derived up to a normalizing constant. The posterior distribution as well as the posterior mean for each parameter are estimated using the RJMCMC method. We are particularly interested in the posterior distribution for g which gives the best partition for tools. The details about how to apply RJMCMC to our tool partition problem in Section 2.2 which includes five move types:

1. updating the parameter of the group mean θ_j, for $j=1, 2\ldots J$,

2. updating the parameter of the group variance σ^2,

3. updating the hyperparameters τ^2 and μ_k, for $k=1, 2\ldots\kappa$,

4. updating the partition *g*, with “birth”, that is, splitting one group into two,

5. updating the partition g, with “death”, that is, combining two groups into one.

At each iteration in the chain, one of the above five moves is randomly selected regarding some pre-set probabilities p_1, p_2, p_3, p_4, p_5, where p_j is for the j-th move type, p_4 and p_5 usually depend on the degree of current partition g (i.e., κ) and $\sum_{j=1}^{5} p_j = 1$. Naturally, we set $p_5 = 0$ when $\kappa = 1$ and $p_4 = 0$ when κ is the maximum allowed value.

Given the partition g in each MCMC iteration, the parameter values are updated according to the conditional distributions of all parameters in the set Θ = $\{\theta_1, \ \theta_2 \ldots \ \theta_J, \ \sigma^2, \ \mu_1, \ \mu_2 \ldots \ \mu_\kappa$, and $\tau^2 \}$ via the Gibbs sampling [41] [42] (Appendix B). We will use the notation of $\Theta_{-\theta_j}$ to indicate the set of all parameters except the parameter of θ_j. The full conditional distributions of all parameters of Θ are given below and the detailed derivations for these conditional distributions can be found in Appendix C.

1. $\theta_j | \Theta_{-\theta_j} \sim$ Normal $(\frac{b_j}{2a_j}, \frac{1}{2a_j})$, (2)

where $b_j = (\frac{\mu_k}{\tau^2} + \frac{n_j \bar{y}_{.j}}{\sigma^2})$, $a_j = \frac{1}{2\tau^2} + \frac{n_j}{2\sigma^2}$, for $j \in S_k$, and j =1,2 ,…, J,

2. $\mu_k | \Theta_{-\mu_k} \sim$ Normal $(\frac{\sum_{j \in S_k} \theta_j}{w_k}, \frac{\tau^2}{w_k}) \mathrm{I}_{\mu_k}(a,b)$ where $w_k = \{\#$ of j in $S_k\}$,

3. $\tau^2 | \Theta_{-\tau^2} \sim$ Inverse Gamma $(\frac{\nu + J}{2}, \frac{\nu s^2 + F}{2})$, where

$$F = \sum_{j=1}^{J} \left(\theta_j - \sum_{k=1}^{\kappa} \mu_k I_{(j \in S_k)} \right)^2, \ g = \bigcup_{k=1}^{\kappa} S_k,$$

4. $\sigma^2 | \Theta_{-\sigma^2}$ ~Inverse Gamma ($\frac{\nu_1 + N}{2}, \frac{\nu_1 s_1^2 + E}{2}$),

where $E= \sum_{j=1}^{J} \sum_{i=1}^{n_j} (y_{ij} - \theta_j)^2$, and $N= \sum_{j=1}^{J} n_j$.

The structure parameter g is updated by the birth and death move types. The birth move means adding one group from the current partition g. In contrast, the death move means eliminating one group from the current partition g. We introduce the schemes for implementing these two move types below. Suppose that the current partition is $g^{(1)}$ with degree $\kappa^{(1)}$ and mean vector $\mu^{(1)}$. For the "birth" move type, we first choose a group to split randomly among those with at least two tools to form a new partition $g^{(2)}$ with degree $\kappa^{(2)}$= ($\kappa^{(1)}$+1) groups. At the same time, the mean vector $\mu^{(2)}$ needs a change for both the dimension and the value. The relocation can be simply done by adding a new random variable z that is independently distributed as Normal (μ_z, σ_z^2). In Section 2.4, we will discuss how to define μ_z and σ_z^2 in detail. Suppose that a proposal birth splits group S_k for $k \in \{1, 2\ldots\kappa^{(1)}\}$ into two groups S_{k_1} and S_{k_2}. Let μ_k be the current value and μ_{k_1}, μ_{k_2} be the new values for the two groups. Then we set

$$\mu_{k_1} = \mu_k + \frac{w_{k_2}}{w_k^{2}} z, \quad \mu_{k_2} = \mu_k - \frac{w_{k_1}}{w_k^{2}} z, \tag{3}$$

where w_i ={# of j in S_i } for i= k , k_1, and k_2, with w_k = $w_{k_1} + w_{k_2}$, $S_{k_1} \cup S_{k_2} = S_k$, and $S_{k_1} \cap S_{k_2} = \phi$. So we have $g^{(2)}=\{ S^{(1)}{}_{-S_k} \} \cup \{ S_{k_1}, S_{k_2} \}$ and

$\mu^{(2)} = (\mu^{(1)}_{-\mu_k}, \mu_{k_1}, \mu_{k_2})$. Similarly, for a "death" move, we will randomly choose two groups S_{k_1} and S_{k_2} from the current partition $g^{(2)}$ and merge them into a new group S_k to generate a new partition $g^{(1)}$ with $\mu_k = \dfrac{w_{k_1}\mu_{k_1} + w_{k_2}\mu_{k_2}}{w_k}$, $w_k = w_{k_1} + w_{k_2}$, and $z = (\mu_{k_1} - \mu_{k_2})\, w_k$ by solving the simultaneous equations in (3).

After specifying the jumping proposal, the acceptance probabilities for the birth and death in the RJMCMC algorithm are min $\{1,A\}$ and min$\{1,1/A\}$ respectively, where

$$A = \frac{P(g^{(2)})}{P(g^{(1)})} e^{\frac{-1}{2\tau^2}\left\{\sum_{j=1}^{J}\left(\theta_j - \sum_{k=1}^{\kappa}\mu_k^{(2)} I_{\left(j\in S_k^{(2)}\right)}\right)^2 - \sum_{j=1}^{J}\left(\theta_j - \sum_{k=1}^{\kappa}\mu_k^{(1)} I_{\left(j\in S_k^{(1)}\right)}\right)^2\right\}} \times \frac{1}{(b-a)} \times \frac{\prod_{k=1}^{\kappa^{(2)}} \mathrm{I}_{\mu_k^{(2)}}(a,b)}{\prod_{k=1}^{\kappa^{(1)}} \mathrm{I}_{\mu_k^{(1)}}(a,b)} \frac{P_{death}}{P_{birth}}$$

$$\times \ \text{\# of } \{ k; w_k \geq 2,\ k=1,2,\ldots,\kappa^{(1)} \} \times \frac{1}{\kappa^{(1)}(\kappa^{(1)}+1)} \{2^{w_k} - 2\} \frac{1}{w_k} \frac{1}{f(z)},$$

where $\kappa^{(i)}$ is the degree of $g^{(i)}$, for i=1,2, with $\kappa^{(2)} = \kappa^{(1)} + 1$, $g^{(i)} = \bigcup_{k=1}^{\kappa^{(i)}} S_k{}^{(i)}$, w_k is the number of tools in $S_k{}^{(1)}$, P_{death}, and P_{birth} are the proposal probability for the death move type and birth move type respectively (i.e., p_4 and p_5 in the earlier context), $\dfrac{1}{w_k}$ is the Jacobian, and $f(z)$ is the density function of z.

The derivations for the acceptance probabilities of the birth and death move types are given in Appendix D for details.

2.4 Guidelines for Choosing Initial Values for Parameters and Hyperparameters in TCP

The engineers can set up the initial parameters and hyperparameters in the priors based on their knowledge for each problem. However, when there is no enough prior information, we will suggest some guidelines to set up the priors to facilitate the practice of TCP in the semiconductor industry and in other applications. The details of our guidelines include the following:

1. μ_k ~ Uniform (a, b), for $k=1,2,\ldots, K$, let a=min {$\overline{y_{.j}}$: $j=1,2,\ldots,J$} and b=max{$\overline{y_{.j}}$, $j=1,2,\ldots,J$ }. The reason is described as follows: Since these θ_j's belong to the same group S_k , these θ_j s will be distributed as the same normal distribution , (that is, θ_j ~ Normal (μ_k , τ^2), for $j \in S_k$, j=1,2,…, J .) On the contrary, if θ_{j_1} and θ_{j_2} belong to two different groups S_{k_1} and S_{k_2} , θ_{j_1} and θ_{j_2} are distributed as two different normal distribution with two different means μ_{k_1} and μ_{k_2} . Therefore, in tool comparison problem, μ_k can represent the performances of different tool groups, and (a, b) represents the range of tool performances among tools. So, we suggest to use the empirical estimate for the range, (min { $\overline{y_{.j}}$ }, max{ $\overline{y_{.j}}$ }) for the prior specification.

2. τ^{2} ~ scaled inverse chi-squared (ν, s^2) with $\nu = 20$ and $s^2 = \left(tolerance/6\right)^2$.

3. σ^2~ scaled inverse chi squared (ν_1, s_1^{2}) with $\nu_1 = 20$ and $s_1^{2} =$ sample variance of { $(y_{ij} - \overline{y_{.j}})$, *j=1, 2…J, i=1, 2…* n_j }. Note that, s_1^{2} is an unbiased estimator of σ^2.

The values of ν and ν_1 are only for recommendation, as they only impacts the convergence speed of the TCP method. Based on our experience, since the population mean $\frac{\nu}{\nu-2}s^2$ of the scaled inverse chi-squared distribution (ν, s^2) is very close to its mode $\frac{\nu}{\nu+2}s^2$ and the variance $\frac{2\nu^2}{(\nu-2)^2(\nu-4)}s^4$ is small when ν and ν_1 are large than 20, the prior distribution of τ^{2} will produce more τ^{2}s with values closed to *(tolerance/*6*)*^{2}. This effect usually speeds up the convergence for the Markov chain toward our target distribution based on the tolerance control.

Following the above guidelines, the engineers only need to determine the numerical level of the *tolerance* parameter. Since the tolerance concept is widely used in the semiconductor industry and in other applications, it is not difficult but friendly for the engineers to make this setup. For example, there exist some unavoidable errors from the metrology systems and minor deviations with respect to the product specifications. Hence, one can set up the *tolerance* based on the engineers' knowledge, the product specifications, and the tool limitations. In Section 3.2, we present some simulation results of the sensitivity analysis to show that the TCP method could generate the optimal partitions with respect to different levels of *tolerances*.

2.5 Convergence Assessment for RJMCMC

Before conducting the Bayesian inference using RJMCMC samples, the output should be analyzed to determine the required run length for the MCMC sequences. Gelman and Rubin [43] proposed a convergence diagnostic, the potential scale reduction factor (PSRF), obtained by running multiple I chains with overspread starting values. Books and Gelman [44] provided a generalization of Gelman and Rubin’s method that considers several parameters simultaneously. For a Bayesian model selection, Brooks and Giudici [45] suggested selecting a scale summary of parameters and decomposing its variance within the RJMCMC simulation output into two distinct groups, within and between chains, to monitor the convergence.

The convergence check for TCP method proceeds as follows. We begin with simulating five independent chains (I=5) of length $T_1 + T_2$, each starts with different initial values which are overspread. After discarding the first T_1 iterations for burn-in and retaining only the last T_2 iterations, we first compute the -2(log-likelihood) value L_i^t observed for the *i*-th chain and up to the *t*-th iteration according to the equation (1). Then, we calculate the following 6 variation quantities:

$\hat{V}$: the total variance of L_i^t ,

$\hat{W_c}$: the averaged within-chain variance of L_i^t ,

$\hat{B_m}$: the estimated between-model variance of L_i^t ,

$\widehat{B_m W_c}$: the estimated between-model and within-chain variance of L_i^t ,

$\hat{W_m}$: the estimated within-model variance of L_i^t ,

$\widehat{W_m W_c}$: the estimated within-model and within- chain variance of L_i^t .

For convenience, we let l_{im}^k be the value of L_i^t corresponding to the k th observation of the m th model in the i th chain for k =1,2,…, $K(i,m)$, m = 1, 2,…,M, and i =1,2,…,I, where $K(i,m)$ denotes the number of times that the m th model is observed in the i -th chain and M is the number of models in I chains. By definition, we have $\sum_{i=1}^{I}\sum_{m=1}^{M}K(i,m)=T_2$. The expressions for $\hat{V}$, $\hat{W_c}$, $\hat{B_m}$, $\widehat{B_m W_c}$, $\hat{W_m}$, and $\widehat{W_m W_c}$ are listed below:

$$\hat{V} = \sum_{i=1}^{I}\sum_{i=T_1+1}^{T_{1+}T_2}(L_i^t - \bar{L}_{\cdot}^{\cdot})^2 \Big/ (IT_2 - 1), \quad \text{where } \bar{L}_{\cdot}^{\cdot} = \sum_{i=1}^{I}\sum_{i=T_1+1}^{T_{1+}T_2} L_i^t \Big/ IT_2 ,$$

$$\hat{B}_m = \sum_{m=1}^{M}(\bar{l}_{\cdot m}^{\cdot} - \bar{l}_{\cdot\cdot}^{\cdot})^2 \Big/ (M-1) ,$$

$$\hat{W}_m = \frac{1}{M}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)}(l_{im}^k - \bar{l}_{\cdot m}^{\cdot})^2 \Big/ (K_m - 1) ,$$

$$\hat{W}_c = \frac{1}{I}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)}(l_{im}^k - \bar{l}_{i.}^{\cdot})^2 \Big/ (K_i - 1) ,$$

$$\widehat{B_m W_c} = \frac{1}{I}\sum_{i=1}^{I}\sum_{m=1}^{M}(\bar{l}^{\cdot}_{im} - \bar{l}^{\cdot}_{i.})^{2} \Big/ (M-1)\,,$$

$$\widehat{W_m W_c} = \frac{1}{IM}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)}(l^{k}_{im} - \bar{l}^{\cdot}_{im})^{2} \Big/ (K(i,m)-1)\,, \qquad (4)$$

where $K_i = \sum_{m=1}^{M} K(i,m)$, $K_m = \sum_{i=1}^{I} K(i,m)$,

$$\bar{l}^{\cdot}_{.m} = \frac{1}{K_m}\sum_{i=1}^{I}\sum_{k=1}^{K(i,m)} l^{k}_{im}\,,\quad \bar{l}^{\cdot}_{i.} = \frac{1}{K_i}\sum_{i=1}^{I}\sum_{k=1}^{K(i,m)} l^{k}_{im}\,,$$

$$\bar{l}^{\cdot}_{im} = \frac{1}{K(i,m)}\sum_{k=1}^{K(i,m)} l^{k}_{im}\,,\quad \bar{l}^{\cdot}_{..} = \frac{1}{IT}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)} l^{k}_{im}\,.$$

We follow the method by Brooks and Giudici [45] for our convergence assessment. By monitoring the plots of three pairs, $\hat{V}$ v.s. $\hat{W}_c$, $\hat{B}_m$ v.s. $\widehat{B_m W_c}$ and $\hat{W}_m$ v.s. $\widehat{W_m W_c}$, across iterations, the convergence is achieved among multiple chains when two lines in each plot get closer and stable after some iterations. Consequently, the TCP method achieves the convergence after that iteration.

3. Simulation Studies

To illustrate the performance for the TCP method, we provide three simulation cases in this section. In Section 3.1, we show the limited impacts for the unbalanced usage in the manufacturing. In Section 3.2, we report the sensitivity analysis by using different tolerance controls in the TCP method and compare the results to the pruning results using CART. In Section 3.3, we perform a robustness analysis by simulating the data from a location lognormal distribution to investigate the impact of the normal assumption in the TCP method for non-normal data.

In these simulation cases, we follow the guidelines of choosing initial values for parameters and hyperparameters given in Section 2.4 to show the feasibility for future applications in the semiconductor industry. For convergence assessment, we follow the method in Section 2.5 to monitor the convergence for the RJMCMC by running 5 independent parallel chains with 50,000 iterations. In order to save computational expenses, we calculate $\hat{V}$, $\hat{W}_c$, $\hat{B}_m$, $\widehat{B_mW_c}$, $\hat{W}_m$, and $\widehat{W_mW_c}$ for every 100 iterations. The simulation programs are written by R (in Splus) and run in a PC with 1G CPU and 512M RAM. It takes about 70 minutes to complete a case. The simulation results for each case are summarized in the following subsections.

3.1 Unbalanced Data for Unbalanced Tool Usage

We present three unbalanced data cases to illustrate the influences on the TCP method when different usages exist among five tools. Following the notations defined in Section 2.2, the simulation models are described in detail as follows: y_{ij} ~ Normal (θ_j, σ^2), where i=1, 2…n_j and j=1, 2,…, 5. $(n_1, n_2, n_3, n_4, n_5)$ and $(\theta_1, \theta_2, \ldots, \theta_5)$ are the numbers of observations and the yield means for the five tools respectively.

Case I: $(n_1, n_2, n_3, n_4, n_5)$=(15, 150, 20, 250, 20)

$(\theta_1, \theta_2, \ldots, \theta_5)$=(3, 3, 4, 4, 7)

σ=1.5

In Case I, the true partition for 5 tools is {(T1, T2), (T3, T4), T5}, which is denoted as (11223). The box plot and the related statistics for the simulated data are given in Figure 3.1 and Table 3.1.

Table 3.1. The sample mean, standard deviation, and count for each tool for the simulated data in Case I.

