Postprint of article in *Information and Software Technology* **39** (10): 719–726 (1997)

On the Effectiveness of Test Case Allocation Schemes in Partition Testing *†

F. T. Chan ‡
School of Professional and Continuing Education
The University of Hong Kong
Pokfulam, Hong Kong
Tel. (+852) 2975 5651 Fax (+852) 2975 4953 Email hrxecft@hku.hk

T. Y. Chen
Department of Computer Science and Software Engineering
The University of Melbourne
Parkville 3052, Australia
Tel. (+61) 3 9287 9101 Fax (+61) 3 9348 1184 Email tyc@cs.mu.oz.au

T. H. Tse §
Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong
Tel. (+852) 2859 2183 Fax (+852) 2559 8447 Email thtse@cs.hku.hk

Abstract

In partition testing, the input domain is divided into two or more disjoint subdomains according to some criteria and then test data are selected from each subdomain. Hence we must design a partitioning scheme that governs how to divide the input domain, and a test case allocation scheme that controls how to allocate test cases to the subdomains. Given any partitioning scheme, the test case allocation scheme plays a crucial part in the effectiveness of testing. This paper develops guidelines for comparing different test case allocation schemes. This provides solutions to the class of problems where various test case allocation schemes under the same partitioning scheme are compared for the effectiveness of testing. We illustrate with examples several potential applications of our findings.

Keywords: Partition testing, subdomain testing, effectiveness of testing

† Supported in part by a grant of the Hong Kong Research Grants Council and a grant of the Australian Research Council.

‡ Corresponding author.

§ Part of this research was done when the third author was an HKU-MU Academic Exchange Visitor at the University of Melbourne.

1 Introduction

There are two common approaches to select inputs as test data, namely random testing and subdomain testing. Random testing simply selects test cases randomly from the entire input domain. Subdomain testing partitions the input domain into two or more subdomains according to some criteria that normally make use of the functional or structural information of the program. Test data are then selected from every subdomain either randomly or according to some guidelines. Examples of subdomain testing strategies are branch coverage, path coverage, and White and Cohen's domain testing strategy [1]. The subdomains can be overlapping or disjoint. For example, subdomains usually overlap for branch coverage but are disjoint for path coverage and the domain testing strategy. In the case of disjoint subdomains, the testing strategy is specifically referred to as partition testing.

Intuitively speaking, partition testing should be more effective in revealing program errors than random testing. However, Duran and Ntafos [2] as well as Hamlet and Taylor [3] observed in their simulation investigations that there was only a marginal difference in effectiveness between the two methods. Random testing might even be more cost effective than partition testing when the partitioning and associated costs of partition testing were high.

Their counter-intuitive finding motivated the analytical investigations of Weyuker and Jeng [4], who used a P-measure as an effectiveness metric for testing. It is defined as the probability of detecting at least one failure. In their model, the selection of test cases from the entire input domain or the subdomains were random, independent, with replacement, and based on a uniform distribution. In order to provide a fair comparison between the P-measures of partition testing and random testing, the numbers of test cases for the two methods were chosen to be equal. Their results provided insights on how partition testing could be better than, as good as, or worse than random testing with respect to the P-measure. They also revealed that if all subdomains were of equal size and the same number of test cases were selected from each subdomain, then the P-measure of partition testing was no less than that of random testing. In this paper, this is referred to as the equal-size and equal-number strategy.

Based on the same mathematical model, Chen and Yu [5, 6] generalized the results of Weyuker and Jeng and found several sufficient conditions for partition testing to have an equal or higher P-measure than random testing. One of these was known as the proportional sampling strategy, where the number of test cases selected from each subdomain was proportional to the size of the corresponding subdomain. Obviously, the equal-size and equal-number strategy of Weyuker and Jeng was a special case of this.

Prior to the studies of Weyuker and Jeng, and Chen and Yu, most researches in subdomain testing focused only on how to partition the input domain into subdomains (referred to as the partitioning scheme in this paper). For the allocation of test cases to the subdomains (referred to as the test case allocation scheme), the most commonly adopted approach was the intuitive heuristics of simply choosing at least one from each subdomain. The domain testing strategy of White and Cohen [1] was one of the few testing strategies that addressed how test cases should be selected from a subdomain. In order to detect subdomain boundary faults, they recommended to select test cases on the subdomain boundaries and test cases close to the subdomain boundaries. Weyuker and Jeng as well as Chen and Yu were the first to demonstrate that test case allocation schemes played a crucial part in affecting the P-measures.

In addition to the P-measure, Chen and Yu [7] used another effectiveness metric known as the E-measure, defined as the expected number of failures detected. They again identified several sufficient conditions for partition testing to have higher or equal E-measures than random testing. Moreover, the relationship between the P-measures and E-measures was investigated.

