To appear in *Journal of Systems and Software* **129** (1–4) (2000)

Test Case Selection With and Without Replacement [*] [†]

H. Leung [‡], T. H. Tse [§], F. T. Chan [¶], and T. Y. Chen [‖]

Abstract

Previous theoretical studies on the effectiveness of partition testing and random testing have assumed that test cases are selected with replacement. Although this assumption has been well known to be less realistic, it has still been used in previous theoretical work because it renders the analyses more tractable. This paper presents a theoretical investigation aimed at comparing the effectiveness when test cases are selected with and without replacement, and exploring the relationships between these two scenarios. We propose a new effectiveness metric for software testing, namely the expected number of distinct failures detected, to re-examine existing partition testing strategies.

1 Introduction

Random testing and subdomain testing are two commonly used strategies in software testing. In random testing, test cases are selected randomly from the entire input domain. On the other hand, a subdomain testing strategy consists of two components: the design of a partitioning scheme and the design of a test case allocation scheme. The partitioning scheme divides the input domain into subdomains, while the test case allocation scheme specifies the number of test cases chosen from each subdomain and the manner of selection. In particular, when the subdomains are mutually disjoint, subdomain testing is known as partition testing.

Traditionally, random testing has been regarded by most people as the worst strategy [1, 2, 3, 4] since it does not make use of any information from the software or its specification. Partition testing has been considered much more systematic and hence better than random testing in revealing failures. When Duran and Ntafos [5], as well as Hamlet and Taylor [6], performed simulated and empirical comparisons between partition and random testing, however, they came to a surprising conclusion that the two methods exhibit only a marginal difference in their error detection capabilities. Hence, random testing can be more cost effective than partition testing when there are significant overheads in the latter

[†] This research is supported in part by the Hong Kong Research Grants Council.

[‡] Department of Computer Science, New Mexico State University, Las Cruces, NM 88003, USA.

[§] **(Contact author)** Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong. Telephone: (+852) 2859 2183. Fax: (+852) 2559 8447. Email: thtse@cs.hku.hk .

[¶] School of Professional and Continuing Education, The University of Hong Kong, Pokfulam, Hong Kong.

[‖] School of Information Technology, Swinburne University of Technology, Hawthorn 3122, Australia. Part of the work was done when he was with the Vocational Training Council, Hong Kong.

approach. If we accept that dividing the domain into only one subdomain is a special case of partitioning, we can visualize random testing as a degenerated form of partition testing. Thus, random testing has been widely accepted as a benchmark for the effectiveness of partition testing techniques.

There have been a number of papers comparing the effectiveness between random testing and partition testing [7, 8] and among different test case allocation schemes of partition testing [9]. In all these studies, the effectiveness metrics used have been either the probability of detecting at least one failure (*P-measure*, as used in [8], for example) or the expected number of failures detected (*E-measure*, as used in [7], for example). Furthermore, the test cases have been assumed to be selected *with replacement*. As a consequence, some test cases might be duplicated. In real life, it is undesirable to have duplicated test cases, unless the cost of checking duplication is higher than the cost of test case execution. Thus, the practice of allowing test cases to be repeated is obviously less practical than the alternative practice of drawing test cases without replacement. However, the former gives rise to much simpler mathematical models, and hence facilitates the analysis of testing strategies.

This paper presents a theoretical comparison of the effectiveness between random testing and partition testing when test cases are selected with and without replacement, and a review of partition testing strategies for test case selection without replacement. Furthermore, we introduce a new testing effectiveness metric, known as the *D-measure*, which is defined as the expected number of distinct failures detected. As a note here, D-measure and E-measure are different when test cases are selected with replacement, but are equivalent when test cases are selected without replacement, Our findings will enhance the understanding of the relative effectiveness of testing strategies, and hence help software testers choose the appropriate schemes.

In the next section, we take a look at previous studies on the effectiveness of test case selection with and without replacement. In Section 3, we analyze the probability of detecting at least one failure when test cases are selected without replacement, and review the existing testing strategies with this new assumption. Section 4 defines an effectiveness metric that is more intuitively appealing than the E-measure. Section 5 provides some guidelines to software testers for choosing appropriate test case selection strategies. Section 6 is the conclusion.

2 Previous Work

First, we present the basic notation and concepts here before summarizing previous studies by various authors on the effectiveness of test case selection with and without replacement. Let $\mathcal{D}$ denote the input domain of a program. The elements of $\mathcal{D}$ that produce incorrect outputs, and hence reveal program errors, are called *failure-causing inputs*. Let the variables d, m, and n denote the size of the input domain, the 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. It should be noted that, while the sizes and locations of failure-causing inputs are not known to software testers before the testing process, they are nevertheless fixed for given programs.

For any partition testing strategy P, the k disjoint subdomains formed are denoted by $\mathcal{D}_i$, $i = 1, 2, \ldots, k$, where $k \geq 2$. For any subdomain $\mathcal{D}_i$, let d_i, m_i, and n_i denote its size, the size of the 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.

For clarity, previous work for situations where test cases are selected with and without replacement will be presented separately.

2.1 Test Case Selection With Replacement

In this section, we review the previous work by various authors on situations where test cases are randomly selected based on a uniform distribution, independent of one another and with replacement. For random testing, the probability of detecting at least one failure (P-measure), denoted by $P_R(n)$, is equal to $1 - (1-\theta)^n$, and the expected number of failures detected (E-measure), denoted by $E_R(n)$, is equal to $n\theta$. For partition testing, the P-measure, denoted by $P_P(n_1, n_2, \ldots, n_k)$, is equal to $1 - \prod_{i=1}^{k}(1-\theta_i)^{n_i}$, while the E-measure, denoted by $E_P(n_1, n_2, \ldots, n_k)$, is equal to $\sum_{i=1}^{k} n_i\theta_i$.