Tool	Mean	Std	Count
T1	2.78	1.58	15
T2	2.88	1.54	150
T3	3.78	1.5	20
T4	3.79	1.55	200
T5	7.23	1.4	20

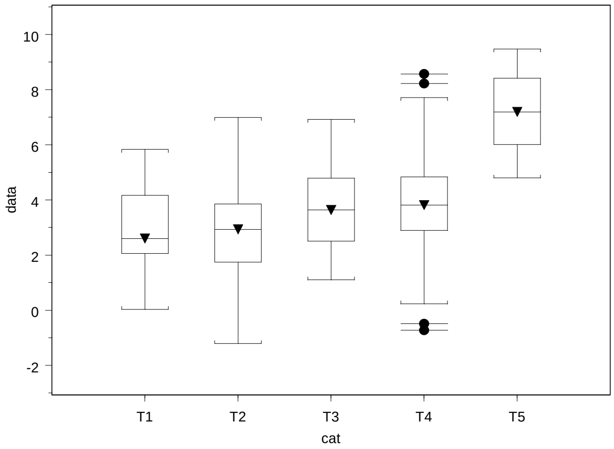


Figure 3.1. Box plots by tools for the simulated data in Case I.

We apply the TCP method to the simulated data in Case I by setting the *tolerance* to be 1. The three convergence plots, including $\hat{V}$ v.s. $\hat{W}_c$, $\hat{B}_m$ v.s. $\hat{B}_m W_c$, and $\hat{W}_m$ v.s. $\hat{W}_m W_c$, are displayed in Figure 3.2. We collect the partition results from the last 10,000 iterations to estimate the posterior distribution of the partition. The estimated posterior distribution for the partition is given in Figure 3.3. The correct partition (denoted as (11223)) is the mode of the estimated posterior distribution with the probability 0.49. From this simulation, we found that the impact of unbalanced data is very limited.

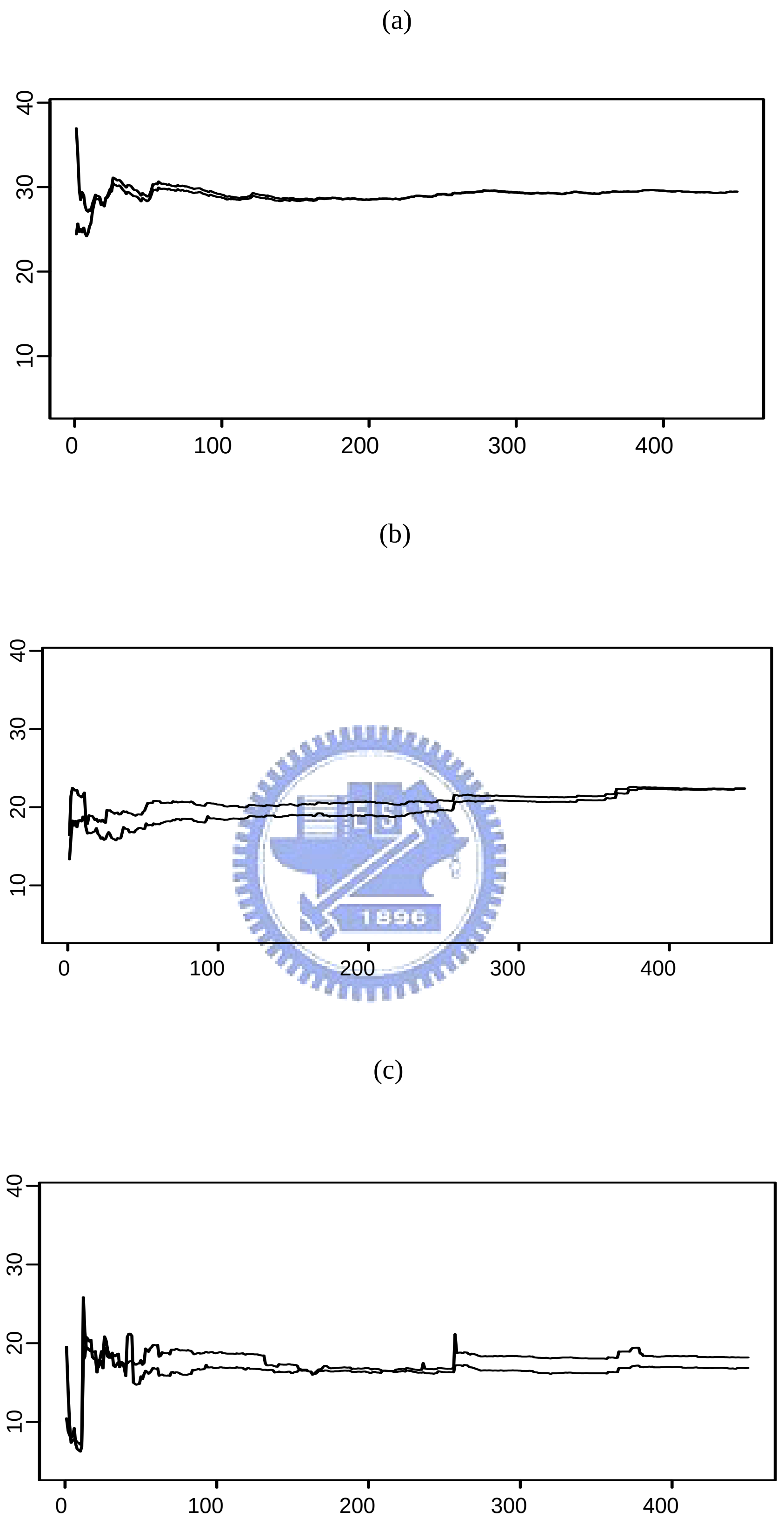


Figure 3.2. Convergence assessment plots for the simulated data in Case I :
(a) $\hat{V}$ *vs.* $\hat{W}_c$ *, (b)* $\hat{W}_m$ *vs.* $W_m\hat{W}_c$ *, and (c)* $\hat{B}_m$ *vs.* $B_m\hat{W}_c$
(one unit in the x-axis is 100 iterations)

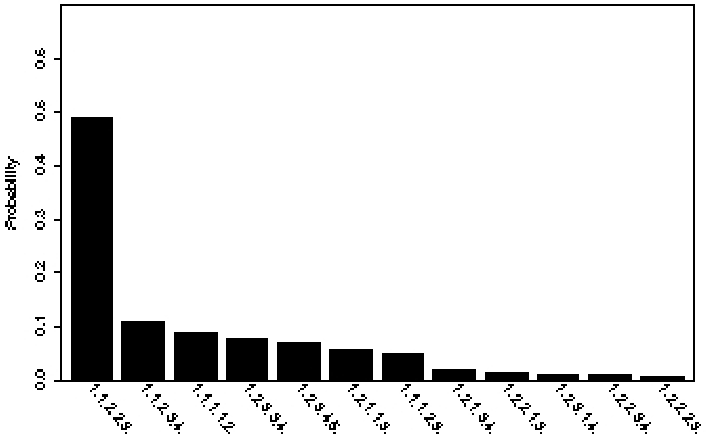


Figure 3.3. Estimated posterior distribution of partition for the simulated data in Case I (and the partition with probability less than 0.005 is not shown).

Table 3.2. Partitioning results with respect to different values of cost complexity in the CART model for the simulated data in Case I.

Partitioning result	Cost-complexity
(1,2,3,4,5)	0
(1,1,2,2,3)	1
(1,1,1,1,2)	90

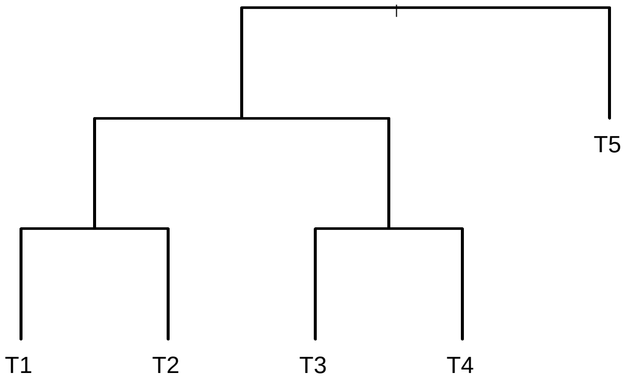


Figure 3.4. The tree obtained by the CART model for the simulated data in Case I.

We also analyze the same data by the CART method for comparison which is implemented by the S-Plus functions of tree and prune.tree [46]. The complete tree result is given in Figure 3.4. and the best partitioning results with respect to different cost-complexities are given in Table 3.2. According to Table 3.2, one should select a correct cost-complexity to get reasonable tree results when applying the CART method. However, it is hard to link the cost complexity with the concept of tolerance control in engineering. For this reason, the TCP method is much easier for engineers to use in the connections with engineering tolerance controls.

Case II: $(n_1, n_2, n_3, n_4, n_5)$=(20, 40, 80, 160, 320)

$(\theta_1, \theta_2, \ldots, \theta_5)$=(5, 3, 5, 3, 3)

σ =1.

Similar to Case I, the box plots by tools and the related statistics for the simulated data are given in Figure 3.5 and Table 3.3. There true partition among 5 tools is denoted as (12122). By applying TCP method with *tolerance* =1 and under similar setups as those in Case I, the estimated posterior distribution for the partition is given in Figure 3.6 and the correct partition is the posterior mode with probability 0.71. For the same data, the complete tree resulted by CART is given in Figure 3.7, and the best partitioning results with respect to different cost-complexities are given in Table 3.4.

Table 3.3. The sample mean, standard deviation, and count for each tool for the simulated data in Case II.

Tool	Mean	Std	Count
T1	4.25	0.84	20
T2	2.86	0.94	40
T3	4.12	0.94	80
T4	3.16	1.04	160
T5	2.99	1	320

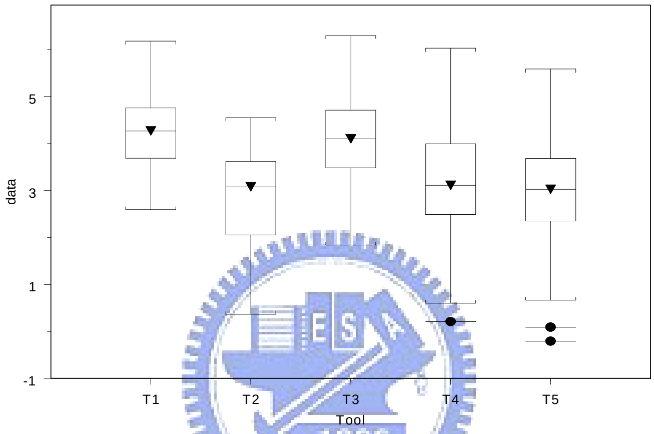


Figure 3.5. Box plots by tools for the simulated data in Case II.

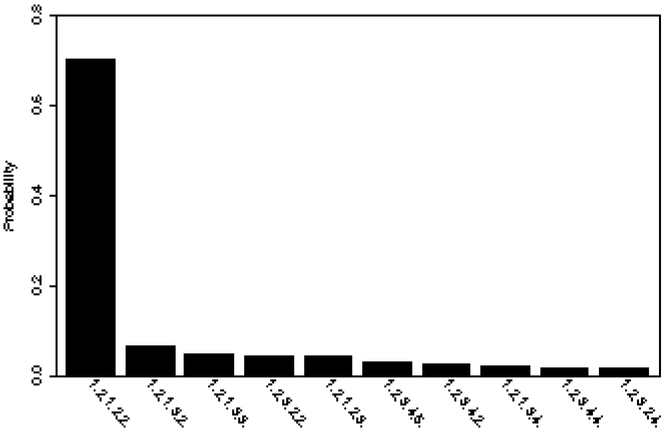


Figure 3.6. Estimated posterior distribution of partition for the simulated data in Case II (the partition with probability less than 0.005 is not shown)

Table 3.4. Partitioning results with respect to different values of cost complexity in the CART model for the simulated data in Case II.

Partitioning result	Cost-complexity
(1,2,3,4,5)	0
(1,2,1,3,3)	1
(1,2,1,2,2)	4

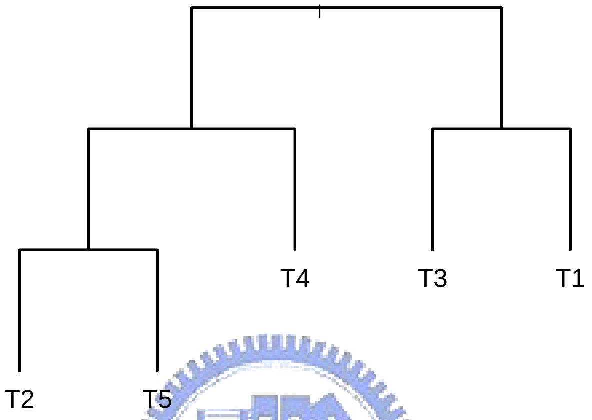


Figure 3.7. The tree obtained by the CART model for the simulated data in Case II.

Case III: $(n_1, n_2, n_3, n_4, n_5)$=(10, 50, 50, 100, 15)

$(\theta_1, \theta_2, \ldots, \theta_5)$=(5, 3, 3, 3, 3)

σ =1

The box plot and the related statistics for the simulated data for Case III are given in Figure 3.8 and Table 3.5. The true partition is denoted as (12222) among 5 tools. Under the similar setups for the TCP method with *tolerance* =1, the estimated posterior distribution of the partition is given in Figure 3.9 and again the correct partition is the posterior mode with the probability 0.71. The complete tree result using CART is given in Figure 3.10, and the best partitioning results with respect to different cost-complexities are given in Table 3.6 for comparison.

Table 3.5. The sample mean, standard deviation, and count of each tool for the simulated data in Case III.

Tool	Mean	Std	Count
T1	5.52	0.78	10
T2	2.76	0.82	50
T3	2.88	0.93	50
T4	3.07	0.93	100
T5	3.07	0.77	15

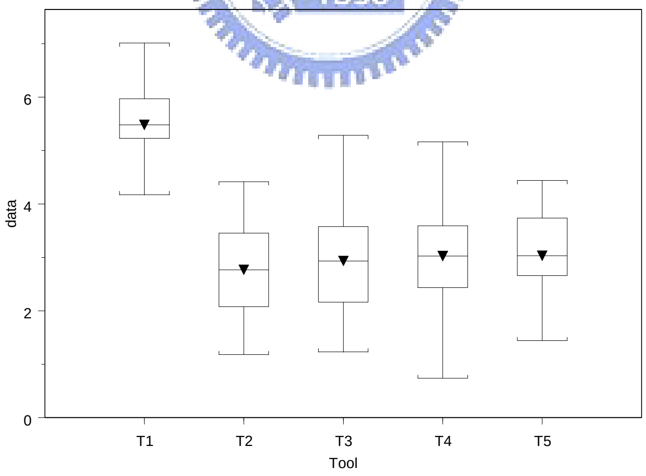


Figure 3.8. Box plots by tools for the simulated data in Case III.

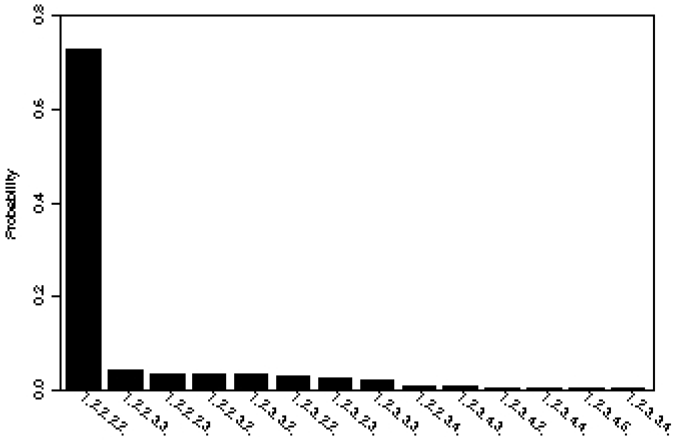


Figure 3.9. Estimated posterior distribution of partition for the simulated data in Case III and (the partition with probability less than 0.005 is not shown).

Table 3.6. Partitioning results with respect to different values of cost complexity in the CART model for the simulated data in Case III.

Partitioning result	Cost-complexity
(1,2,3,4,5)	0
(1,2,2,3,3)	1
(1,2,2,2,2)	4

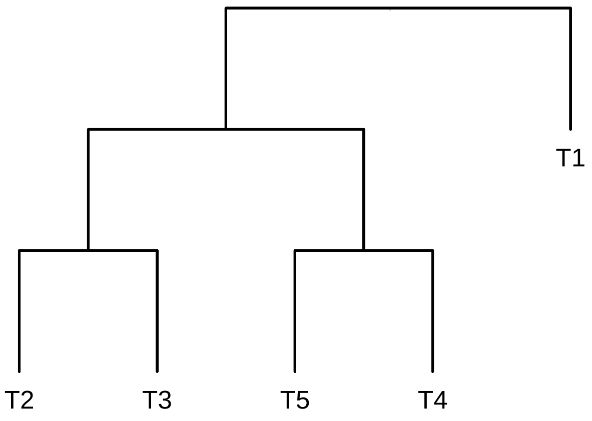


Figure 3.10. The tree obtained by the CART model for the simulated data in Case III.