Instead of comparing partition testing and random testing, this paper makes a comparison of various test case allocation schemes in partition testing. In other words, for a given partitioning scheme, we would like to improve the P-measure or E-measure by adjusting the test case allocation scheme. As an illustration, suppose a partitioning scheme divides the input domain into two disjoint subdomains of sizes 64 and 36, respectively, and the total number of test cases is 10. According to the proportional sampling strategy of Chen and Yu, 6.4 test cases should be allocated to the first subdomain and 3.6 to the second. Obviously, fractional test cases are not possible. Chen and Yu [8] proved that in order for a partitioning strategy to be better than random testing, its test case allocation scheme could not deviate from that of the proportional sampling strategy other than rounding up or down. In this example, it means a 6 and 4 split or a 7 and 3 split. However, their study did not suggest any further guideline to choose between these two allocation schemes. In fact, such a situation belongs to a more general class of problems where various test case allocation schemes are compared for their effectiveness under the same partitioning scheme and the same program. This paper attempts to provide some solutions to this class of problems.

In the next section, the basic concepts and notation are defined. Section 3 describes earlier results relevant to this study. Section 4 states the main results of comparing different test case allocation schemes. Section 5 discusses some possible applications and Section 6 gives the concluding remarks.

2 Basic Concepts and Notation

The assumptions and notation used here basically follow those of Chen and Yu [5]. All random selections of test cases are based on a uniform distribution, independent of one another and drawn with replacement. In this paper, we are only concerned with partition testing, that is, the case when the subdomains are disjoint. For a fair comparison, the same total number of test cases is used for each testing method and each test case allocation scheme.

Let D denote the input domain of a program. The elements of D that produce incorrect outputs, hence revealing program errors, are called failure-causing inputs. The variables d, m, and n are used to denote the size of the input domain, size of the failure-causing inputs, and the total number of test cases, respectively. The failure rate θ and sampling rate σ are defined as $\theta = \frac{m}{d}$ and $\sigma = \frac{n}{d}$, respectively.

For random testing, the probability of detecting at least one failure (P-measure), denoted by P_r, is equal to $1-(1-\theta)^n$ and the expected number of failures detected (E-measure), denoted by E_r, is equal to $n\theta$.

For any partition testing strategy P, the k disjoint subdomains formed are denoted by D_i, $i = 1, 2, \ldots, k$, where $k \geq 2$. For any subdomain D_i, let d_i, m_i, and n_i denote the size, the number of failure-causing inputs, and the number of test cases selected in this subdomain, respectively. Its failure rate θ_i and sampling rate σ_i are defined as $\theta_i = \frac{m_i}{d_i}$ and $\sigma_i = \frac{n_i}{d_i}$, respectively. The P-measure, denoted by P_p, is equal to $1-\prod_{i=1}^{k}(1-\theta_i)^{n_i}$. The E-measure, denoted by E_p, is equal to $\sum_{i=1}^{k} n_i\theta_i$.

Since the total number of test cases is assumed to be the same, we have $n = \sum_{i=1}^{k} n_i$.

3 Related Findings

In their analytical study of partition testing, Weyuker and Jeng [4] proved that if $d_1 = d_2 = \cdots = d_k$ and $n_1 = n_2 = \cdots = n_k$, then $P_p \geq P_r$. It was the first derived sufficient condition for $P_p \geq P_r$. However, it is not very useful in practice as we seldom have all subdomains with the same size.

Chen and Yu [5] generalized Weyuker and Jeng's result and proved the proportional sampling strategy. In formal notation, if $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, then $P_p \geq P_r$. Since $\sigma_i = \sigma_j$ implies $\frac{n_i}{n_j} = \frac{d_i}{d_j}$, it is in fact not necessary to know the absolute sizes of the subdomains in order to apply the proportional sampling strategy. We only need to know the relative sizes of the subdomains to determine the values of n_i's. For example, if $\frac{d_1}{d_2} = \frac{1}{2}$, the proportional sampling strategy just requires the number of test cases in D_2 to be twice as many the number of test cases in D_1. Chan et al. [9] provided some guidelines on how to use the proportional sampling strategy in practice. Chen and Yu [6] further proved that if $(\theta_i - \theta_j)(\sigma_i - \sigma_j) \geq 0$ for all $i, j = 1, 2, \ldots, k$, then $P_p \geq P_r$, and that if $(\theta_i - \theta)(\sigma_i - \sigma) \geq 0$ for all $i = 1, 2, \ldots, k$, then $P_p \geq P_r$. Obviously, when $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, the above two conditions are also satisfied. Hence, the proportional sampling strategy is a special case of these two conditions. It should be noted that, although these conditions are more general, we need some knowledge of the failure rates of the subdomains before they can be applied. Chen and Yu [7] also found very similar relations for the E-measure. For example, if $(\theta_i - \theta_j)(\sigma_i - \sigma_j) \geq 0$ for all $i, j = 1, 2, \ldots, k$, then $E_p \geq E_r$. If $(\theta_i - \theta)(\sigma_i - \sigma) \geq 0$ for all $i = 1, 2, \ldots, k$, then $E_p \geq E_r$. The reverse implications are also true for the E-measure. That is, if $(\theta_i - \theta_j)(\sigma_i - \sigma_j) \leq 0$ for all $i, j = 1, 2, \ldots, k$, then $E_p \leq E_r$; and if $(\theta_i - \theta)(\sigma_i - \sigma) \leq 0$ for all $i = 1, 2, \ldots, k$, then $E_p \leq E_r$. When $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, we have $E_p = E_r$.