3.2 Sensitivity Analysis with Different Tolerance Controls for TCP and the Comparison with CART

In the sensitivity analysis, the data are generated from the same model as Case I in Section 3.1, but the number of observations is 30 for each tool and σ is changed to be 1. The box plots by tools and the related statistics of the simulated data are given in Figure 3.11 and Table 3.7. The true partition is {(T1, T2), (T3, T4), T5} (denoted as (11223)). Here, we apply the TCP method with different values of *tolerance* (= 0.5, 1, 2, 3, 4, 5, 6) to examine its influence on the partition results of the TCP method.

Table 3.7. The sample mean, standard deviation, and count of each tool for the simulated data in Case I with equal sample size.

Tool	Mean	Std	Count
T1	2.90	0.75	30
T2	3.12	0.93	30
T3	3.88	0.99	30
T4	3.86	0.98	30
T5	6.82	1.25	30

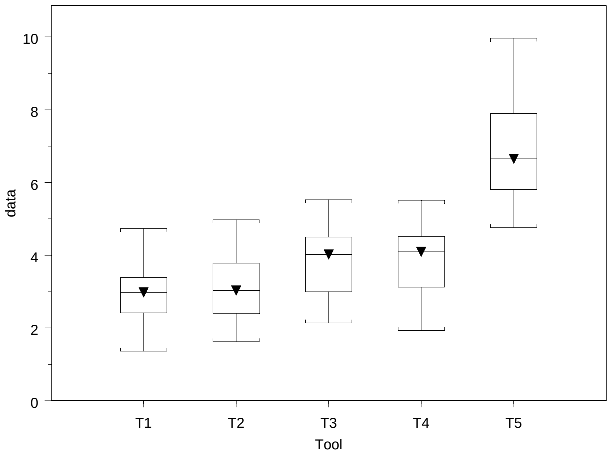


Figure 3.11. Box plots by tools for the simulated data in Case I with equal sample size.

The partition results under various *tolerance* levels are given in Table 3.8. The numbers presented in the table are the posterior probability for each possible partition and the corresponding estimated error (shown in the parentheses) based on 30 realizations under each *tolerance* specification. Now, the group means are 3, 3, 4, 4, and 7 and the within-group standard deviations are 1 in this simulation. When the *tolerance* is 0.5 or 1, the target (11223*) will be the most plausible partition because the between group differences can be as large as 1. When the *tolerance* is 2, 3, 4, or 5, the most plausible partition becomes (11112**) because the between group differences can be as large as 4 and within group standard deviation is 1. When the *tolerance* is 6, the most plausible partition moves to (11111***) because the between group differences are smaller than 6. In the table, we can see that the standard errors for the most plausible partition are very small, demonstrating the robustness of the posterior modes. Another interesting phenomenon is that the averages of the posterior probabilities of three partitions, {11223*}, {11112**} and {11111***}, changes according to the level of *tolerance*. Intuitively, tools tends to be merged if the *tolerance* is large. For this particular case, the averaged posterior probability for the partition {11223*} decreases when the *tolerance* increases. On the other hand, the averaged posterior probability for the partition {11112**} increases when the *tolerance* increases from 0.5 to 3 but decreases when the *tolerance* increases from 3 to 6. Similarly, the averaged posterior probability for the partition {11111***} increases when the *tolerance* increases. Hence, the posterior distributions of the partitions indeed reflect the levels of tolerance controls. Beside the most plausible partitions, the posterior distribution also reveals the next plausible partition with the strength of plausibility (i.e., the posterior probability). This useful information can only be provided by the Bayesian approach. The results of the sensitivity analysis provide the evidence that the partitioning results of TCP will be affected by the level of tolerance controls.

Similarly, based on the same data, the complete tree result using CART is given in Figure 3.12, and the best partitioning results with respect to different cost-complexities are given in Table 3.9. It is evident that Table 3.8 contains more information than Table 3.9.

Table 3.8. Averaged posterior probabilities and their standard errors (in the parentheses) for the partition results in the sensitivity analysis for TCP method with respect to different tolerances.

	Tolerance=0.5	**Tolerance=1**	**Tolerance=2**	**Tolerance=3**	**Tolerance=4**	**Tolerance=5**	**Tolerance=6**
11111* **	0(0)	0(0)	0(0)	0.004(0.0051)	0.050(0.0134)	0.239(0.0223)	**0.444(0.0181)**
11112 **	0.005(0.0056)	0.079(0.0176)	**0.408(0.0142)**	**0.495(0.0145)**	**0.450(0.0157)**	**0.300(0.0168)**	0.159(0.0132)
11122	0(0)	0(0)	0(0)	0.002(0.0014)	0.013(0.0031)	0.027(0.0040)	0.032(0.0042)
11123	0.002(0.0031)	0.018(0.0050)	0.047(0.0048)	0.050(0.0044)	0.046(0.0037)	0.033(0.0024)	0.020(0.0024)
11212	0(0)	0(0)	0(0)	0.003(0.0020)	0.013(0.0038)	0.030(0.0048)	0.033(0.0056)
11213	0.003(0.0038)	0.021(0.0067)	0.052(0.0068)	0.052(0.0064)	0.047(0.0034)	0.032(0.0032)	0.019(0.0029)
11222	0(0)	0(0)	0(0)	0(0)	0.0041(0.0015)	0.012(0.0024)	0.017(0.0031)
11223*	**0.683(0.0230)**	**0.542(0.0321)**	0.180(0.0137)	0.093(0.0084)	0.062(0.0060)	0.038(0.0048)	0.020(0.0026)
11234	0.123(0.0130)	0.118(0.0063)	0.057(0.0061)	0.035(0.0042)	0.029(0.0037)	0.019(0.0022)	0.012(0.0018)
12112	0(0)	0(0)	0(0)	0(0)	0.002(0.0011)	0.007(0.0020)	0.011(0.002)
12113	0(0)	0.006(0.0026)	0.026(0.0034)	0.031(0.0029)	0.032(0.0023)	0.024(0.0036)	0.015(0.0018)
12123	0(0)	0.004(0.0016)	0.0233(0.0039)	0.034(0.0037)	0.035(0.0036)	0.026(0.0033)	0.016(0.0027)
12134	0(0)	0.004(0.0015)	0.013(0.0019)	0.016(0.0020)	0.017(0.0020)	0.013(0.0020)	0.009(0.0016)
12213	0(0)	0.003(0.0015)	0.023(0.0029)	0.035(0.0028)	0.037(0.0033)	0.026(0.0035)	0.016(0.0032)
12221	0(0)	0(0)	0(0)	0(0)	0.001(0.0007)	0.006(0.0015)	0.009(0.0018)
12223	0.009(0.0072)	0.028(0.0078)	0.041(0.0059)	0.037(0.0042)	0.034(0.0037)	0.024(0.0033)	0.014(0.0022)
12234	0.004(0.0024)	0.010(0.0027)	0.017(0.0026)	0.017(0.0023)	0.017(0.0017)	0.014(0.0017)	0.009(0.0017)
12314	0.001(0.0008)	0.004(0.0016)	0.013(0.0021)	0.016(0.0022)	0.017(0.0018)	0.013(0.0024)	0.009(0.0017)
12324	0.005(0.0027)	0.011(0.0032)	0.018(0.0021)	0.018(0.0025)	0.017(0.0021)	0.013(0.0025)	0.009(0.0016)
12334	0.096(0.0115)	0.079(0.0080)	0.034(0.0029)	0.0216(0.0025)	0.018(0.0027)	0.013(0.002)	0.008(0.0018)
12345	0.070(0.0092)	0.073(0.0055)	0.047(0.0048)	0.039(0.0038)	0.040(0.0037)	0.031(0.0038)	0.023(0.0029)

Table 3.9. The different partitioning results with respect to different values of cost complexity in the CART model.

Partitioning result	Cost-complexity
(1,2,3,4,5)	0
(1,1,2,2,3)	5
(1,1,1,1,2)	25

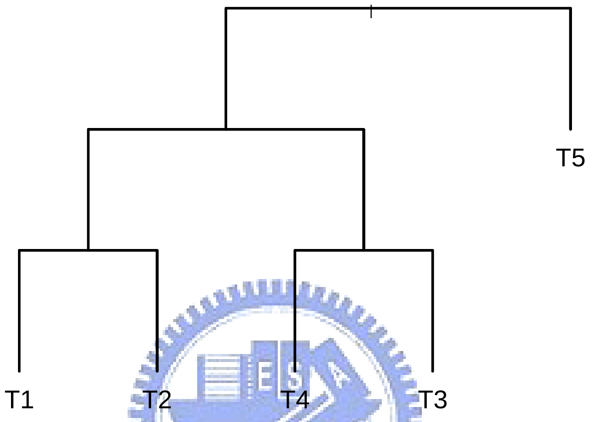


Figure 3.12. A tree obtained by the CART model for the simulated data in the sensitivity analysis.

3.3 Robustness Studies by Balanced Simulation Data with Mean Shifts

In the semiconductor industry, the tool performance often follows a baseline distribution (for example, a normal distribution) when the tools are in control. On the contrary, the mean shifts often occur when the tools are out of control. Therefore, without loss of generality, we generate two cases with different kinds of yield baseline distribution to verify the robustness of the TCP method subject to non-normal data.

3.3.1 Mean Shifts for Lognormal Distribution

The simulation models are described as follows: $y_{ij} \sim \theta_j$ + Lognormal (0,1), where i=1,2,…,n_j and j=1,2,…,5, with $(n_1, n_2, n_3, n_4, n_5)$ = (30, 30, 30, 30, 30), $(\theta_1, \theta_2, \ldots, \theta_5)$=(0, 0, 3, 3, 7). In this experiment, there are 3 groups in 5 tools and the true partition is {(T1, T2), (T3, T4), T5} (denoted as, (1,1,2,2,3)). The box plots by tools and the related statistics for the simulated data are given in Figure 3.13 and Table 3.10.

Table 3.10. The sample mean, standard deviation, and count for each tool for the simulated lognormal data in the robustness study.

Tool	Mean	Std	Count
T1	1.51	1.23	30
T2	1.21	1.22	30
T3	4.24	1.63	30
T4	4.45	1.27	30
T5	8.59	1.98	30

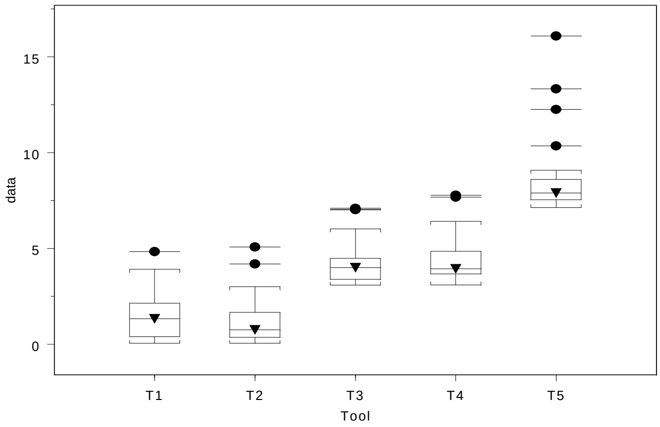


Figure 3.13. Box plots by tools for the simulated lognormal data in the robustness study.

We apply the TCP methods with *tolerance* =1 to get the posterior distribution of partitions given in Figure 3.14. The true partition {(T1, T2), (T3, T4), T5} (denoted as, (1,1,2,2,3)) is the posterior mode with the probability 0.7226. For this experiment, we still get correct partition result although the simulated data violate the normal assumption in the TCP method.

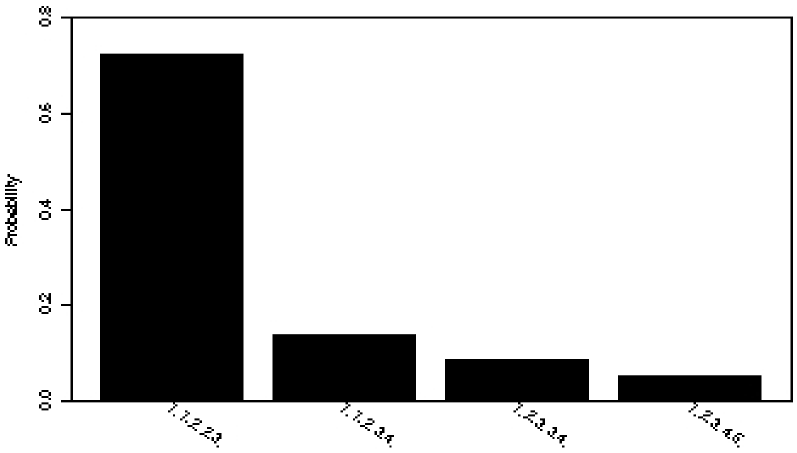


Figure 3.14. Estimated posterior distribution of partition for the lognormal data in the robustness study (the partition with probability less than 0.005 is not shown)

3.3.2 Mean Shifts for t Distribution

The simulation models are described as follows:

$y_{ij} \sim \theta_j +$ (t distribution with the degrees of freedom 5),

where i=1,2,…,n_j and j=1,2,…,5, with $(n_1, n_2, n_3, n_4, n_5)$ = (30, 30, 30, 30, 30), $(\theta_1, \theta_2, \ldots, \theta_5)$=(3, 3, 4, 4, 7). There are 3 groups in 5 tools and the true partition is {(T1, T2), (T3, T4), T5} (denoted as, (1,1,2,2,3)). The box plots by tools and the related statistics for the simulated data are given in Figure 3.15 and Table 3.11. Using the TCP methods with *tolerance* =1, the posterior distribution of partition for this t-distributed data set is obtained in Figure 3.16 and the posterior mode is still the correct partition {(T1, T2), (T3, T4), T5} (denoted as, (1,1,2,2,3)) with the probability 0.70364. Again, the TCP method gives a good partition result even when the data are not normal with mean shifts.

Table 3.11. The sample mean, standard deviation, and count for each tool for the simulated t-distributed data in the robustness study.

Tool	Mean	Std	Count
T1	2.81	1.56	30
T2	3.09	1.17	30
T3	4.21	0.98	30
T4	4.05	1.4	30
T5	7.25	0.98	30

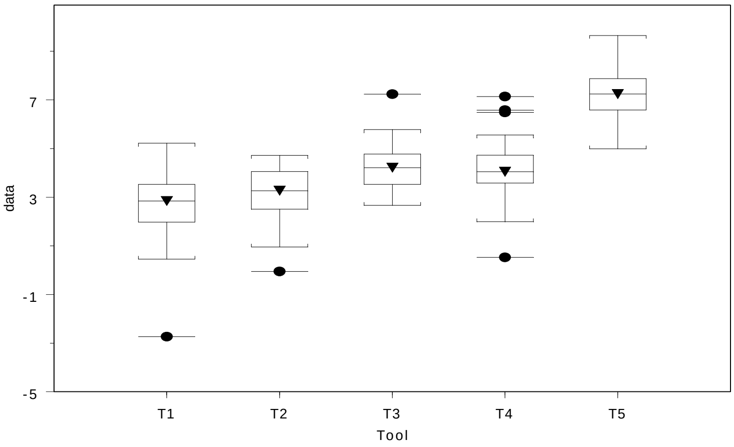


Figure 3.15. Box plots by tools for the simulated t-distributed data in the robustness study.

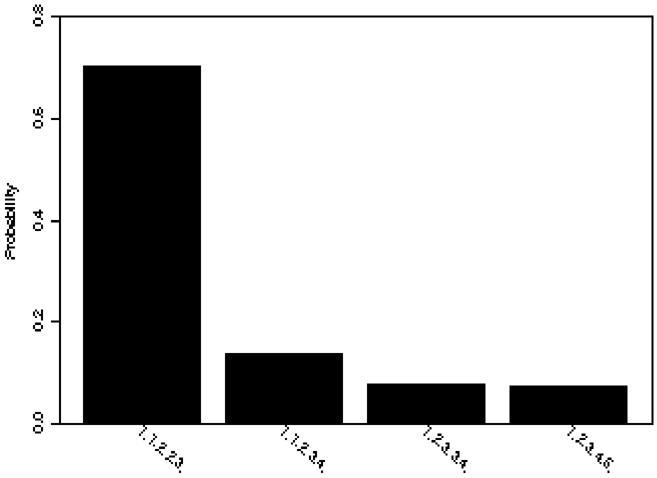


Figure 3.16. Estimated posterior distribution of partition for the t-distributed data in the robustness study (the partition with probability less than 0.005 is not shown)

4. Two Applications in the Semiconductor Industry

Two real applications in the semiconductor industry are illustrated to show the effectiveness of the proposed TCP method for improving the product quality. In Section 4.1, we first demonstrate an application about the yield enhancement by detecting tool differences. In Section 4.2, we apply the TCP method to the problem related to Cp/Cpk enhancement. For this particular application, we also describe a possible implication of the TCP method to another related problem. All data are from a semiconductor company in Taiwan.

4.1 Ramp Up Yield Using The TCP Method

As introduced in Chapter 1, semiconductor manufacturing has a very long process cycle including 150-400 process steps to complete the entire manufacturing process. After completing all process steps, each lot is inspected via WAT, WST and FT (final test) with approximately 100 test items for each inspection test. We analyze the “Srow” measurement for each lot which is one of the key test items in wafer sort testing. A larger value of the Srow measurement indicates a worse yield performance. The considered the Srow data consist of 439 lots with the sample mean 5.98 and the sample standard deviation 1.85. For this Srow measurement, the engineers have found 52 suspected steps from 221 process steps by performing ANOVA for tool comparison for each process step. In particular, the 10th process step is one of the suspected steps. The box plots and the related statistics of the Srow measurements for various tools in the 10th step are shown in Figure 4.1 and Table 4.1, respectively. It clearly shows that two tools, SPU03 and SPU05, have relatively worse performance at this problematic step.

Table 4.1. The sample mean, sample standard deviation, and the counts for the Srow measurements for various tools in the 10th process step.

Tool	Mean	Std	Count
SPU03	9.05	1.85	14
SPU04	5.79	1.51	48
SPU05	7.17	1.27	96
SPU07	5.54	1.4	36
SPU14	5.83	1.7	93
SPU16	5.16	1.78	152

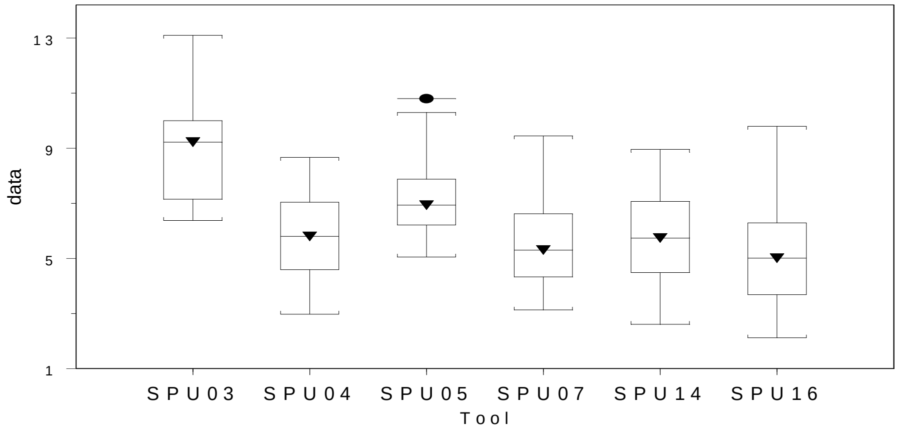


Figure 4.1. Box plots of the Srow measurements for various tools in the 10th process step.

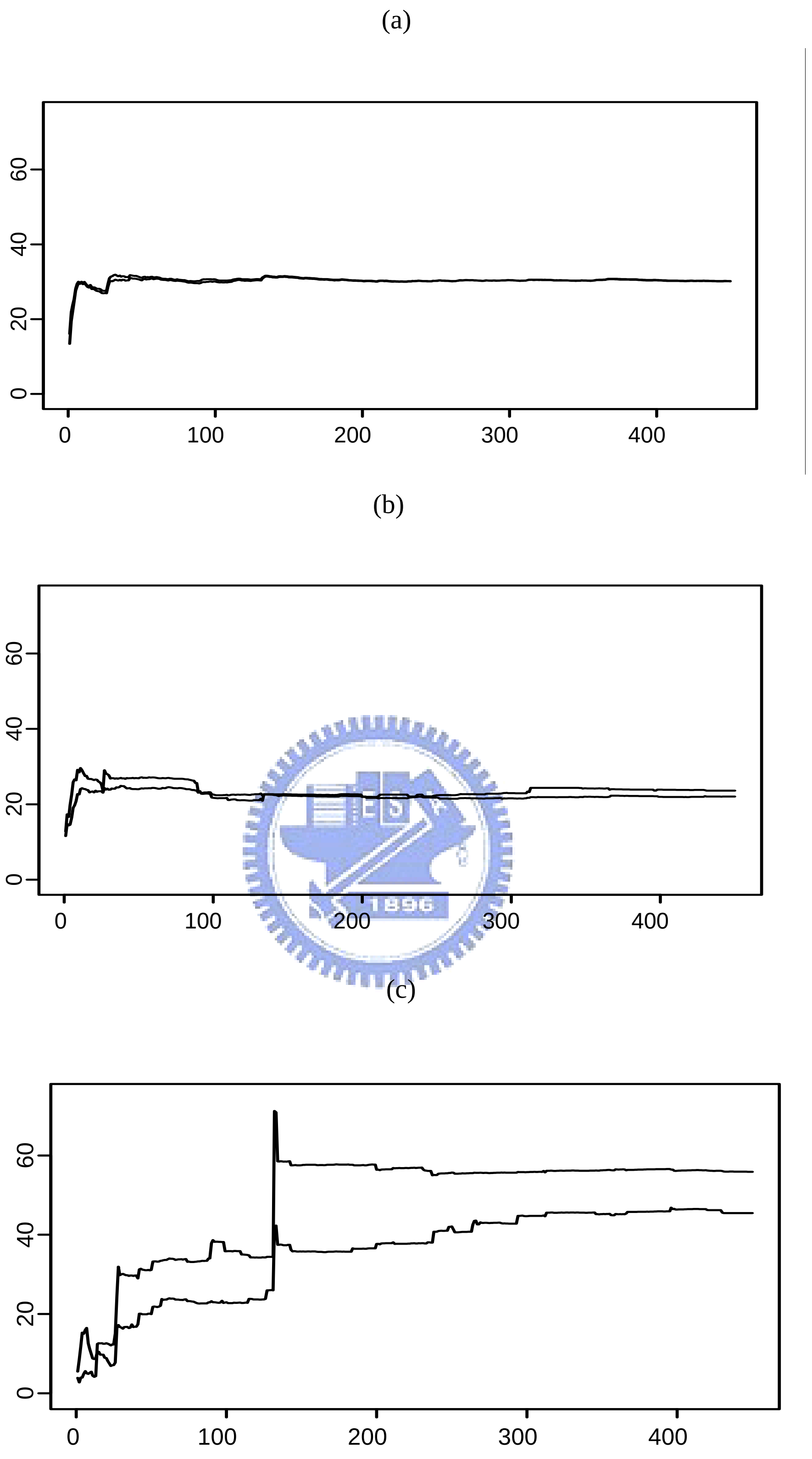


Figure 4.2. Convergence monitoring: (a) $\hat{V}$ *versus* $\hat{W}_c$ *, (b)* $\hat{W}_m$ *versus* $W_m\hat{W}_c$ *,*

(c) $\hat{B}_m$ *versus* $B_m\hat{W}_c$ *(one unit in the x-axis is 100 iterations).*

For this problem, according to the engineering knowledge, the acceptance *tolerance* of difference is set to be 1. We carry out the TCP method by running 5 independent chains with 50,000 iterations, including 5,000 burn-in iterations, and monitor the convergence of RJMCMC samplers by examining $\hat{V}$ v.s. $\overset{\wedge}{W_c}$, $\overset{\wedge}{B_m}$ v.s. $\overset{\wedge}{B_m W_c}$, and $\overset{\wedge}{W_m}$ v.s. $\overset{\wedge}{W_m W_c}$, as described in Chapter 2.5. The convergences of the above three sets of comparisons can be visualized in Figures 4.2 (a)-(c).

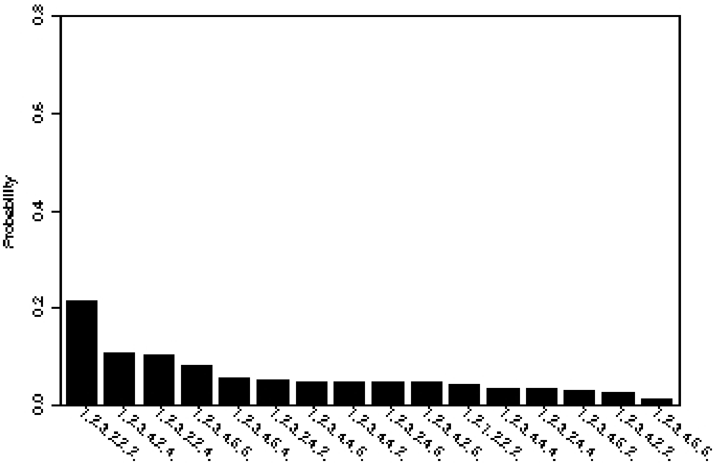


Figure 4.3. Estimated posterior distribution of the partition (the partitions with probability less than 0.005 are not displayed).

Finally, we summarize the results based on the last 10,000 MCMC iterations and the posterior distribution for the tool partition is displayed in Figure 4.3 where the partitions with probability less than 0.005 are not shown. The best partition for the set of tools, {SPU16, SPU14, SPU07, SPU05, SPU04, SPU03}, with the highest posterior probability 0.2244 is {SPU16, SPU14, SPU07, SPU04}, {SPU05}, and {SPU03} (denoted as (111213) in Figure 4.3). This partition result is consistent with that obtained by the CART method [46] with cost-complexity=30 shown in Table 4.2

and Figure 4.4. Although two different approaches reach the same partition result, it is somehow difficult for engineers to understand and interpret the meaning of cost-complexity=30 in CART method. In contrast, the engineering tolerance control is much easy to set and interpret in the TCP method.

Table 4.2. The partitioning results using the CART method with different values of the cost complexity.

Partitioning result	Cost-complexity
(1,2,3,4,5,6)	0
(1,2,2,3,2,4)	5
(1,1,1,2,1,3)	30
(1,1,1,2,1,2)	45

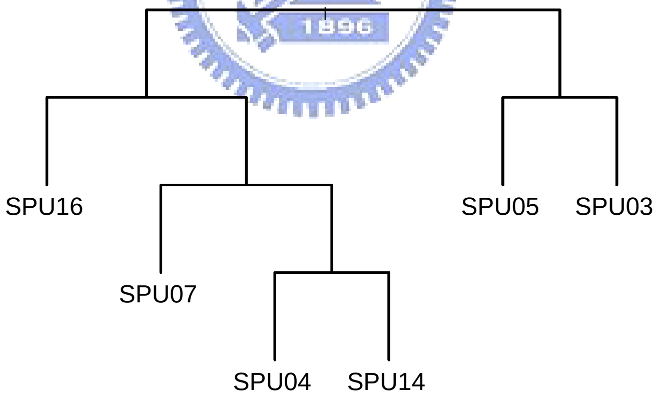


Figure 4.4. Tree obtained by the CART method with cost complexity=30.

After further checking on this problematic step, the engineers find that there are two different tool types: one type includes SPU03 and SPU05 and another type includes SPU16, SPU14, SPU07, and SPU04. Because different tool types use different process chemicals, the contaminated chemical is the main source of bad performance of SPU03 and SPU05. After eliminating the contaminated chemical, the performance of SPU03 and SPU05 becomes regular and the Srow measurements are as same as those for other tools. Accordingly, the overall sample mean for Srow among tools reduces from 5.98 to 5.4 and the sample variance reduces from 1.85 to 1.52 after the adjustment. It really enhances the product yield.

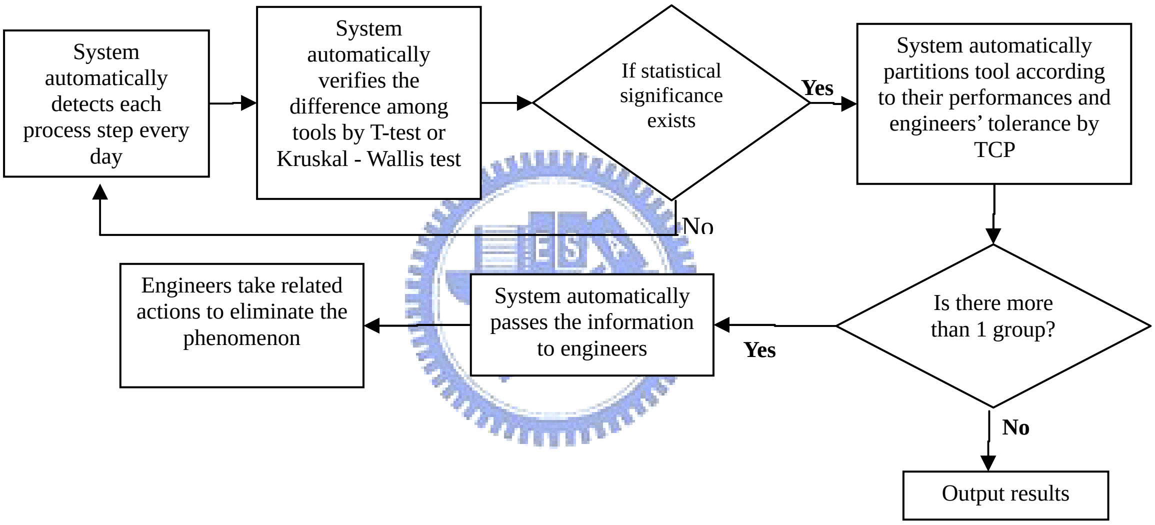

System automatically detects each process step every day
System automatically verifies the difference among tools by T-test or Kruskal - Wallis test
If statistical significance exists
Yes
System automatically partitions tool according to their performances and engineers' tolerance by TCP
No
Is there more than 1 group?
Yes
System automatically passes the information to engineers
Engineers take related actions to eliminate the phenomenon
No
Output results

Figure 4.5. Engineer daily trouble shooting flow by combining statistical tests and TCP method.

From this application, we suggest to integrate the TCP method with statistical tests into a statistical dashboard [4] to form an analysis flow, as shown in Figure 4.5. After building automatic systems according to the analysis flow, systems could execute the analysis automatically at night for each item of each product to compare the tool performances according to the pre-defined tolerances. Then, at the beginning of the daily work, the engineers could quickly detect the possible problematic tools for yield enhancement as demonstrated in Table 4.3. This will dramatically shorten the time for engineers to find out the root causes of yield variance and eliminate the problematic tools. This working flow for yield enhancement not only avoids the subjective engineering judgments in tool comparisons, but also links with a well management plan through an engineering discussion about the reasonable tolerances. Based on the example, we have shown the TCP method can really help engineers enhance yield by automatically partitioning the tools according their performances.

Table 4.3. An illustration example for automatically detecting the performance difference among tools for each process step.

	P value of	TCP Result
Step	T or Kruskal Wallis Test	(Group, Tool List; Mean)
Step 10	**0.000005**	**(1.SPU03; 9.05);(2.SPU05; 7.17); (3.SPU04,SPU07,SPU14,SPU16; 5.58)**
Step 15	**0.0003**	**(1.TEC02; 8.32);(2.TEC01; 8.08);**
Step 2	0.06	(1.ACE01,ACE02; 8.1);
Step 4	0.08	(1.PHO01,PHO0202, PHO03; 8.21);
…	…	…

4.2 Process Capability Indices Enhancement

The process capability indices C_p and C_{pk} [47] have been widely used in the manufacturing industry for measuring the process performance and product quality. These two indices are defined as

$$C_p = \frac{USL - LSL}{6\sigma} \quad , \quad C_{pk} = \min\left\{\frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma}\right\},$$

where LSL and USL are the lower and upper specification limits, respectively which are defined by the process engineers or the product designers, μ is the process mean and σ is the process standard deviation. A larger value of C_p or C_{pk} indicates better product quality and process capability. Conventionally, we use $\overline{X} = (\sum_{i=1}^{n} X_i) \Big/ n$ and $s = [\sum_{i=1}^{n} (X_i - \overline{X})^2 \Big/ (n-1)]^{1/2}$ as the estimators for μ and σ respectively, so the natural estimators of C_p and C_{pk} are $\hat{C}_p = \frac{USL - LSL}{6s}$ and $\hat{C}_{pk} = \min\left\{\frac{USL - \overline{X}}{3s}, \frac{\overline{X} - LSL}{3s}\right\}$. In general, the minimum requirement for $\hat{C}_p$ and $\hat{C}_{pk}$ is 1.33. Some leading companies may set a higher standard, such as $\hat{C}_p$ and $\hat{C}_{pk}$ >=2, to guarantee their competitiveness. Therefore, enhancing $\hat{C}_p$ and $\hat{C}_{pk}$ is one of the major tasks for process engineers in the semiconductor industry.

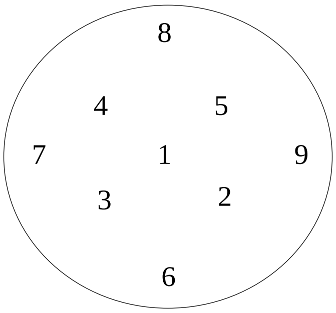
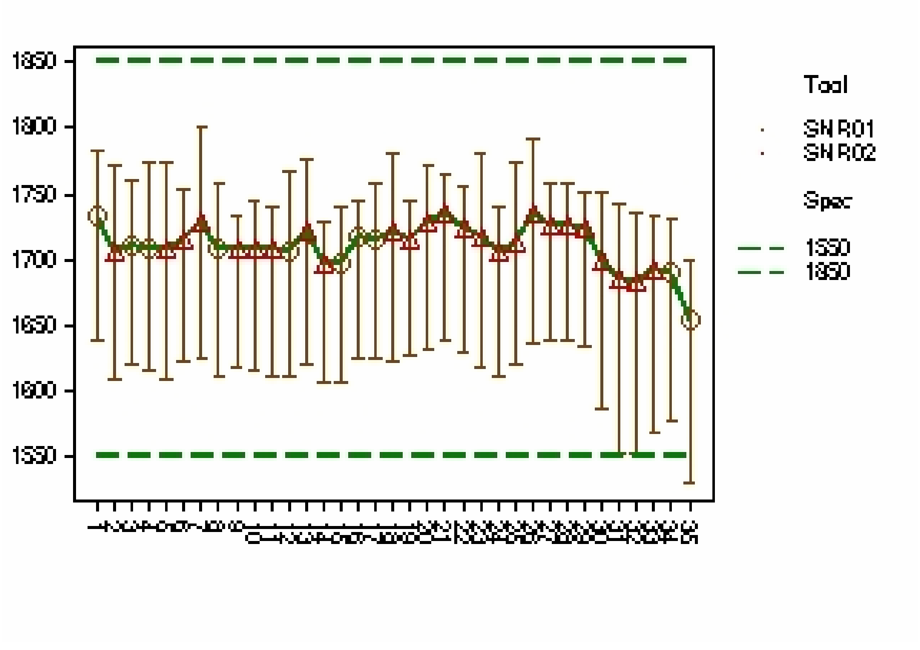


Figure 4.6. Site locations in each wafer.