All the above results only deal with the comparison of effectiveness between random testing and partition testing. In this paper, we are interested in comparing the effectiveness of different test case allocation schemes of partition testing, assuming the same partitioning scheme on the same program.

4 Test case allocation schemes comparisons

For a given partitioning scheme, we would like to compare the P-measures of various test case allocation schemes. Suppose there are two test case allocation schemes A and B. Let n_{A_i} and n_{B_i} denote the number of test cases selected from D_i of schemes A and B, respectively. We use P_A and P_B to denote the P-measures of schemes A and B, respectively.

Proposition 1 (partial sums condition)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ *and* $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. *If* $\sum_{i=1}^{m} n_{A_i} \geq \sum_{i=1}^{m} n_{B_i}$ *for all* $m = 1, 2, \ldots, (k-1)$, *then* $P_A \geq P_B$.

Proof

$$
\begin{aligned}
P_A &= 1 - (1-\theta_1)^{n_{A_1}}(1-\theta_2)^{n_{A_2}} \cdots (1-\theta_k)^{n_{A_k}} \\
&= 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_1)^{n_{A_1}-n_{B_1}}(1-\theta_2)^{n_{A_2}} \cdots (1-\theta_k)^{n_{A_k}} \\
&\geq 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{A_1}-n_{B_1}}(1-\theta_2)^{n_{A_2}} \cdots (1-\theta_k)^{n_{A_k}} \\
&= 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_2)^{(n_{A_1}+n_{A_2})-(n_{B_1}+n_{B_2})}(1-\theta_3)^{n_{A_3}} \cdots (1-\theta_k)^{n_{A_k}} \\
&\geq 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_3)^{(n_{A_1}+n_{A_2})-(n_{B_1}+n_{B_2})}(1-\theta_3)^{n_{A_3}} \cdots (1-\theta_k)^{n_{A_k}} \\
&= 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_3)^{n_{B_3}}(1-\theta_3)^{(n_{A_1}+n_{A_2}+n_{A_3})-(n_{B_1}+n_{B_2}+n_{B_3})} \\
&\quad \cdots (1-\theta_k)^{n_{A_k}} \\
&\cdots \\
&= 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_3)^{n_{B_3}} \cdots \\
&\quad (1-\theta_{k-1})^{n_{B_{k-1}}}(1-\theta_{k-1})^{(n_{A_1}+n_{A_2}+\cdots+n_{A_{k-1}})-(n_{B_1}+n_{B_2}+\cdots+n_{B_{k-1}})}(1-\theta_k)^{n_{A_k}} \\
&\geq 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_3)^{n_{B_3}} \cdots \\
&\quad (1-\theta_{k-1})^{n_{B_{k-1}}}(1-\theta_k)^{(n_{A_1}+n_{A_2}+\cdots+n_{A_{k-1}})-(n_{B_1}+n_{B_2}+\cdots+n_{B_{k-1}})}(1-\theta_k)^{n_{A_k}} \\
&= 1 - (1-\theta_1)^{n_{B_1}}(1-\theta_2)^{n_{B_2}}(1-\theta_3)^{n_{B_3}} \cdots (1-\theta_{k-1})^{n_{B_{k-1}}}(1-\theta_k)^{n_{B_k}} \\
&= P_B
\end{aligned}
$$

□

For convenience, we refer to the condition when $\sum_{i=1}^{m} n_{A_i} \geq \sum_{i=1}^{m} n_{B_i}$ for all $m = 1,\ 2,\ \ldots,\ (k-1)$ as the partial sums condition.

Is it possible to obtain such information? In practice, the sizes of the failure causing inputs and their distribution in the subdomains are normally not available. We are, however, only interested in knowing the relative ordering rather than the exact failure rates of the subdomains. There are many possible ways to estimate such relative ordering based on expert judgement using information from other sources. For example, if the input domain is partitioned in such a way that each subdomain corresponds to one function of the software specification, it would be reasonable to assume that a subdomain associated with a more complex function has a larger failure rate than the one associated with a simpler function. In general, we can try to establish some metric to give a quantitative measure for each subdomain. The metric should be reflecting the effort or complexity of the associated software development. The measures obtained provide an estimation of the relative ordering of the failure rates of the subdomains. Alternatively, the history of program failure occurrences of a software system could be recorded. When the software is revised and testing is performed, this past history can be used as an estimate of the relative ordering of the subdomain failure rates as well. One can argue that when the relative ordering of the failure rates of all subdomains is known, the highest P-measure and E-measure can be achieved by simply selecting test cases from the subdomain with the highest failure rate. In spite of this obvious and simple fact, test cases are never selected in such a way in practice. First, for a program with more than one fault, it is unlikely that all failure-causing inputs will fall into the same subdomain. Selecting all test cases from a single subdomain would never reveal program faults whose failure-causing inputs lie outside the subdomain. Such a practice is therefore not recommended because the ultimate goal of testing is to reveal all program faults. Secondly, the P-measure and E-measure are normally not the only nor the most important testing criteria that the software testers are concerned with. There may be other more important testing criteria or measures. For example, a tester may wish to have a test suite that will test all functionalities up to a certain coverage level. These criteria and measures may be given a higher priority than P-measures or E-measures. Some illustrative examples will be given in the next section. There exist some other interesting relationships that are also sufficient conditions for $P_A \geq P_B$. In fact, these conditions are special cases of the partial sums condition.