As introduced in Chapter 1, the semiconductor manufacturing processes by lot batches with 25 wafers per lot. After completing each process step, we sample one or several wafers from 25 wafers to measure the related parameters at 5-9 pre-determined sites on each wafer as shown in Figure 4.6. In this example, the process parameter considered is the critical oxide thickness after one important diffusion process step during the semiconductor manufacturing. The data are measured at 9 sites (shown on Figure 4.6) for each sampled wafer and only one wafer was sampled from each lot from 2006/8/6 to 2006/8/12. In Figure 4.7, we illustrate a particular lot-trend and the histogram of the oxide thickness in which the error bar indicates the minimum and maximum values within each lot. Based on the product specification with USL =1850 and LSL=1550, the C_p and C_{pk} are equal to 1.132, and 1.105, respectively.

(a)



(b)

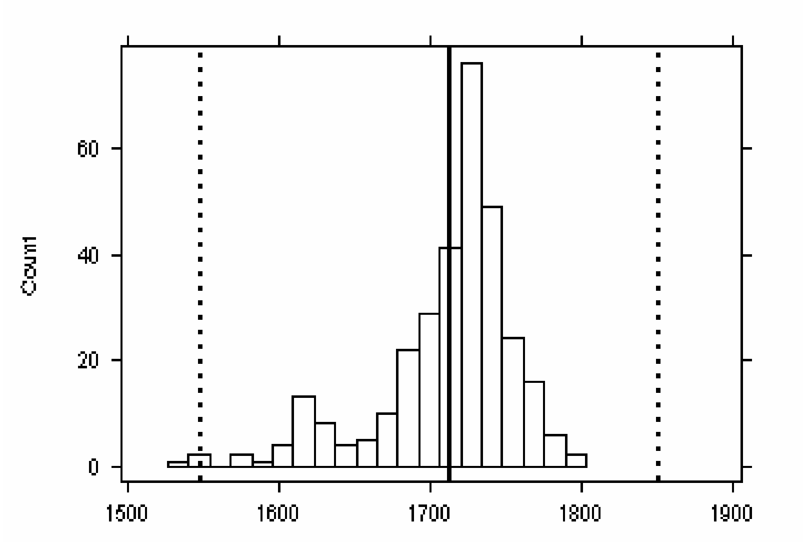


*Figure 4.7. (a) The trend chart (b) The histogram of Oxide thickness from 2006/0806 to 2006/08/12 Note that data count=9*1*35(site*wafer*lot), USL=1850, and LSL=1550.*

Table 4.4. The ANOVA result for tool effect.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
tool	1	4480.967937	4480.967937	2.259479857	0.133806921
Residuals	313	620737.0956	1983.185609		

Table 4.5. The ANOVA result for site effect.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
site	8	409274.2921	51159.28651	72.49452748	0
Residuals	306	215943.7714	705.6985994		

After applying the ANOVA analysis to the oxide thickness data summarized in Table 4.4 and Table 4.5, we found that the tool effect is not significant but the site effect is extremely significant. The site effects are also clearly seen from the box plots by sites in Figure 4.8. Therefore, we apply the TCP method to partition these sites. We first define the *tolerance* to be 10% of the USL-LSL, which is equal to 30. We carry out the TCP method by running 5 independent chains with 200,000 iterations including 75,000 burn-in iterations. The convergence monitoring is plotted in Figure 4.9.

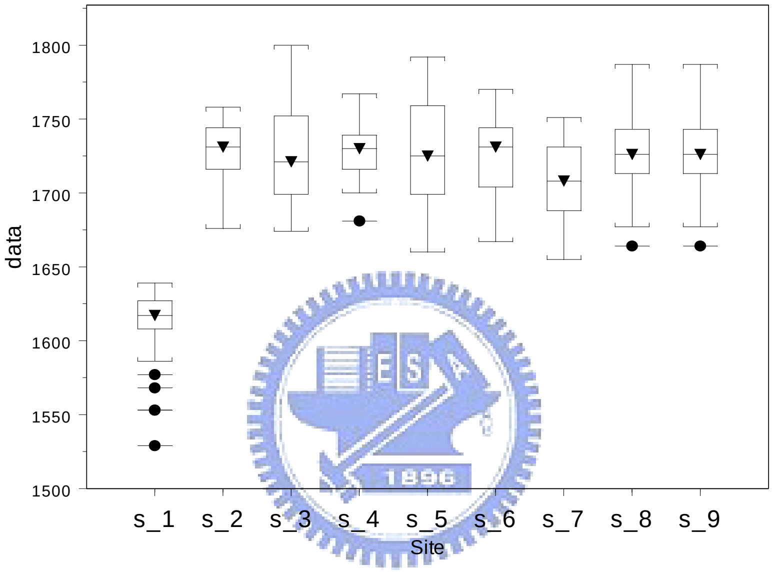


Figure 4.8. Box plots of oxide thickness by different sites.

Finally, the posterior distribution for the partition is calculated based on the last 10,000 MCMC iterations, shown in Figure 4.10. The best partition has two groups {site1}, {site2, site3, site4, site5, site6, site7, site8, site9} (denoted as (122222222) in the figure for simplicity) and the posterior probability 0.60. Again, this result is consistent with the site phenomenon by the box plots in Figure 4.8 and related test results. After fine-tuning the process recipe, the site difference was eliminated and C_p and C_{pk} increased to 1.86 and 1.56, respectively.

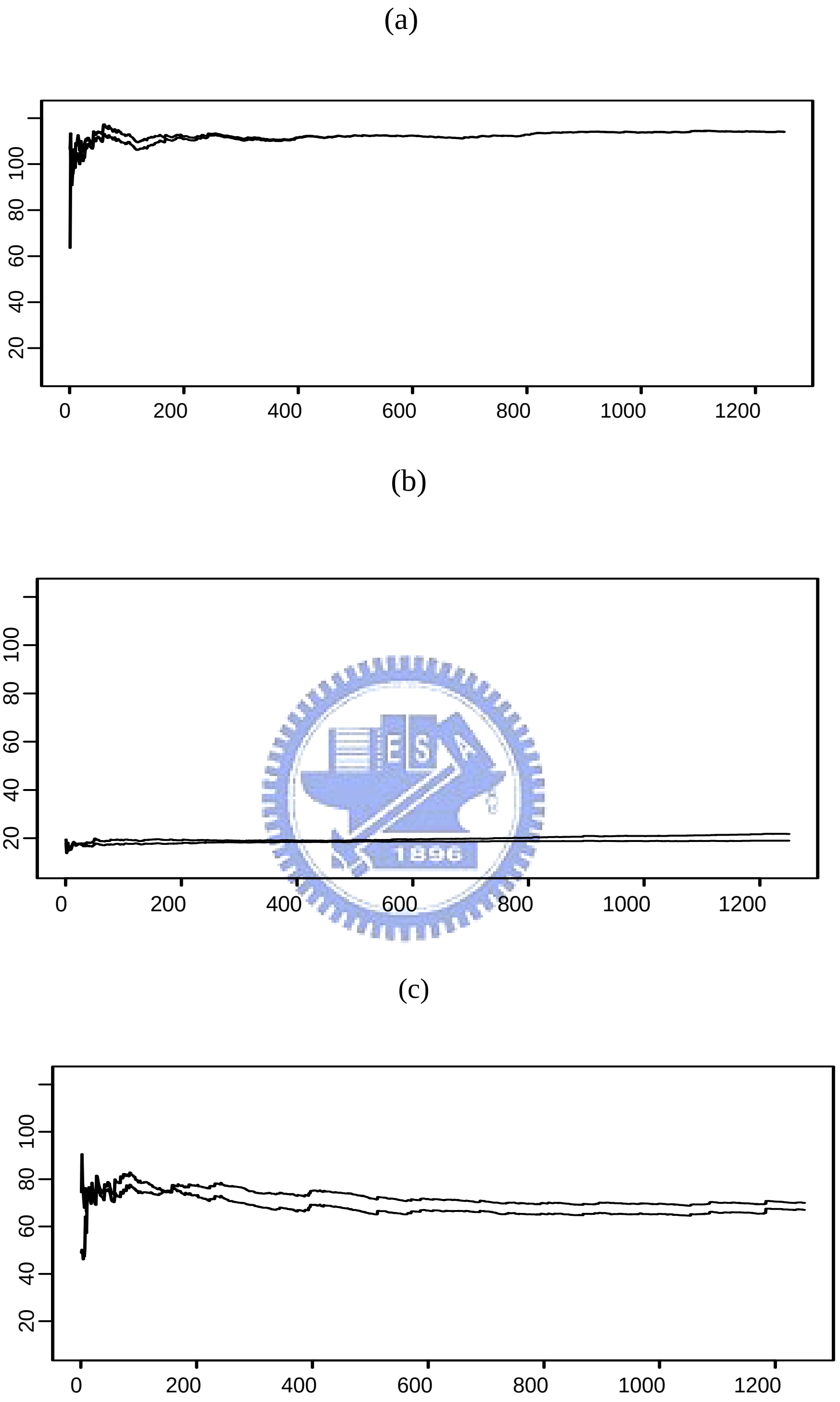


Figure 4.9. Convergence monitoring: (a) $\hat{V}$ *versus* $\hat{W}_c$ *, (b)* $\hat{W}_m$ *versus* $W_m\hat{W}_c$ *, and (c)* $\hat{B}_m$ *versus* $B_m\hat{W}_c$ *(one unit in the x-axis is 100 iterations).*

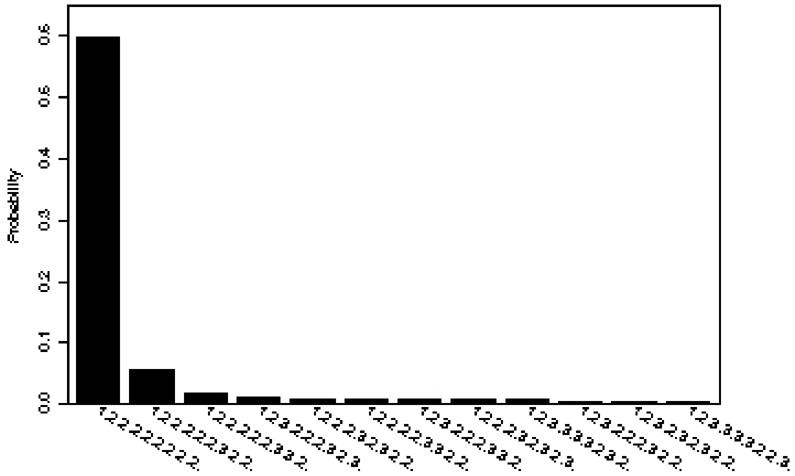


Figure 4.10. The posterior distribution of the site partition (the partitions with probability less than 0.005 are not displayed).

Similarly, for comparison, we perform the CART method with various cost-complexity values. The complete tree based on the CART method and the partition results under different cost-complexity are given in Figure 4.11 and Table 4.6. It turns out that the same partition result can be obtained by the CART method with the cost-complexity=9775. But, again, choosing an appropriate cost-complexity is a harder problem than setting a meaningful tolerance for engineers in practice. Therefore, TCP method is better method for engineers to enhance Cp/Cpk according their engineering tolerance controls.

Furthermore, we suggest that we can integrate the TCP method into the capability enhancement procedure to form an automatic fine-tuning toolbox [27]-[29] as shown in Figure 4.12. This toolbox includes automatically detecting the site differences and automatically adjusting the recipe to eliminate the site differences. It is expected to enhance the capability more efficiently and reduce a lot of workload for engineers. Based on the example, we illustrate TCP method can help engineers enhance Cp/Cpk performances.

Table 4.6. Site partition results based on CART method with various cost complexity values.

Partitioning result	Cost-complexity
(1,2,3,4,5,6,7,8,9)	0
(1,2,3,4,2,2,5,3,3)	20
(1,2,3,2,2,2,4,3,3)	40
(1,2,2,2,2,2,3,2,2)	200
(1,2,2,2,2,2,2,2,2)	9775

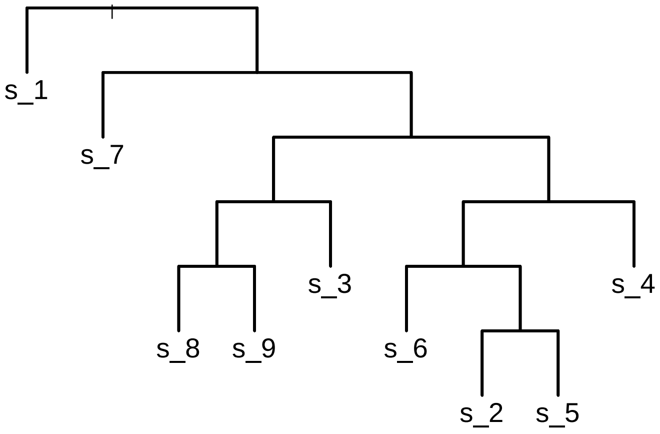


Figure 4.11. Tree structure for the site partition based on CART method.

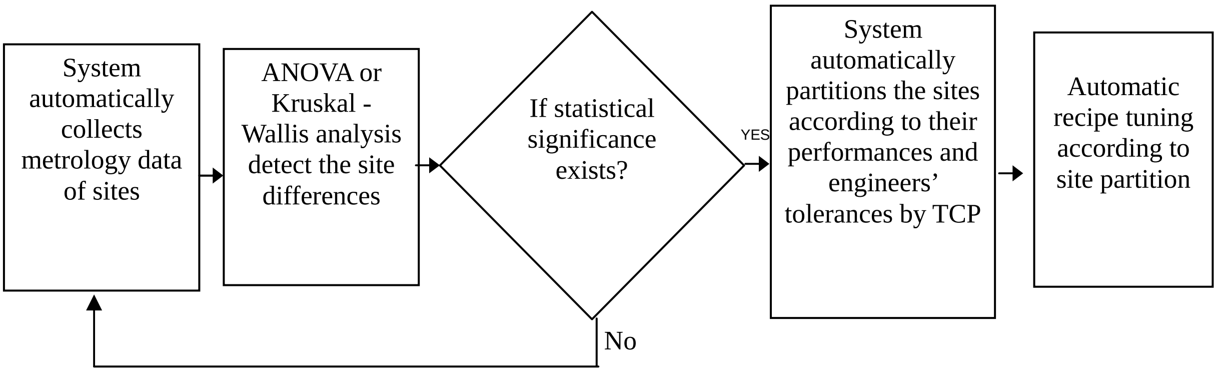

System automatically collects metrology data of sites
ANOVA or Kruskal - Wallis analysis detect the site differences
If statistical significance exists?
YES
No
System automatically partitions the sites according to their performances and engineers' tolerances by TCP
Automatic recipe tuning according to site partition

Figure 4.12. Auto process capability enhancement mechanism by integrating ANOVA, TCP method and the auto-recipe-tuning tools.

5. Conclusion and Discussion

In the semiconductor industry, yield enhancement is one of major challenges to make the companies profitable. Tool comparison is a key task for yield enhancement. After comparing the tool differences, engineers can identify the best groups or problematic groups of tools to enhance product quality or to reject the worst tools, respectively.

From literature, ANOVA, the Kruskal-Wallis [5] test and the CART method [22] are among the most common methodologies used to compare tool differences [6] [7]. At each process step, the tools are compared by these methods. If a statistically significant difference is detected, an alarm is triggered and engineers perform further investigation. This process saves engineers a great deal of time in finding variance among sources and identifying abnormal tools.

However, there are several phenomena by the existing methods to compare tool performances, such as the followings: (1) non-uniform tool usage in most process steps should be taken into account, (2) engineers still need to take time to identify problematic tools after detecting statistical significant differences. It is still very time-consuming, (3) there are many different methods by multiple pairwise comparison procedures [9]-[15], but all these methods could not directly partition different treatments (or tools) into several homogenous groups to allow engineers to quickly understand the overall profile of several treatments (or tools), (4) several cluster approaches [16]-[19] that partition these tools in a balanced design based on the results of likelihood ratio test, Studentized range test, rank test, and simultaneous F-test at level α respectively, however they may get too many partition groups with small differences when the number of observations for each tool is large, and (5) there are many different values of the parameters in the pruning methods to develop different sizes of trees by CART method that are difficult to be related to the tolerance and related criteria used by engineers, and this phenomenon generates the problem of parameter selection to users.

To sum up, there are three major challenges in tool comparisons: (1) to take into account of unbalanced tool usage in manufacturing processes, (2) to further partition these tools into several homogenous groups by related metrology results instead of detecting only the significant differences, and (3) to partition these tools and get a reasonable partition result according to engineers' tolerance controls. However, existing methods can not solve these challenges very well.

We propose a TCP method to overcome these challenges. In Section 3.1, we showed that the TCP method can reduce the influences of unbalanced data by several simulation cases. In Section 3.2, we showed TCP method can partition the tools into several homogenous groups according to engineers' tolerances. In Section 3.3, we showed the robustness of TCP method with non-normal data.