Proposition 2 (differences condition)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ *and* $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. *If*

> *there exists some* h*, where* $0 < h < k$*, such that* $n_{A_i} \geq n_{B_i}$ *for all* $i = 1,\ 2,\ \ldots,\ h$ *and* $n_{A_i} \leq n_{B_i}$ *for all* $i = (h+1),\ (h+2),\ \ldots,\ k$*,* — (1)

then $\sum_{i=1}^{m} n_{A_i} \geq \sum_{i=1}^{m} n_{B_i}$ *for all* $m = 1,\ 2,\ \ldots,\ (k-1)$*.*

Proof

(a) $m = 1$ to h:

$\sum_{i=1}^{m} n_{A_i} - \sum_{i=1}^{m} n_{B_i} = \sum_{i=1}^{m} (n_{A_i} - n_{B_i}) \geq 0$, because $n_{A_i} \geq n_{B_i}$ for all $i = 1,\ 2,\ \ldots,\ h$.

(b) $m = (h+1)$ to $(k-1)$:

$\sum_{i=m}^{k} n_{A_i} - \sum_{i=m}^{k} n_{B_i} = \sum_{i=m}^{k} (n_{A_i} - n_{B_i}) \leq 0$, because $n_{A_i} \leq n_{B_i}$ for all $i = (h+1),\ (h+2),\ \ldots,\ k$.

Hence,

$$\sum_{i=1}^{m} n_{A_i} - \sum_{i=1}^{m} n_{B_i} = \sum_{i=1}^{m} n_{A_i} - \sum_{i=1}^{k} n_{B_i} + \sum_{i=m+1}^{k} n_{B_i} = \sum_{i=1}^{m} n_{A_i} - \sum_{i=1}^{k} n_{A_i} + \sum_{i=m+1}^{k} n_{B_i}$$
$$= -\sum_{i=m+1}^{k} n_{A_i} + \sum_{i=m+1}^{k} n_{B_i} = -\sum_{i=m+1}^{k} (n_{A_i} - n_{B_i}) \geq 0.$$

From (a) and (b), therefore, we have $\sum_{i=1}^{m} n_{A_i} \geq \sum_{i=1}^{m} n_{B_i}$ for all $m = 1, 2, \ldots, (k-1)$. □

Let us refer to Condition (1) as the differences condition. It follows immediately from Propositions 1 and 2 that the differences condition is also a sufficient condition for $P_A \geq P_B$.

Corollary 1
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ and $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. If there exists some h, where $0 < h < k$, such that $n_{A_i} \geq n_{B_i}$ for all $i = 1, 2, \ldots, h$ and $n_{A_i} \leq n_{B_i}$ for all $i = (h+1), (h+2), \ldots, k$, then $P_A \geq P_B$.

Consider the test case allocation schemes A and B such that $n_{A_1} = 10$, $n_{A_2} = 5$, $n_{A_3} = 8$, $n_{A_4} = 2$, $n_{B_1} = 8$, $n_{B_2} = 6$, $n_{B_3} = 7$, and $n_{B_4} = 4$. Obviously, the partial sums condition holds but not the differences condition. Hence, the differences condition is a special case of the partial sums condition.

The following are some interesting special cases of the differences condition, and hence also special cases of the partial sums condition.

Proposition 3 (decreasing differences condition)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ and $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. If $(n_{A_1} - n_{B_1}) \geq (n_{A_2} - n_{B_2}) \geq \cdots \geq (n_{A_k} - n_{B_k})$, then there exists some h, where $0 < h < k$, such that $n_{A_i} \geq n_{B_i}$ for all $i = 1, 2, \ldots, h$ and $n_{A_i} \leq n_{B_i}$ for all $i = (h+1), (h+2), \ldots, k$.

Proof
Since $(n_{A_i} - n_{B_i})$ is a monotonically decreasing series, the proof is complete if we can show that $(n_{A_1} - n_{B_1}) \geq 0$ and $(n_{A_k} - n_{B_k}) \leq 0$. By then, there must be a turning point, say h, such that $(n_{A_i} - n_{B_i}) \geq 0$ for $i = 1, 2, \ldots, h$ and $(n_{A_i} - n_{B_i}) \leq 0$ for $i = (h+1), (h+2), \ldots, k$.

To prove $(n_{A_1} - n_{B_1}) \geq 0$, let us assume the opposite and show that it will lead to contradiction. It follows that $(n_{A_i} - n_{B_i}) < 0$ for $i = 1, 2, \ldots, k$ because $(n_{A_1} - n_{B_1}) \geq (n_{A_2} - n_{B_2}) \geq \cdots \geq (n_{A_k} - n_{B_k})$. Then $\sum_{i=1}^{k} n_{A_i} < \sum_{i=1}^{k} n_{B_i}$, which contradicts $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. Hence, $(n_{A_1} - n_{B_1}) \geq 0$.