In comparisons with CART methods, the TCP method can automatically partition tools into several homogenous groups with the built-in control of tolerance in engineering. Therefore, it also resolves the difficulty of determining related parameters for the CART methods. Instead of using a hierarchical tree structure as CART does, the posterior distribution of the partition is used to discover the partitioning structure of tools. We also provide a method that includes the set-up of initial values and the integration with the natural concept of tolerance in the engineering to facilitate its practice for engineers.

Using two real applications from the semiconductor industry, we not only show that our method could provide correct information for engineers to enhance yield/process capability, but we also provide an idea to build a practical mechanism by integrating engineers' daily work flow and the capability of the most advanced tools. This idea will make it much easier for engineers to realize the performance differences among tools (or treatments) and enhance the yield and process capability.

In the semiconductor industry, the TCP method could also be applied to all similar cases such as recipes or material comparisons. The TCP method can also be applied in multiple comparison problems [48].

6. Future Works

There are many potential extensions of the TCP method for future work. Tools with unequal variances and other types of distribution models are natural extensions for practical engineering applications. All of the above cases could also be extended to a multivariate situation for engineers to perform simultaneous comparisons on a collection of responses.

We also can investigate the integration of the Bayesian hierarchical model and the random variable for dimension matching in RJMCMC for the Bayesian CART Model. With built-in control of tolerance in engineering, this will resolve the difficulty of pruning and regrouping in the Bayesian CART methods. We could extend the methodology of the TCP method to a regression trees.

Finally, we can develop a new method to automatically judge the convergence of the TCP method such that the TCP method will become a fully automatically partitioning method.

Based on the advantage that the TCP method can automatically partition tools to several homogenous groups according engineers' tolerances, another important extension is to develop an automatic system to integrate the TCP method with the ANOVA and the Kruskal-Wallis test. Therefore, it will be possible to automatically alarm possible excursions in automatic process control (APC) [27]-[29]. Thus, we would provide more improvements for automatic process control by statistical methods in the semiconductor industry.

7. Appendix

Appendix A. Introduction of regression trees

The methodology of classification and regression trees (CART) [22], is a recursive partitioning algorithm to partition data into several homogenous groups. Classification trees and regression trees are applied for categorical response and continuous response, respectively. The following figure, Figure A.1, makes a brief introduction using a graphical representation to construct the trees. In our research, the type of our response is continuous, so we only focus on the introduction of regression trees in the followings below.

Suppose our data consists of p input variables and a response, for each of N observations: that is, $(x_i,\ y_i),\ i=1,\ ...,\ N,$ with $x_i=(x_{i1},\ x_{i2},\ ...,\ x_{ip})$. The algorithm will automatically decide the splitting variables from $(x_{i1},\ x_{i2},\ ...,\ x_{ip})$ and split points. Suppose first that we have a partition into κ regions $R_1,\ R_2,\ ...,\ R_\kappa$. Then our response model is denoted by

$$f(x)=\sum_{k=1}^{\kappa} c_m I(x \in R_k) \qquad \text{(A.1)}$$

where c_m is a constant in each region.

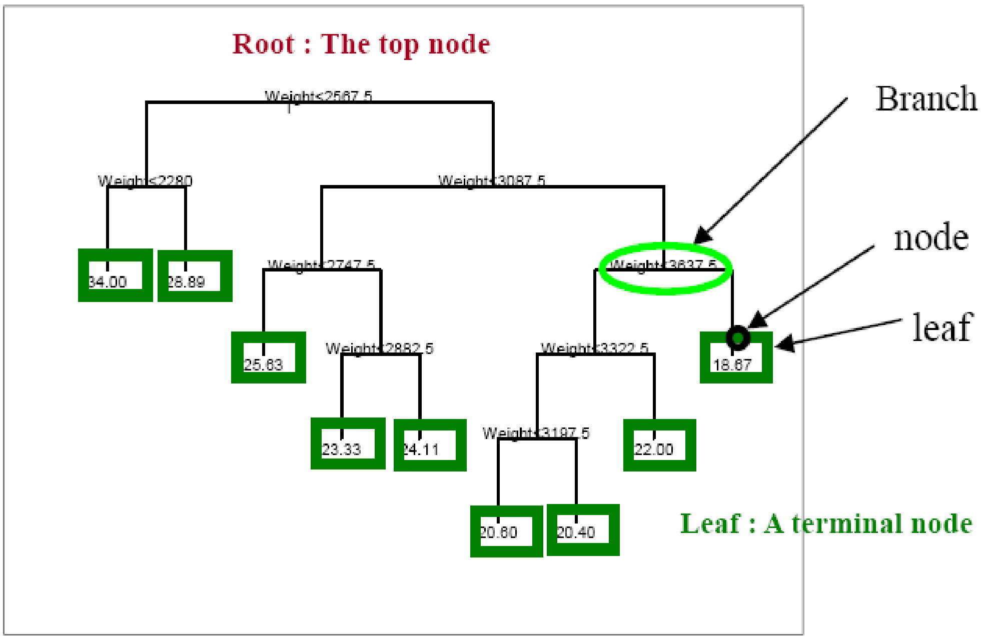


Figure A.1 Construction of a tree.

If we adopt minimization of the sum of squares $\sum(y_i - f(x_i))^2$ as the criterion for split rule, we will use $\hat{c}_m$ to estimate c_m, where $\hat{c}_m$ is the average of y_i in the region R_k.

$$\hat{C}_{\mathrm{m}} = \text{average}(y_i|x_{\,i} \in R_m) \tag{A.2}$$

We will illustrate the regression trees in three sections. We will present how to find the best splitting variable and split point to partition the data at each node in A.1 , how to decide the tree size in A.2, and how to view the tree result in A.3.

A.1. How to find the best splitting variable and split point to partition the data at each node

Start from all of the data and choose a splitting variable X_j. If the split variable X_j is continuous variable, then a split point *s* will define the pair of half-planes

$$R_1(j,s)=\{X|X_j \le s\} \text{ and } R_2(j,s)=\{X|X_j > s\}. \qquad \text{(A.3)}$$

If the split variable X_j is categorical variable, then we find a split set *s* and split data into the pair of half planes

$$R_1(j,s)=\{X|X_j \in s\} \text{ and } R_2(j,s)=\{X|X_j \notin s\}.$$

To seek the best splitting variable *j* and split point (or split set) *s* by solving

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2] \qquad \text{(A.4)}$$

For any choice of *j* and *S*, the solution of c_1 and c_2 are estimated by $\hat{c}_1$ and $\hat{c}_2$, where $\hat{c}_1$ and $\hat{c}_2$ are as follows below.

$$\begin{aligned} \hat{c}_1 &= \text{average}(y_i|x_i \in R_1(j,\ s)) \\ \hat{c}_2 &= \text{average}(y_i|x_i \in R_2(j,\ s)) \end{aligned} \qquad \text{(A.5)}$$

We partition the data into two resulting regions and repeat the splitting process on each of the two regions. Then the process will grow a tree step by step and split the data into several terminal nodes.

A.2. How to decide the tree size

How large should we grow the tree? A large tree might over-fit the data; on the contrary, a small tree might not describe the important structure. Tree size will be a parameter to control the model’s complexity. So how to choose a reasonable tree size is very important in CART algorithm.

Traditionally, there are two ways to prune trees for choosing a tree size. One is pre-pruning, and the other is post-pruning. The algorithm of pre-pruning is setting some criteria to determine how to stop growing the tree from growing, and the algorithm of post-pruning is pruning a tree by some criteria after growing a complete tree. Since the criterion in pre-pruning is difficult in determining the value, we will use post-pruning to choose the tree size in our research.

Cost-complexity pruning is one of the popular pos-pruning methods. Suppose T is a tree getting from the splitting method as in A.1. $|T|$ is the number of terminal nodes in T, then we partition all of the data into $|T|$ regions. We index terminal nodes by k, and we represent the respective region by R_k. Suppose

$$\hat{c}_k = \frac{1}{N_k} \sum_{x_i \in R_k} y_i \tag{A.6}$$

$$Q_k(T) = \frac{1}{N_k} \sum_{x_i \in R_k} (y_i - \hat{c}_k)^2 , \tag{A.7}$$

the cost complexity criterion is represented by

$$C_\alpha(T) = \sum_{k=1}^{|T|} N_k Q_k(T) + \alpha \times |T| \tag{A.8}$$

where

N_k is the number of the observation data falling in the region R_k.

k is the index of terminal nodes on the binary tree T.

$|T|$ is the number of terminal nodes in T, and.

α is the cost-complexity ($\alpha \geq 0$).

For given a cost-complexity α, we can get a subtree T_α of T to minimize $C_\alpha(T)$. From this formula, we can find the larger value α, the smaller size of subtree T_α that we will get. For given each value α, we can get a unique smallest subtree T_α. If $\alpha = 0$, we can get a full tree.

As the meaning of cost-complexity α is difficult to connect with the concept of engineering tolerance control, so engineers are hard to choose a correct cost-complexity α in order to get a reasonable tree size.

A.3. How to view the tree result

The structure of tree is very important information from regression tree algorithm. We can realize the similarity in our data. The data which belong into the same terminal nodes means they have the highest similarity by regression tree algorithm. When the data does not belong to the same terminal nodes, it will split into different terminal nodes later if the data is more similar. So we could realize the similarities of tool performances by the tree structure in our research. We illustrate the result of Case I in section 3.1 as an example of how to read the tree structure.

Table A.1. Partitioning results with respect to different values of cost complexity in the CART model for the simulated data in Case I.

Partitioning result	Cost-complexity
(1,2,3,4,5)	0
(1,1,2,2,3)	1
(1,1,1,1,2)	90

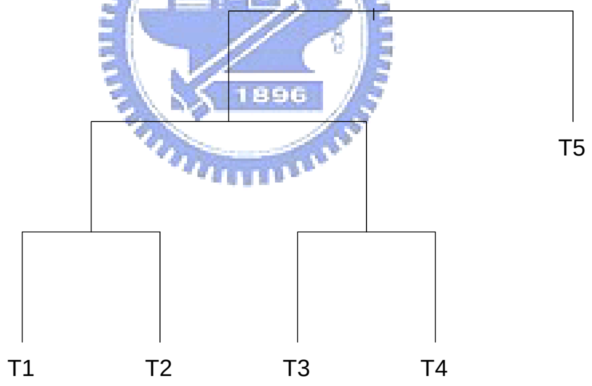


Figure A.2. A tree obtained by the CART model for section 3.1.

From Table A.1, we will get the partition result (12345) when the cost-complexity is 0; that is each tool is partitioned into different group. If we set cost-complexity is 1, we will get the partition result (11223); that is the tools T1 and

T2 belong to one group, T3 and T4 belong to one group, and T5 belong to another one. We also can get the same information from the structure of the tree in Figure A.2. As the similarity of T1 to T2 is higher than that of T1 to T4, the time that T1 and T2 split into different groups is later than that of T1 and T4. As such way, we can understand the similarity among T1, T2, T3,T4, and T5 by the tree structure.

Appendix B. The introduction of Gibbs sampling

Gibbs sampling, also called alternating conditional sampling, is a particular Markov chain algorithm and useful in many multidimensional problem. It is named by Geman and Geman [49], who used it for analyzing Gibbs distributions on lattice. Nevertheless, the works of Gelfand and Smith [50] and Gelfand et al. [51] introduced Gibbs sampling into the mainstream statistics. To date, most statistical applications of MCMC have used Gibbs sampling.

Suppose $P(y\,|\,\theta)$ is the data distribution with *d*-dimensional parameter vector $\theta=(\theta_1,\theta_2,...,\theta_d)$ and $P(\theta)$ is the related prior distribution, then $P(\theta,y)$ is the joint density of θ and y with $P(\theta,y)=P(y\,|\,\theta)\;\;P(\theta)$ and $P(\theta\,|\,y)$ is the posterior density with $P(\theta\,|\,y)=\dfrac{P(\theta,y)}{P(y)}=\dfrac{P(y\,|\,\theta)P(\theta)}{P(y)}\propto P(y\,|\,\theta)P(\theta).$ For Bayesian inference, our target density is the posterior density $P(\theta\,|\,y)$, then we can use Gibbs sampling to construct a Markov chain which will converge to the target density $P(\theta\,|\,y)$.

Suppose $P(\theta_j\,|\,\theta_{-j},y)$ is the conditional distribution given all other component of θ, where θ_{-j} represents all components of θ, except for θ_j. The illustration about how to construct a Markov chain by Gibbs sampling is as follows:

At each iteration *t*, we can choose one of the components of θ_j to update. When we select to update the *j*th component θ_j of θ, $\theta^t{}_j$ is sampled from the conditional distribution $P(\theta_j \mid \theta_{-j}^{t-1}, y)$ where $\theta^t{}_j$ represents the *j*th component of θ at iteration *t* and θ_{-j}^{t-1} represents the all the components of θ, except for θ_j, at their current values of the iteration *t-1*. By repeating such iterations, we construct a Markov chain. If the Markov chain satisfy irreducible and aperiodic properties, then it will converge to our target density $P(\theta \mid y)$.

Therefore, if we can get the conditional distribution $P(\theta_j \mid \theta_{-j}, y)$ for *j*=1, …, *d*. we could construct the Markov chain by Gibbs sampling.

Appendix C. The derivation of conditional distributions in TCP method

In the Appendix C, we present the inductions for the conditional distributions of all parameters in Θ = $\{\theta_1,\ \theta_2,\ldots,\theta_J,\ \sigma^2,\ \mu_1,\ \mu_2\ldots\ \mu_\kappa$, and τ^2 } based on a given partition g . At the followings, we will use the notation of $\Theta_{-\theta_j}$ to indicate the set of all parameters except the parameter of θ_j

C.1 $\theta_j|\Theta_{-\theta_j}$ ~ Normal $(\frac{b_j}{2a_j},\frac{1}{2a_j})$

where $b_j=(\frac{\mu_k}{\tau^2}+\frac{n_j\bar{y}_{.j}}{\sigma^2})$, $a_j=(\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2})$ for $j\in S_k$, and j =1,2 ,…, J , and n_j is the number of observations for tool *j*.

<Proof>

$P(\ g,\ \boldsymbol{\mu},\ \tau^2,\ \boldsymbol{\theta},\ \sigma^2,\ \boldsymbol{y})=$

$$P(g)\left[\prod_{k=1}^{\kappa}P(\mu_k\,|\,\tau^2,g)\right]P(\tau^2)\left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right]P(\sigma^2)$$

where $\boldsymbol{\mu}=(\mu_1,\mu_2,\ldots,\mu_\kappa)^t$, $\boldsymbol{\theta}=(\theta_1,\theta_2,\ldots,\theta_J)^t$,

and the partition is $g=\bigcup_{k=1}^{\kappa}S_k$,

$$= A\left[\prod_{j=1}^{J}\left(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}}\right)\right]\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}\left(\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}}\right)\right]$$

where $A = P(g)\left[\prod_{k=1}^{\kappa}P(\mu_k \mid \tau^2, g)\right]P(\tau^2)P(\sigma^2),$

$$= \quad A^* \cdot e^{\sum_{j=1}^{J}\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}+\sum_j\sum_i\frac{-(y_{ij}^2-2y_{ij}\theta_j+\theta_j^2)}{2\sigma^2}}$$

where $A^* = A\cdot(\frac{1}{\sqrt{2\pi\tau^2}})^J \cdot(\frac{1}{\sqrt{2\pi\sigma^2}})^N$ and $N = n_1 + n_2 + \cdots + n_J$,

$$= A^* \cdot e^{\sum_{j=1}^{J}\frac{-(\theta_j^2-2\theta_j\mu_j^*+\mu_j^{**})}{2\tau^2}+\sum_{j=1}^{J}\frac{-(n_j\theta_j^2-2\theta_j n_j\overline{y}_{.j}+\sum_i y_{ij}^2)}{2\sigma^2}}$$

where $\mu^*{}_j = \sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)}$, and $\mu^{**}{}_j = (\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2$,

$$= A^* \cdot e^{-\sum_{j=1}^{J}\left\{a(\theta_j-\frac{b}{2a})^2+C_j\right\}}$$

where $a = (\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2})$; $b = (\frac{\mu_j^*}{\tau^2}+\frac{n_j\overline{y}_{.j}}{\sigma^2})$ and $C_j = \frac{\sum_i y_{ij}^2}{2\sigma^2}-\frac{b^2}{4a}+\frac{\mu_j^{**}}{2\tau^2}$.

Since $\sum_{j=1}^{J}\frac{-(\theta_j^2-2\theta_j\mu_j^*+\mu_j^{**})}{2\tau^2}+\sum_{j}^{J}\frac{-[n_j\theta_j^2-2\theta_j n_j\overline{y}_{.j}+\sum_i y_{ij}^2]}{2\sigma^2}$,

$$= -\sum_{j=1}^{J}\left\{(\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2})\theta_j^2-(\frac{\mu_j^*}{\tau^2}+\frac{n_j\overline{y}_{.j}}{\sigma^2})\theta_j+\frac{\sum_i y_{ij}^2}{2\sigma^2}+\frac{\mu_j^{**}}{2\tau^2}\right\},$$

$$= -\sum_{j=1}^{J}\left\{a_j(\theta_j-\frac{b_j}{2a_j})^2+\frac{\sum_i y_{ij}^2}{2\sigma^2}-\frac{b_j{}^2}{4a_j}+\frac{\mu_j^{**}}{2\tau^2}\right\}$$

where $a_j = (\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2})$; $b_j = (\frac{\mu_j^*}{\tau^2}+\frac{n_j\overline{y}_{.j}}{\sigma^2})$,

$$= -\sum_{j=1}^{J}\left\{a_j(\theta_j-\frac{b_j}{2a_j})^2+C_j\right\} \qquad \text{where } C_j = \frac{\sum_i y_{ij}^2}{2\sigma^2}-\frac{b_j{}^2}{4a_j}+\frac{\mu_j^{**}}{2\tau^2}.$$

Then

$P(\theta_j \mid \Theta_{-\theta_j})$

$$= P(g,\ \boldsymbol{\mu},\ \tau^{2},\ \boldsymbol{\theta},\ \sigma^{2},\ \boldsymbol{y}) / P(g,\ \boldsymbol{\mu},\ \tau^{2},\ \sigma^{2},\ \boldsymbol{y})$$

$$= A^{*} \cdot e^{-\sum_{j=1}^{J}\left\{a_j(\theta_j-\frac{b_j}{2a_j})^2+C_j\right\}} / \iiint_{\theta_1\ldots\theta_J} A^{*}\cdot e^{-\sum_{j=1}^{J}\left\{a_j(\theta_j-\frac{b_j}{2a_j})^2+C_j\right\}} d\theta_1 \ldots d\theta_J$$

$$= \prod_{j=1}^{J} (\frac{\sqrt{2a_j}}{\sqrt{2\pi}}) \cdot e^{-a_j(\theta_j-\frac{b_j}{2a})^2}$$

$$= \prod_{j=1}^{J} \text{Normal}(\frac{b_j}{2a_j}, \frac{1}{2a_j}), \quad \text{where } a_j = (\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2});\ b_j = (\frac{\mu_j^{*}}{\tau^2}+\frac{n_j \bar{y}_{\cdot j}}{\sigma^2})$$

$$= \prod_{j=1}^{J} \text{Normal}(\frac{b_j}{2a_j}, \frac{1}{2a_j}),$$

$$\text{where } a_j = (\frac{1}{2\tau^2}+\frac{n_j}{2\sigma^2});\ b_j = (\frac{\mu_k}{\tau^2}+\frac{n_j \bar{y}_{\cdot j}}{\sigma^2}) \quad \text{when} \quad j \in S_k$$

C.2 $\mu_k \mid \Theta_{-\mu_k} \sim \text{Normal}\,(\frac{\sum_{j\in S_k}\theta_j}{w_k}, \frac{\tau^2}{w_k})\,\mathrm{I}_{\mu_k}(a,b)$ where $w_k = \{\#\text{ of } j \text{ in } S_k\}$.

<Proof>

$$P(g,\ \boldsymbol{\mu},\ \tau^{2},\ \boldsymbol{\theta},\ \sigma^{2},\ \boldsymbol{y}) =$$

$$P(g)\left[\prod_{k=1}^{\kappa} P(\mu_k \mid \tau^2, g)\right] P(\tau^2) \left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}} e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right] P(\sigma^2)$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_\kappa)^t$, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_J)^t$,

and the partition is $g = \bigcup_{k=1}^{\kappa} S_k$,

$$= B \cdot \left[\prod_{k=1}^{\kappa} P(\mu_k \mid \tau^2, g)\right]\left[\prod_{j=1}^{J} e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}}\right]$$

$$\text{where } B = P(g)P(\tau^2)(\frac{1}{\sqrt{2\pi\tau^2}})^J \left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right] P(\sigma^2),$$

$$= B \cdot \left[\prod_{k=1}^{\kappa} P(\mu_k \mid \tau^2, g) \right] \left[\prod_{k=1}^{\kappa} \prod_{j \in S_k} e^{\frac{-(\theta_j - \mu_k)^2}{2\tau^2}} \right],$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) \prod_{j \in S_k} e^{\frac{-(\theta_j - \mu_k)^2}{2\tau^2}} \right],$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) \prod_{j \in S_k} e^{\frac{-(\theta_j^2 - 2\theta_j \mu_k + \mu_k^2)}{2\tau^2}} \right],$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) e^{\frac{-1}{2\tau^2}\left[w_k \mu_k^2 - 2\mu_k (\sum_{j \in S_k} \theta_j) + \sum_{j \in S_k} \theta_j^2 \right]} \right] \quad \text{where } w_k = \{\# \text{ of } j \text{ in } S_k\},$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) e^{\frac{-1}{2\tau^2}\left[w_k \mu_k^2 - 2\mu_k \theta^* + \theta^{**} \right]} \right] \quad \text{where } \theta^* = \sum_{j \in S_k} \theta_j, \text{ and } \theta^{**} = \sum_{j \in S_k} \theta_j^2,$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) e^{\frac{-1}{2\tau^2}\left[w_k (\mu_k - \frac{\theta^*}{w_k})^2 - w_k (\frac{\theta^*}{w_k})^2 + \theta^{**} \right]} \right],$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[P(\mu_k \mid \tau^2, g) e^{\frac{-1}{2\tau^2}\left[w_k (\mu_k - \frac{\theta^*}{w_k})^2 \right] + D_k} \right] \quad \text{where } D_k = \frac{-1}{2\tau^2}\left[-w_k (\frac{\theta^*}{w_k})^2 + \theta^{**} \right],$$

$$= B \cdot \frac{1}{(b-a)} \cdot \prod_{k=1}^{\kappa} \left[\mathrm{I}_{\mu_k}(a,b) \cdot e^{\frac{-w_k}{2\tau^2}(\mu_k - \frac{\theta^*}{w_k})^2 + D_k} \right],$$

Then

$$P(\mu_k \mid \Theta_{-\mu_k})$$

$$= P(g,\ \boldsymbol{\mu},\ \tau^2,\ \boldsymbol{\theta},\ \sigma^2,\ \boldsymbol{y}) / P(g,\ \tau^2,\ \boldsymbol{\theta},\ \sigma^2,\ \boldsymbol{y}),$$

$$= B \cdot \prod_{k=1}^{\kappa} \left[\mathrm{I}_{\mu_k}(a,b) \cdot e^{\frac{-w_k}{2\tau^2}(\mu_k - \frac{\theta^*}{w_k})^2 + D_k} \right] / \iiint_{\mu_1 \ldots \mu_\kappa} B \cdot \prod_{k=1}^{\kappa} \left[\mathrm{I}_{\mu_k}(a,b) \cdot e^{\frac{-w_k}{2\tau^2}(\mu_k - \frac{\theta^*}{w_k})^2 + D_k} \right] d\mu_1 \ldots d\mu_\kappa \quad ,$$

$$= \prod_{k=1}^{\kappa} \left[\mathrm{I}_{\mu_k}(a,b) \cdot e^{\frac{-w_k}{2\tau^2}(\mu_k - \frac{\theta^*}{w_k})^2} \right],$$

$$=\prod_{k=1}^{\kappa}\left[\mathrm{I}_{\mu_k}(a,b)\cdot\mathrm{Normal}(\frac{\sum\limits_{j\in S_k}\theta_j}{w_k},\frac{\tau^2}{w_k})\right],$$

$$=\prod_{k=1}^{\kappa}\left[\mathrm{I}_{\mu_k}(a,b)\cdot\mathrm{Normal}(\frac{\theta^*}{w_k},\frac{\tau^2}{w_k})\right] \quad \text{where} \quad w_k=\{\#\text{ of } j \text{ in } S_k\}.$$

C.3 $\tau^2\ |\Theta_{-\tau^2}$ ~Inverse Gamma $(\frac{\nu+J}{2},\frac{\nu s^2+F}{2})$

where $F=\sum\limits_{j=1}^{J}(\theta_j-\sum\limits_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2$,

and the partition is $g=\bigcup\limits_{k=1}^{\kappa}S_k$.

<Proof>

$P(g,\ \mu,\ \tau^2,\ \theta,\ \sigma^2,\ y)=$

$$P(g)\left[\prod_{k=1}^{\kappa}P(\mu_k\,|\,\tau^2,g)\right]P(\tau^2)\left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum\limits_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right]P(\sigma^2)$$

where $\mu=(\mu_1,\mu_2,...,\mu_\kappa)^t$, $\theta=(\theta_1,\theta_2,...,\theta_J)^t$,

and the partition is $g=\bigcup\limits_{k=1}^{\kappa}S_k$,

$$=C\cdot P(\tau^2)\left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum\limits_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]$$

where $C=P(g)\left[\prod\limits_{k=1}^{\kappa}P(\mu_k\,|\,\tau^2,g)\right]\left[\prod\limits_{j=1}^{J}\prod\limits_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right]P(\sigma^2)$,

$$= C \cdot P(\tau^2)(\frac{1}{\sqrt{2\pi\tau^2}})^J e^{\sum_{j=1}^{J}\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}}$$

$$= C \cdot P(\tau^2)(\frac{1}{\sqrt{2\pi\tau^2}})^J e^{\frac{-F}{2\tau^2}} \quad \text{where } F = \sum_{j=1}^{J}(\theta_j - \sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2 ,$$

$$= C \cdot \frac{(\frac{\nu}{2})^{\nu/2}}{\Gamma(\frac{\nu}{2})} \cdot s^{\nu} \cdot (\tau^2)^{-(\frac{\nu}{2}+1)} \cdot e^{\frac{-\nu s^2}{2\tau^2}} (\frac{1}{\sqrt{2\pi\tau^2}})^J e^{\frac{-F}{2\tau^2}} \quad \text{as } \tau^2 \sim \text{Scaled inverse } \chi^2(\nu, s^2) ,$$

$$= C^* \cdot (\tau^2)^{-(\frac{\nu+J}{2}+1)} \cdot e^{\frac{-(\nu s^2+F)}{2\tau^2}} \quad \text{where } C^* = C \cdot \frac{(\frac{\nu}{2})^{\nu/2}}{\Gamma(\frac{\nu}{2})} \cdot s^{\nu} \cdot (\frac{1}{\sqrt{2\pi}})^J \ ,$$

Then

$P(\tau^2 \,|\, \Theta_{-\tau^2})$

$= P(g,\ \mu,\ \tau^2,\ \theta,\ \sigma^2,\ \mathbf{y}) / P(g,\ \mu,\ \theta,\ \sigma^2,\ \mathbf{y}),$

$$= C^* \cdot (\tau^2)^{-(\frac{\nu+J}{2}+1)} \cdot e^{\frac{-(\nu s^2+F)}{2\tau^2}} \Big/ \int_{\tau^2} C^* \cdot (\tau^2)^{-(\frac{\nu+J}{2}+1)} \cdot e^{\frac{-(\nu s^2+F)}{2\tau^2}} \, d\tau^2$$

$$= \text{Scaled inverse } \chi^2(\nu + J, \frac{\nu s^2 + F}{\nu + J}) \ ,$$

$$= \text{Inverse Gamma } (\frac{\nu + J}{2}, \frac{\nu s^2 + F}{2}\ ) \quad \text{where } F = \sum_{j=1}^{J}(\theta_j - \sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2 .$$

C.4 $\sigma^2 \,|\, \Theta_{-\sigma^2}$ ~Inverse Gamma $(\frac{\nu_1 + N}{2}, \frac{\nu_1 s_1^2 + E}{2})$

where $E = \sum_{j=1}^{J} \sum_{i=1}^{n_j} (y_{ij} - \theta_j)^2$ and $N = \sum_{j=1}^{J} n_j$.

<Proof>

$P(g,\ \mu,\ \tau^2,\ \theta,\ \sigma^2,\ \mathbf{y}) =$

$$P(g)\left[\prod_{k=1}^{\kappa}P(\mu_k\,|\,\tau^2,g)\right]P(\tau^2)\left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right]P(\sigma^2)$$

where $\mu=(\mu_1,\mu_2,...,\mu_\kappa)^t$, $\theta=(\theta_1,\theta_2,...,\theta_J)^t$,

and the partition is $g=\bigcup_{k=1}^{\kappa}S_k$,

$$=D\cdot\left[\prod_{j=1}^{J}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{\sigma^2}}e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})\right]P(\sigma^2)$$

where $D=(\frac{1}{\sqrt{2\pi}})^N P(g)\left[\prod_{k=1}^{\kappa}P(\mu_k\,|\,\tau^2,g)\right]P(\tau^2)\left[\prod_{j=1}^{J}(\frac{1}{\sqrt{2\pi\tau^2}}e^{\frac{-(\theta_j-\sum_{k=1}^{\kappa}\mu_k I_{(j\in S_k)})^2}{2\tau^2}})\right]$

and $N=n_1+n_2+\cdots+n_J$,

$$=D\cdot P(\sigma^2)\cdot\left[(\sigma^2)^{\frac{-N}{2}}e^{\sum_{j=1}^{J}\sum_{i=1}^{n_j}\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}}\right],$$

$$=D\cdot P(\sigma^2)\cdot\left[(\sigma^2)^{\frac{-N}{2}}\right]\cdot e^{\frac{-E}{2\sigma^2}}\quad\text{where } E=\sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij}-\theta_j)^2\,,$$

$$=D\cdot\frac{(\frac{\nu_1}{2})^{\nu_1/2}}{\Gamma(\frac{\nu_1}{2})}\cdot s_1^{\nu_1}\cdot(\sigma^2)^{-(\frac{\nu_1}{2}+1)}\cdot e^{\frac{-\nu_1 s_1^2}{2\sigma^2}}\cdot\left[(\sigma^2)^{\frac{-N}{2}}\right]\cdot e^{\frac{-E}{2\sigma^2}}$$

as $\sigma^2\sim$ Scaled inverse $\chi^2(\nu_1,s_1^2)$,

$$=D\cdot\frac{(\frac{\nu_1}{2})^{\nu_1/2}}{\Gamma(\frac{\nu_1}{2})}\cdot s_1^{\nu_1}\cdot(\sigma^2)^{-(\frac{\nu_1+N}{2}+1)}\cdot e^{\frac{-(\nu_1 s_1^2+E)}{2\sigma^2}}\quad,$$

$$=D^*\cdot(\sigma^2)^{-(\frac{\nu_1+N}{2}+1)}\cdot e^{\frac{-(\nu_1 s_1^2+E)}{2\sigma^2}}\quad\text{where } D^*=D\cdot\frac{(\frac{\nu_1}{2})^{\nu_1/2}}{\Gamma(\frac{\nu_1}{2})}\cdot s_1^{\nu_1}\,.$$

Then

$$P(\sigma^2 \mid \Theta_{-\sigma^2})$$

$$= P(\, g,\ \mu,\ \tau^{\,2},\ \theta,\ \sigma^2,\ \boldsymbol{y})/\, P(\, g,\ \mu,\ \tau^{\,2}, \theta,\ \boldsymbol{y}),$$

$$= D^* \cdot (\sigma^2)^{-(\frac{\nu_1+N}{2}+1)} \cdot e^{\frac{-(\nu_1 s_1{}^2+E)}{2\sigma^2}} \ \Big/ \int_{\sigma^2} D^* \cdot (\sigma^2)^{-(\frac{\nu_1+N}{2}+1)} \cdot e^{\frac{-(\nu_1 s_1{}^2+E)}{2\sigma^2}} d\sigma^2 \quad ,$$

$$= \text{Scaled inverse } \chi^2(\nu_1 + N, \frac{\nu_1 s_1{}^2 + E}{\nu_1 + N}) \quad ,$$

$$=\text{Inverse Gamma } (\frac{\nu_1 + N}{2}, \frac{\nu_1 s_1{}^2 + E}{2}\ )\ \text{ where } E = \sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij} - \theta_j)^2\ .$$

Appendix D. The derivation of acceptance probability in TCP method

Following the introduction of RJMCMC in Section 2.1, the acceptance probability *R* of jumping from current Model M_k (that is, (k, θ_k)) to new model $M_{k'}$ (that is, $(k^{'}, \theta_{k'})$) is the minimum of {1 , *A*}. The detailed formula of *A* is as follows below:

$$A = \frac{p(y \mid \theta_{k'}, k')\, p(\theta_{k'})\, p(k')}{p(y \mid \theta_k, k)\, p(\theta_k)\, p(k)} \frac{J(M_{k'} \to M_k)\, q(v' \mid \theta_{k'}, k', k)}{J(M_k \to M_{k'})\, q(v \mid \theta_k, k, k')} \left| \frac{\delta g_{k,k'}(\theta_k, v)}{\delta(\theta_k, v)} \right| \quad \text{(A.1)}$$

where $J(M_k \to M_{k'})$ is proposal jump probability for a jump from current Model M_k to model $M_{k'}$,

$q(v \mid \theta_k, k, k')$ is a proposal density for dimensional matching,

$g_{k,k'}(\cdot)$ is a bijection function between (θ_k, v) and $(\theta_{k'}, v')$

with $(\theta_{k'}, v') = g_{k,k'}(\theta_k, v)$.

Hence, A = *LR * PJR*PBD*J,*

where *LR* = $\frac{p(y \mid \theta_{k'}, k')\, p(\theta_{k'})\, p(k')}{p(y \mid \theta_k, k)\, p(\theta_k)\, p(k)}$ is the likelihood ratio of two Models,

PJR= $\frac{J(M_{k'} \to M_k)}{J(M_k \to M_{k'})}$ is the proposal jump probability ratio,

PBR= $\frac{q(v' \mid \theta_{k'}, k', k)}{q(v \mid \theta_k, k, k')}$ is the proposal probability ratio,

and *J*= $\left| \frac{\delta g_{k,k'}(\theta_k, v)}{\delta(\theta_k, v)} \right|$ is the Jacobin of bijection function $g_{k,k'}(\theta_k, v)$.