Suppose $(n_{A_k} - n_{B_k}) > 0$. Then $(n_{A_i} - n_{B_i}) > 0$ for $i = 1, 2, \ldots, k$. Hence, $\sum_{i=1}^{k} n_{A_i} > \sum_{i=1}^{k} n_{B_i}$, which contradicts $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. Therefore, $(n_{A_k} - n_{B_k}) \leq 0$.

Hence, the proof is done. □

The condition $(n_{A_1} - n_{B_1}) \geq (n_{A_2} - n_{B_2}) \geq \cdots \geq (n_{A_k} - n_{B_k})$ is referred to as the decreasing differences condition.

Similarly, we define the sampling rates condition as $(\sigma_{A_1} - \sigma_{B_1}) \geq (\sigma_{A_2} - \sigma_{B_2}) \geq \cdots \geq (\sigma_{A_k} - \sigma_{B_k})$. It can be easily shown that this condition also implies the differences condition. Hence, we have the following proposition.

Proposition 4 (sampling rates condition)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ and $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. If $(\sigma_{A_1} - \sigma_{B_1}) \geq (\sigma_{A_2} - \sigma_{B_2}) \geq \cdots \geq (\sigma_{A_k} - \sigma_{B_k})$, then there exists some h, where $0 < h < k$, such that $n_{A_i} \geq n_{B_i}$ for all $i = 1, 2, \ldots, h$ and $n_{A_i} \leq n_{B_i}$ for all $i = (h+1), (h+2), \ldots, k$.

The proof of this proposition is very similar to the proof of Proposition 3. Hence, it is not repeated here.

Let us refer to the condition $\frac{n_{A_1}}{n_{B_1}} \geq \frac{n_{A_2}}{n_{B_2}} \geq \cdots \geq \frac{n_{A_k}}{n_{B_k}}$ as the decreasing ratios condition. This also turns out to be a special case of the differences condition.

Proposition 5 (decreasing ratios condition)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ *and* $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. *If* $\frac{n_{A_1}}{n_{B_1}} \geq \frac{n_{A_2}}{n_{B_2}} \geq \cdots \geq \frac{n_{A_k}}{n_{B_k}}$, *then there exists some* h, *where* $0 < h < k$, *such that* $n_{A_i} \geq n_{B_i}$ *for all* $i = 1, 2, \ldots, h$ *and* $n_{A_i} \leq n_{B_i}$ *for all* $i = (h+1), (h+2), \ldots, k$.

Again, the proof of the above proposition is similar to the proof of Proposition 3 and is not repeated.

Since $\frac{\sigma_{A_i}}{\sigma_{B_i}} = \frac{n_{A_i}}{d_i} \times \frac{d_i}{n_{B_i}} = \frac{n_{A_i}}{n_{B_i}}$, the decreasing ratios condition can be expressed equivalently as $\frac{\sigma_{A_1}}{\sigma_{B_1}} \geq \frac{\sigma_{A_2}}{\sigma_{B_2}} \geq \cdots \geq \frac{\sigma_{A_k}}{\sigma_{B_k}}$. It follows immediately from Corollary 1 and Propositions 3 to 5 that, given the decreasing differences condition, the sampling rates condition, *or* the decreasing ratios condition, P_A will be no less than P_B.

Corollary 2
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ *and* $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. *If the decreasing differences condition, the sampling rates condition, or the decreasing ratios condition holds, then* $P_A \geq P_B$.

To summarize, we know that the differences condition is not equivalent to, but a special case of, the partial sums condition. Furthermore, the other three conditions (decreasing differences, sampling rates, and decreasing ratios) in Propositions 3 to 5 have been shown to imply the differences condition. An obvious question is whether any of these three conditions is in fact equivalent to the differences condition. The following example demonstrates that the answer is no.

Example 1
Suppose $k = 4$, $d_1 = 50$, $d_2 = 100$, $d_3 = 200$, and $d_4 = 800$. Let $n_{A_1} = 10$, $n_{A_2} = 18$, $n_{A_3} = 7$, $n_{A_4} = 5$, $n_{B_1} = 9$, $n_{B_2} = 15$, $n_{B_3} = 6$, and $n_{B_4} = 10$. It can easily be checked that the differences condition holds but not any of the other three conditions. ◇

The next question is whether any of these three conditions will imply the others. The following example demonstrates that they are independent of each other.

Example 2
Suppose $k = 4$, $d_1 = 50$, $d_2 = 100$, $d_3 = 200$, and $d_4 = 800$.

Case 1:

Let $n_{A_1} = 17$, $n_{A_2} = 12$, $n_{A_3} = 4$, $n_{A_4} = 7$, $n_{B_1} = 15$, $n_{B_2} = 10$, $n_{B_3} = 5$, and $n_{B_4} = 10$. Only the decreasing differences condition holds but not the sampling rates and decreasing ratios conditions.

Case 2:

Let $n_{A_1} = 17$, $n_{A_2} = 13$, $n_{A_3} = 4$, $n_{A_4} = 6$, $n_{B_1} = 15$, $n_{B_2} = 10$, $n_{B_3} = 5$, and $n_{B_4} = 10$. Only the sampling rates condition holds but not the other two conditions.

Case 3:

Let $n_{A_1} = 21$, $n_{A_2} = 8$, $n_{A_3} = 4$, $n_{A_4} = 7$, $n_{B_1} = 15$, $n_{B_2} = 10$, $n_{B_3} = 5$, and $n_{B_4} = 10$. Only the decreasing ratios condition holds but not the other two conditions. ◇

It appears that we need to know the relative ordering of all the subdomain failure rates in order to make use of the above results. In fact, this is not the case. These results can be applied even if the ordering of the failure rates of the subdomains are only partially known. A more general version of the proposition is given below.

Suppose we organize the subdomains into disjoint groups such that within each group there is a total order for the failure rates. Then we have the following proposition:

Proposition 6 (partial orders condition)
Suppose the subdomains $D_1,\ D_2,\ \ldots,\ D_k$ are organized into y disjoint groups, denoted by $G_i = \{D^i_1,\ D^i_2,\ \ldots,\ D^i_{r_i}\}$ for $i = 1,\ 2,\ \ldots,\ y$, such that for each G_i,

(a) $\theta^i_1 \geq \theta^i_2 \geq \cdots \geq \theta^i_{r_i}$, *and*

(b) $\sum_{j=1}^{r_i} n_{A^i_j} = \sum_{j=1}^{r_i} n_{B^i_j}$, *where* $n_{A^i_j}$ *and* $n_{B^i_j}$ *are the number of test cases selected from subdomain* D^i_j *under test case allocation schemes A and B, respectively.*

If the set of $n_{A^i_j}$*'s and the set of* $n_{B^i_j}$*'s for each* G_i *observe the condition in Proposition 1, then* $P_A \geq P_B$.

Proof
Let the probabilities of detecting at least one failure for the group of subdomains, G_i, be P_{A_i} and P_{B_i} under the schemes A and B, respectively. It follows from Proposition 1 that $P_{A_i} \geq P_{B_i}$. Since $P_A = 1 - \prod_{i=1}^{y}(1 - P_{A_i})$ and $P_B = 1 - \prod_{i=1}^{y}(1 - P_{B_i})$, we have $P_A \geq P_B$. □

Hence, as long as two test case allocation schemes use the same number of test cases in each respective group of subdomains, we can still draw useful conclusions.

As a note, when $y = 1$, we have a total order for the failure rates of the subdomains. When $y = k$, the ordering of the failure rates is completely unknown and there is only one subdomain in each G_i.

We recall that there is another effectiveness metric E-measure [7], which is the expected number of failures detected. The propositions and corollaries for the P-measure stated above have counterparts for the E-measure. In other words, all the previous propositions and corollaries will still hold when the P-measures are replaced by the corresponding E-measures. It is sufficient to prove that Proposition 1 holds for the E-measure. The other propositions and corollaries will follow automatically as they are special cases of the sufficient condition in Proposition 1. For a given partitioning scheme and two test case allocation schemes A and B, let n_{A_i} and n_{B_i} denote the number of test cases selected from D_i of schemes A and B, respectively, and E_A and E_B denote the E-measures of schemes A and B, respectively.

Proposition 7 (partial sums condition for E-measure)
Suppose $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ *and* $\sum_{i=1}^{k} n_{A_i} = \sum_{i=1}^{k} n_{B_i}$. *If* $\sum_{i=1}^{m} n_{A_i} \geq \sum_{i=1}^{m} n_{B_i}$ *for all* $m = 1,\ 2,\ \ldots,\ (k-1)$, *then* $E_A \geq E_B$.

Proof

$$
\begin{aligned}
E_A &= n_{A_1}\theta_1 + n_{A_2}\theta_2 + \cdots + n_{A_k}\theta_k \\
&= n_{B_1}\theta_1 + (n_{A_1} - n_{B_1})\theta_1 + n_{A_2}\theta_2 + \cdots + n_{A_k}\theta_k \\
&\geq n_{B_1}\theta_1 + (n_{A_1} - n_{B_1})\theta_2 + n_{A_2}\theta_2 + \cdots + n_{A_k}\theta_k \\
&= n_{B_1}\theta_1 + n_{B_2}\theta_2 + ((n_{A_1} + n_{A_2}) - (n_{B_1} + n_{B_2}))\theta_2 + n_{A_3}\theta_3 + \cdots + n_{A_k}\theta_k \\
&\geq n_{B_1}\theta_1 + n_{B_2}\theta_2 + ((n_{A_1} + n_{A_2}) - (n_{B_1} + n_{B_2}))\theta_3 + n_{A_3}\theta_3 + \cdots + n_{A_k}\theta_k \\
&= n_{B_1}\theta_1 + n_{B_2}\theta_2 + n_{B_3}\theta_3 + ((n_{A_1} + n_{A_2} + n_{A_3}) - (n_{B_1} + n_{B_2} + n_{B_3}))\theta_3 + \cdots + n_{A_k}\theta_k \\
&\ldots
\end{aligned}
$$