At first, we consider the birth move type: the current partition $g^{(1)}$ with degree $\kappa^{(1)}$ jump to the new partition $g^{(2)}$ with degree $\kappa^{(2)} = (\kappa^{(1)}+1)$ by choosing a group which included at least two tools from $g^{(1)}$ to split randomly. Since the length of $\overline{\boldsymbol{\mu}^{(1)}} = (\mu_1,\ \mu_2,\ \ldots \mu_{\kappa^{(1)}})$ also increases by one, we add a new random variable z that is independently distributed as Normal (μ_z, σ_z^2) for dimension matching. Suppose that we choose the group S_k from $g^{(1)}$ to split into two new groups S_{k_1} and S_{k_2}. Let μ_k be the current value and μ_{k_1}, μ_{k_2} be the new values for the two groups S_{k_1} and S_{k_2}. Then we set

$$\mu_{k_1} = \mu_k + \frac{w_{k_2}}{w_k^{\,2}} z\,, \qquad \mu_{k_2} = \mu_k - \frac{w_{k_1}}{w_k^{\,2}} z\,, \tag{3}$$

where w_i ={# of j in S_i} for $i= k$, k_1, and k_2

with $w_k = w_{k_1} + w_{k_2}$, $S_{k_1} \cup S_{k_2} = S_k$, and $S_{k_1} \cap S_{k_2} = \phi$.

So we have $\overline{\boldsymbol{\mu}^{(2)}} = \{\overline{\boldsymbol{\mu}^{(1)}}_{-\mu_k}\} \cup \{\mu_{k_1}, \mu_{k_2}\}$ and $g^{(2)} = \{S^{(1)}{}_{-S_k}\} \cup \{S_{k_1}, S_{k_2}\}$, and then we replace these densities in (A.1) according to our data distribution, prior distributions, proposal jump probability and bijection function in TCP, we can have

$$LR= \frac{P(g^{(2)})\prod_{k=1}^{\kappa^{(2)}} P(\mu_k^{(2)} \mid \tau^2, g)\prod_{k=1}^{\kappa^{(2)}}\prod_{j\in S_{(k)}^{(2)}} (\frac{1}{\sqrt{2\pi\tau^2}} e^{\frac{-(\theta_j-\mu_k^{(2)})^2}{2\tau^2}})\prod_{j=1}^{\mathbf{J}}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})}{P(g^{(1)})\prod_{k=1}^{\kappa^{(1)}} P(\mu_k^{(1)} \mid \tau^2, g)\prod_{k=1}^{\kappa^{(1)}}\prod_{j\in S_{(k)}^{(1)}} (\frac{1}{\sqrt{2\pi\tau^2}} e^{\frac{-(\theta_j-\mu_k^{(1)})^2}{2\tau^2}})\prod_{j=1}^{\mathbf{J}}\prod_{i=1}^{n_j}(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_{ij}-\theta_j)^2}{2\sigma^2}})},$$

$$\text{where} \quad P(g^{(i)}) \propto \frac{(\kappa^{(i)})^{-1}}{\{\#\ \text{of partition whose degree} = \kappa^{(i)}\}}$$

$$= \frac{1}{\kappa^{(i)} N_{\kappa^{(i)}}}$$

and $N_{\kappa^{(i)}} = \#\{g':\ \text{the degree of partition}\ g' = \kappa^{(i)}\}$

$$= \left[(\kappa^{(i)})^{J} + \sum_{i=1}^{\kappa^{(i)}-1} (-1)^{i} C_i^{\kappa^{(i)}} (\kappa^{(i)} - i)^{J} \right] / (\kappa^{(i)}\ !) \quad \text{for } i\text{=1, 2.}$$

$$= \frac{P(g^{(2)})}{P(g^{(1)})} \left(\frac{1}{b-a} \right) \frac{\prod_{k=1}^{\kappa^{(2)}} \mathrm{I}_{\mu_k^{(2)}}(a,b)}{\prod_{k=1}^{\kappa^{(1)}} \mathrm{I}_{\mu_k^{(1)}}(a,b)} e^{\sum_{j \in S_{k_1}} \left(\frac{(\theta_j - \mu_k)^2}{2\tau^2} - \frac{(\theta_j - \mu_{k_1})^2}{2\tau^2} \right) + \sum_{j \in S_{k_2}} \left(\frac{(\theta_j - \mu_k)^2}{2\tau^2} - \frac{(\theta_j - \mu_{k_2})^2}{2\tau^2} \right)},$$

$$= \frac{P(g^{(2)})}{P(g^{(1)})} \cdot \frac{1}{(b-a)} \cdot \frac{\prod_{k=1}^{\kappa^{(2)}} \mathrm{I}_{\mu_k^{(2)}}(a,b)}{\prod_{k=1}^{\kappa^{(1)}} \mathrm{I}_{\mu_k^{(1)}}(a,b)} \cdot e^{\frac{-1}{2\tau^2} \{ \sum_{j=1}^{J} (\theta_j - \sum_{k=1}^{\kappa^{(2)}} \mu_k^{(2)} I_{(j \in S_k^{(2)})})^2 - \sum_{j=1}^{J} (\theta_j - \sum_{k=1}^{\kappa^{(1)}} \mu_k^{(1)} I_{(j \in S_k^{(1)})})^2 \}},$$

where $\overline{\boldsymbol{\mu}^{(1)}} = (\mu_1,\ \mu_2 \ldots \mu_{\kappa^{(1)}})$, and $\overline{\boldsymbol{\mu}^{(2)}} = \{ \overline{\boldsymbol{\mu}^{(1)}}_{-\mu_k} \} \cup \{ \mu_{k_1}, \mu_{k_2} \}$,

and the partition $g^{(i)} = \bigcup_{k=1}^{\kappa^{(i)}} S_k^{(i)}$.

PJR= $\dfrac{P_{death}}{P_{birth}}$. where P_{death}, and P_{birth} are the proposal probability for the death move type and birth move type respectively.

PBR= $\left\{ \#\ \text{of}\ S_k\ \text{whose}\ w_k\ \geq 2 \right\} \dfrac{2}{\kappa^{(1)}(\kappa^{(1)}+1)} \left\{ 2^{w_k - 1} - 1 \right\} \dfrac{1}{f(z)}$.

$$J = \begin{vmatrix} \dfrac{\partial \mu_{k_1}}{\partial \mu_k} & \dfrac{\partial \mu_{k_1}}{\partial z} \\ \dfrac{\partial \mu_{k_2}}{\partial \mu_k} & \dfrac{\partial \mu_{k_2}}{\partial z} \end{vmatrix} = \begin{vmatrix} 1 & \dfrac{w_{k_2}}{{w_k}^2} \\ 1 & -\dfrac{w_{k_1}}{{w_k}^2} \end{vmatrix} = \frac{w_{k_2}}{{w_k}^2} + \frac{w_{k_1}}{{w_k}^2} = \frac{1}{w_k},$$

$$\mu_{k_1} = \mu_k + \frac{w_{k_2}}{{w_k}^2} z, \quad \text{and} \quad \mu_{k_2} = \mu_k - \frac{w_{k_1}}{{w_k}^2} z .$$

As *A*= *LR* * *PJR***PBD***J*, we substitute *LR*, *PJR*, *PBD*, and *J* into *A*

Therefore,

$$A = \frac{P(g^{(2)})}{P(g^{(1)})} e^{\frac{-1}{2\tau^2}\{\sum_{j=1}^{J}(\theta_j - \sum_{k=1}^{\kappa^{(2)}} {\mu_k}^{(2)} I_{(j \in S_k{}^{(2)})})^2 - \sum_{j=1}^{J}(\theta_j - \sum_{k=1}^{\kappa^{(1)}} {\mu_k}^{(1)} I_{(j \in S_k{}^{(1)})})^2\}} \times \frac{1}{(b-a)} \times \frac{\prod_{k=1}^{\kappa^{(2)}} \mathrm{I}_{\mu_k^{(2)}}(a,b)}{\prod_{k=1}^{\kappa^{(1)}} \mathrm{I}_{\mu_k^{(1)}}(a,b)} \frac{P_{death}}{P_{birth}}$$

$$\times \{\ \#\ \text{of}\ \ k\ \text{whose}\ w_k \geq 2,\ k{=}1,2,\ldots,\kappa^{(1)}\ \} \times \frac{1}{\kappa^{(1)}(\kappa^{(1)}+1)} \{2^{w_k} - 2\} \frac{1}{w_k} \frac{1}{f(z)},$$

where $\kappa^{(i)}$ is the degree of $g^{(i)}$, for *i*=1,2, with $\kappa^{(2)} = \kappa^{(1)} + 1$,

the partition is $g^{(i)} = \bigcup_{k=1}^{\kappa^{(i)}} S_k{}^{(i)}$,

$w_k = \{\#\ \text{of}\ j\ \text{in}\ S_k{}^{(1)}\}$,

$\dfrac{1}{w_k}$ is the Jacobian, and $f(z)$ is the p. d. f. of *z,*

and we get the acceptance probability is min{1, *A*} for the birth type.

Without loss of generality, since the death move type is the reverse of the birth move type, we can get the acceptance probability is min{1, *1/A*} for the death move type.

8. References

[1] Semiconductor Industry Association (2005), "*SIA 2005 Annual Report*," San Jose: SIA .

[2] P. K. Chatterjee and R. R. Doering, "The future of microelectronics," *Proceedings of the IEEE*, Vol. 86, No. 1, 1998, pp.176-183.

[3] P. J. Silverman, "Capital Productivity: Major Challenge for the Semiconductor Industry." *Solid State Technology*, Vol. 37, No. 3, 1994, pp.140.

[4] F. Bergeret and Y. Chandon, "Improving yield in IC manufacturing by statistical analysis of large data base," *Micro*, 1999, pp.59-75.

[5] W. H. Kruskal and W. A. Wallis, "Use of Ranks in One-Criterion Analysis of Variance," *Journal of the American Statistical Association*, 47, 1952, pp.583-621.

[6] G. Kong, "Tool commonality analysis for yield enhancement," in *Proc. 2002 IEEE/SEMI Advanced Semiconductor Manufacturing Conf.*, 2002, pp.202-205.

[7] L. K. Garling and G. P. Woods, "Determining equipment performance using analysis of variance," in *Proc. 1990 Int. Semiconductor Manufacturing Science Symp.*, 1990, pp.85-89.

[8] T. McCray, J. McNames, and D. Abercrombie, "Locating disturbances in semiconductor manufacturing with stepwise regression," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 18, No. 3, 2005, pp.458-468.

[9] R. A. Fisher, *Statistical Methods for Research Workers*, 13th edition. Oliver and Boyd, Edinburgh. 1958

[10] J. W. Tukey, *The Problem of Multiple Comparison*s, 1953, Mimeographed monograph.

[11] M. Keuls, “The use of the ‘Studentized range’ in connection with an analysis of variance, “ *Euphytica*, 1. 1952, pp.112-122.

[12] D. B. Duncan, “Multiple Range and Multiple F Tests,” *Biometrics*, 11, 1955, pp.1-42.

[13] H. Scheffe, “A Method for Judging All Contrasts in the Analysis of Variance,” *Biometrika*, 40, 1953, pp.87-104.

[14] C.W. Dunnett, ”A multiple comparison procedure for comparing several treatments with a control,” *Journal of the American Statistical Association*, 50, 1955, pp.613-621.

[15] C.W. Dunnett, ”Robust multiple comparisons,” *Commun. Statist.*, 11, 1982, pp.2611-2629.

[16] A. J. Scott and M. Knott, “A Cluster Analysis Method for Grouping Means in the Analysis of Variance,” *Biometrics*, 30, 1974, pp.507-512.

[17] K. J. Worsley, “A nonparametric extension of a cluster analysis method by Scott and Knott,” *Biometrics*, 33, 1977, pp.532-535.

[18] T. Calinski and L. C. A. Corsten, “Clustering Means in ANOVA by simultaneous testing,” *Biometrics*, 41, 1998, pp.39-48.

[19] I. T. Jolliffe, “Cluster Analysis as a Multiple Comparison Method.” In *Proc. Conf. Appl. Statist.* (ED. R.P. Gupta), Amsterdam: North Holland, 1975, pp.159-168.

[20] R. M. Gardner, J. Bieker, and S. Elwell, “Solving tough semiconductor manufacturing problems using data mining,” in *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conf. and Workshop*, Boston, MA, Sep. 2000, pp.46-55.

[21] F. Mieno, T. Sato, Y. Shibuya, K. Odagiri, H. Tsuda, and R. Take, “Yield improvement using data mining system,” in *Proc. IEEE Int. Symp. Semiconductor Manufacturing Conf.*, Santa Clara, CA, Oct. 1999, pp. 391-394.

[22] L. Breiman, J. Friedman, R., Olshen, and C. Stone, *Classification and regression trees*, Belmont, CA: Wadsworth, 1984.

[23] H. S. Stern, “Neural networks in applied statistics,” *Technometrics*, vol. 38, no. 3, 1996, pp.205-213.

[24] D.C. Montgomery, *Design and analysis of experiments*, 3rd edition. New York.

[25] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd edition. New York.

[26] P. J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, 82, 1995, pp.711-732.

[27] J. Maeritz and A. Schels, “Production enhancement of lithography through APC methods,” *Future Fab*, Vol.14, 2003.

[28] E. K. Lada, J. C. Lu and J. R. Wilson, “A Wavelet –Based Procedure for Process Fault Detection,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 15, No. 1, 2002, pp.79-90.

[29] S. T. Tseng, A. B. Yeh, F. Tsung, and Y. Y. Chan, “A study of variable EWMA controller,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 16, No. 4, 2003, pp.633-643.

[30] G. Consonni and P. Veronese, “A Bayesian method for combining results from several binomial experiments,” *Journal of the American Statistical Association*, 90, 1995, pp.935-944.

[31] S. Richardson and P. J. Green, “On Bayesian analysis of mixtures with an unknown number of components,” *J. R. Statist. Soc. B*, Vol. 59, No. 4, 1997, pp.731-792.

[32] A. Nobile and P. J. Green, “Bayesian analysis of factorial experiments by mixture modeling,” *Biometrika*, 87, 2003, pp.15-35.

[33] F. Bergeret and C. L. Gall, “Yield improvement using statistical analysis of process data,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 16, No. 3, 2003, pp.535-542.

[34] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith, “A Bayesian CART Algorithm,” *Biometrika*, 85, 1998, pp.363-377.

[35] H. A. Chipman, E. I. George, and R. E. McCulloch, “Bayesian CART Model Search,” *Journal of the American Statistical Association*, Vol. 93, No. 443, 1998, pp.935-948.

[36] H. Zhang, Comment on “Bayesian CART Model Search,” *Journal of the American Statistical Association*, Vol. 93, No. 443, 1998, pp.948-950.

[37] C. C. Holmes, D. G. T. Denison, S. Ray, and B. K. Mallick, “Bayesian Prediction via Partitioning,” *Journal of Computational and Graphical Statistics*, Vol. 14, No. 4, 1998, pp.811-830.

[38] Y. Wu, H. Tjelmeland, and M. West, “Bayesian CART: Prior Specification and Posterior Simulation,” *Journal of Computational and Graphical Statistics*, Vol. 16, No. 1, 2007, pp.44-66.

[39] C. Siddhartha, and G. Edward, “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, Vol. 49, 1995, pp.327-335.

[40] K. Knight, R. Kustra, and R. Tibshirani, Comment on “Bayesian CART Model Search,” *Journal of the American Statistical Association*, Vol. 93, No. 443, 1998, pp.950-954.

[41] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, “Markov chain Monte Carlo in practice,” in *Interdisciplinary Statistics*. London, U.K.: Chapman & Hall, 1996.

[42] G. Casella and E. I. George, “Explaining the Gibbs Sampler,” *The American Statistician*, Vol. 46, No. 3, 1998, pp.167-174.C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[43] A. Gelman, and D.B. Rubin, “Inference from Iterative Simulation Using Multipple Sequences,” *Statistical science,* Vol. 7, No. 4, 1992a, pp.457-511.

[44] S. P. Brooks, and A. Gelman, “General Methods for Monitoring Convergence of Iterative Simulations,” *Journal of Computational and Graphical Statistics,* Vol. 7, No. 4, 1998, pp.434-455.

[45] S. P. Brooks and P. Giudici, “Markov Chain Monte Carlo Convergence Assessment via Two-Way Analysis of Variance,” *Journal of Computational and Graphical Statistics,* Vol. 9, No. 2, 2000, pp.266-285

[46] L. A. Clark and D. Pregibon, “Tree based models,” In *Statistical Models* in S, eds. J. Chambers and T. Hastie, Belmont, CA: Wadsworth, 1992

[47] D. C. Montgomery, “*Introduction to Statictical Quality Control,*” 2nd edition , New York.

[48] H. Yosef and C. T. Ajit, “*Multiple Comparison Procedures,*” 2nd edition , New York.

[49] S. Gelman and D. Gelman, “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,” *IEEE Trans. Pattn. Anal. Mach. Intel.,* 6, 1984, pp.721-741.

[50] A. E. Gelfand and A. F. M. Smith, “Sampling-based approaches to calculating marginal densities,” *Journal of the American Statistical Association,* 85, 1990, pp.389-409.

[51] A. E. Gelfand, Hillls, Racine-Poon S. E. and A. F. M. Smith, “Illustration of Bayesian inference in normal data models using Gibbs sampling,” *Journal of the American Statistical Association,* Vol. 85, 1990, pp.972-985.