$$
\begin{aligned}
&= n_{B_1}\theta_1 + n_{B_2}\theta_2 + n_{B_3}\theta_3 + \cdots \\
&\quad + n_{B_{k-1}}\theta_{k-1} + ((n_{A_1} + n_{A_2} + \cdots + n_{A_{k-1}}) - (n_{B_1} + n_{B_2} + \cdots + n_{B_{k-1}}))\theta_{k-1} + n_{A_k}\theta_k \\
&\geq n_{B_1}\theta_1 + n_{B_2}\theta_2 + n_{B_3}\theta_3 + \cdots + n_{B_{k-1}}\theta_{k-1} \\
&\quad + ((n_{A_1} + n_{A_2} + \cdots + n_{A_{k-1}}) - (n_{B_1} + n_{B_2} + \cdots + n_{B_{k-1}}))\theta_k + n_{A_k}\theta_k \\
&= n_{B_1}\theta_1 + n_{B_2}\theta_2 + n_{B_3}\theta_3 + \cdots + n_{B_{k-1}}\theta_{k-1} + n_{B_k}\theta_k \\
&= E_B
\end{aligned}
$$

□

5 Applications

We use two case studies to illustrate some applications of the results described in the previous section. For simplicity and clarity, only P-measures are referred to in these case studies. The discussion is, however, applicable to E-measures as well.

Case Study 1
Assume the input domain of the program being tested is divided into four disjoint subdomains (D_1, D_2, D_3, and D_4) based on the functional specification of the program, and it is decided to run a total of 100 test cases. The primary testing objective is that, subject to integral constraints, the number of test cases allocated to a subdomain is proportional to the number of statements associated with it. Suppose the numbers of statements associated with D_1, D_2, D_3, and D_4 are 1500, 800, 950, and 350, respectively. Thus, 41.67, 22.22, 26.39, and 9.72 test cases should be allocated to the respective subdomains. The feasible schemes, subject to rounding and truncation, are as follows:

	Number of Test Cases in each Subdomain			
Scheme	n_1	n_2	n_3	n_4
A	42	23	26	9
B	42	22	27	9
C	42	22	26	10
D	41	23	27	9
E	41	23	26	10
F	41	22	27	10

When the relative ordering of the θ_i's is known, we are able to select the one that will give the highest P-measure from these six schemes. Say, if $\theta_1 \geq \theta_2 \geq \theta_3 \geq \theta_4$, Scheme A should be selected according to Proposition 2.

Even if we do not know the relative ordering of all the θ_i, the propositions may still be useful. Suppose we only know that $\theta_1 \geq \theta_2$ and $\theta_3 \geq \theta_4$. Proposition 6 tells us that Scheme B is better than Schemes C, D, and E. We can then confine the selection from Schemes A, B, and F. Thus, we are able to shortlist the test case allocation schemes by applying the propositions. ◇

Case Study 2
Suppose each subdomain has a desirable testing coverage. For example, the desirable number of test cases of a subdomain is a certain percentage of the number of statements associated with that subdomain,

or is a factor of the estimated complexity of the subdomain. Let the primary testing objective be to allocate at least one test case to each subdomain and the secondary testing objective be to allocate test cases to satisfy as many desirable values of the subdomains as possible. After satisfying the primary and secondary objectives, we would then like to maximize the effectiveness of testing in terms of the P-measure.

When the total number of test cases is greater than or equal to the sum of all desirable values, the allocation of test cases is trivial. First, we simply allocate the desirable values to each subdomain. We then allocate all the remaining test cases to the subdomain with the highest failure rate.

Therefore, we need only consider the case when the total number of test cases is smaller than the sum of all desirable values. In this case, we can only achieve the desirable values of some subdomains. We first allocate one test case to each subdomain. We then work out a feasible way of allocating test cases so that the desirable values are satisfied for the maximum number of subdomains. Finally, we allocate all the remaining test cases to the subdomain with the highest failure rate among those subdomains whose desirable values have not yet been reached. The following scenarios illustrate the feasible schemes for the given conditions as well as the selection process.

Scenario (a):
Total number of test cases = 40

Subdomains	D_1	D_2	D_3	D_4	D_5	D_6
Desirable number of test cases	19	2	17	3	20	18

We can see that only at most three desirable values can be satisfied. Assuming $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_6$, we have the following possible test case allocation schemes:

	Number of Test Cases in each Subdomain					
Scheme	n_1	n_2	n_3	n_4	n_5	n_6
A	19	2	14	3	1	1
B	16	2	17	3	1	1
C	13	2	1	3	20	1
D	15	2	1	3	1	18
E	1	2	17	1	1	18

It can be concluded easily from Proposition 1 that Scheme A gives the best P-measure among all the possible test case allocation schemes as its partial sums are greater than or equal to the corresponding partial sums of all the other schemes.

Scenario (b):
Total number of test cases = 40

Subdomains	D_1	D_2	D_3	D_4	D_5	D_6
Desirable number of test cases	16	17	13	5	18	6

In this case, at most three desirable values can be satisfied. Assuming $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_6$, we have the following possible test case allocation schemes.

	Number of Test Cases in each Subdomain					
Scheme	n_1	n_2	n_3	n_4	n_5	n_6
A	16	4	13	5	1	1
B	16	3	13	1	1	6
C	16	11	1	5	1	6
D	3	17	13	5	1	1
E	2	17	13	1	1	6
F	10	17	1	5	1	6
G	2	1	13	5	18	1
H	14	1	13	5	1	6
I	1	1	13	1	18	6
J	9	1	1	5	18	6

Based on the propositions, it can be derived that Scheme A is better than Schemes B, D, E, G, H, I, and J, and Scheme C is better than Schemes F, H, I, and J. However, it cannot be decided whether Scheme A is better than Scheme C.

When only the relative order of the subdomain failure rates is known, it is not always possible to determine the scheme that has the highest P-measure. For example, Scheme A can be better than Scheme C in certain situations but not in other cases. When $\theta_1 = 0.03$, $\theta_2 = 0.02$, $\theta_3 = 0.018$, $\theta_4 = 0.015$, $\theta_5 = 0.01$, and $\theta_6 = 0.005$, we have $P_A = 0.5914 > 0.5698 = P_C$. However, when $\theta_1 = 0.03$, $\theta_2 = 0.02$, $\theta_3 = 0.015$, $\theta_4 = 0.01$, $\theta_5 = 0.01$, and $\theta_6 = 0.00799$, we have $P_A = 0.5652 < 0.5653 = P_C$.

If we look at the E-measures of the above two situations, for the first set of θ_i's, $E_A = 0.884$ and $E_C = 0.833$. For the second set of θ_i's, $E_A = 0.82299$ and $E_C = 0.82294$. In both cases, E_A is greater than E_C. Hence, the P-measures and E-measures of two different test case allocation schemes might not always follow the same direction of comparison results. To select an 'optimal' test case allocation scheme is, therefore, not a straightforward task. $\diamond$

6 Conclusion

This paper investigates the effectiveness of different test case allocation schemes for a given partitioning scheme, using the probability of detecting at least one failure (P-measure) and the expected number of failures detected (E-measure) as the metrics. We identify several sufficient conditions for a test case allocation scheme to be more effective than another.

The application of these sufficient conditions depends on some knowledge of the failure rates of the subdomains. However, it is not the absolute values of the subdomain failure rates that are needed but just their relative ordering. While these absolute values are not known in practice, the relative ordering can reasonably be estimated in a number of ways using other sources of information. Examples are the professional judgement of the software testers, the past testing history in the case of revising a program already in production run, and some form of complexity measures associated with the subdomains. Further investigations on the reliable estimation of this relative ordering will be very useful in the application of partition testing. It should also be noted that even partial information about the relative ordering of the subdomain failure rates is useful. If the relative order of the failure rates of some, but not all, subdomains is available, the propositions and corollaries can be applied as well.

Although allocating all test cases solely to the subdomain with the highest failure rate will give the highest P-measure and E-measure than any other test case allocation schemes, it is not adopted for many reasons. First, to develop a test suite with such a strategy may jeopardize the testing objective of detecting all program faults. Also, the P-measure and E-measure are not the only testing criteria of measuring the effectiveness of software testing, nor necessarily the most important ones.

For example, it is common that all the functions of a software should be tested up to a certain minimum level of coverage. It has been illustrated that in many occasions more than one test case allocation scheme can be employed when the P-measure or E-measure is not yet taken into consideration. Under such circumstances, we can base on these measures to fine tune the test case allocation schemes. The propositions and corollaries can be used to guide the selection of the ones giving higher P-measures or E-measures, hence producing better overall testing performance.

Sometimes, the comparison of different test case allocation schemes may not give a totally ordered list, but several partially ordered lists. In such a case, an optimal scheme cannot be decided. Nevertheless, we can still narrow down the choice of a scheme to a few local optima.

References

[1] White, L J and Cohen, E I, 'A domain strategy for computer program testing' *IEEE Transactions on Software Engineering* Vol 6 No 3 (1980) pp 247–257.

[2] Duran, J W and Ntafos, S C, 'An evaluation of random testing' *IEEE Transactions on Software Engineering* Vol 10 No 4 (1984) pp 438–444.

[3] Hamlet, R and Taylor, R, 'Partition testing does not inspire confidence' *IEEE Transactions on Software Engineering* Vol 16 No 12 (1990) pp 1402–1411.

[4] Weyuker, E J and Jeng, B, 'Analyzing partition testing strategies' *IEEE Transactions on Software Engineering* Vol 17 No 7 (1991) pp 703–711.

[5] Chen, T Y and Yu, Y T, 'On the relationship between partition and random testing' *IEEE Transactions on Software Engineering* Vol 20 No 12 (1994) pp 977–980.

[6] Chen, T Y and Yu, Y T, 'A more general sufficient condition for partition testing to be better than random testing' *Information Processing Letters* Vol 57 No 3 (1996) pp 145–149.

[7] Chen, T Y and Yu, Y T, 'On the expected number of failures detected by subdomain testing and random testing' *IEEE Transactions on Software Engineering* Vol 22 No 2 (1996) pp 109–119.

[8] Chen, T Y and Yu, Y T, 'Constraints for safe partition testing strategies' *The Computer Journal* Vol 39 No 7 (1996) pp 619–625.

[9] Chan, F T, Chen, T Y, Mak, I K and Yu, Y T, 'Proportional sampling strategy: guidelines for software testing practitioners' *Information and Software Technology* Vol 38 No 12 (1996) pp 775–782.