Suppose n_r test cases are selected randomly with replacement. Let d be the size of the input domain. The expected number of distinct test cases n_d is $d\left[1 - \left(1-\frac{1}{d}\right)^{n_r}\right]$. The following corollary follows immediately.

Corollary 1
Suppose test cases are selected randomly with replacement from an input domain of size d. In order to expect n_d of these test cases to be distinct, where $0 \le n_d \le d-1$, we should select $n_r = \frac{\log\left(1-\frac{n_d}{d}\right)}{\log\left(1-\frac{1}{d}\right)}$ test cases with replacement.

2.1.1 Proportional Sampling Strategy

In their analytical study of partition testing, Weyuker and Jeng [8] proved that if $d_1 = d_2 = \cdots = d_k$ and $n_1 = n_2 = \cdots = n_k$, then $P_P(n_1, n_2, \ldots, n_k) \ge P_R(n)$. This was the first sufficient condition derived for $P_P(n_1, n_2, \ldots, n_k) \ge P_R(n)$.

Chen and Yu [10] generalized Weyuker and Jeng's result and proposed the proportional sampling strategy. They proved that if $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, then $P_P(n_1, n_2, \ldots, n_k) \ge P_R(n)$. Since $\sigma_i = \sigma_j$ implies $\frac{n_i}{n_j} = \frac{d_i}{d_j}$, it is not necessary to know the absolute sizes of the subdomains in order to apply the proportional sampling strategy. Only the relative sizes of the subdomains are required to determine the values of n_i's. For example, if $\frac{d_1}{d_2} = \frac{1}{2}$, the proportional sampling strategy simply recommends that test cases should be allocated to these two subdomains in a 1 to 2 ratio. As suggested by Chan *et al.* [11], a handy technique to estimate the relative sizes of the subdomains is the Monte Carlo method [12]. The proportional sampling strategy is more useful in practice than that of Weyuker and Jeng, since the latter requires all subdomains to be of equal sizes.

Similarly, Chen and Yu [7] proved that the proportional sampling strategy also ensures that the E-measure of partition testing is just as good as that of random testing. In other words, if $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, then $E_p(n_1, n_2, \ldots, n_k) = E_r(n)$.

Since a strict application of the proportional sampling strategy may cause practical problems, such as a non-integral number of test cases, Chan *et al.* [11] provided some guidelines to handle this kind of problems.

2.1.2 Optimally Refined Proportional Sampling Strategy

Based on a refinement of the proportional sampling strategy, Chan *et al.* [11] proposed an optimally refined proportional sampling (ORPS) strategy, which divides the input domain into as many equal-sized subdomains as the number of test cases, and selects one test case randomly from each subdomain.

It should be noted that the ORPS strategy is also a special case of Weyuker and Jeng's equal-sized and equal-numbered strategy.

When test cases are selected with replacement, the ORPS strategy has a higher probability of detecting at least one failure than random testing. However, the partitioning overheads may offset the improvement in revealing failures. Hence, Chan *et al.* recommended that, when it is fairly easy to divide the input domain into subdomains of equal sizes, the ORPS strategy is preferred to random testing. Their empirical study on a sample of published programs showed that the P-measure of the ORPS strategy was about 7.5% higher than that of random testing.

2.1.3 Follow-the-Crowd Strategies

Without loss of generality, assume that $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$. Chen and Yu [7] proved that if $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, then $P_P(n_1,\ n_2,\ \ldots,\ n_k) \geq P_R(n)$, and that if $(\theta_i - \theta)(\sigma_i - \sigma) \geq 0$ for all $i = 1,\ 2,\ \ldots,\ k$, then $P_P(n_1,\ n_2,\ \ldots,\ n_k) \geq P_R(n)$. Obviously, when $\sigma_1 = \sigma_2 = \cdots = \sigma_k$, the above two conditions are satisfied. Hence, the proportional sampling strategy is a special case of these two strategies.

Assuming that $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$, they found similar conditions for the E-measure, namely that if $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, then $E_P(n_1,\ n_2,\ \ldots,\ n_k) \geq E_R(n)$, and if $(\theta_i - \theta)(\sigma_i - \sigma) \geq 0$ for all $i = 1,\ 2,\ \ldots,\ k$, then $E_P(n_1,\ n_2,\ \ldots,\ n_k) \geq E_R(n)$. The reverse is also true. Thus, if $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_k$, then $E_P(n_1,\ n_2,\ \ldots,\ n_k) \leq E_R(n)$, and if $(\theta_i - \theta)(\sigma_i - \sigma) \leq 0$ for all $i = 1,\ 2,\ \ldots,\ k$, then $E_P(n_1,\ n_2,\ \ldots,\ n_k) \leq E_R(n)$.

The merit of these strategies is that they can be applied once the relative ordering of the failure rates in respective subdomains is known. There are various ways to estimate the relative ordering of the failure rates. For example, if the input domain is partitioned according to the specified functions, a subdomain associated with a more complex function may be estimated to have a higher failure rate.

No name was given to these strategies. For the ease of discussion in this paper, we shall refer to them as the follow-the-crowd strategies because, intuitively speaking, they recommend more test cases where the failure rate is higher.

2.1.4 Partial Sums Condition

Consider two partition testing strategies with the same partitioning scheme but different test case allocation schemes. Software testers would be interested in knowing how one partition testing strategy compares with another. Chan *et al.* [9] proved a partial sums condition for this purpose. Suppose the subdomains are labeled in decreasing order of the failure rates, so that $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$. The partial sums of a partitioning strategy A is defined as $\sum_{i=1}^{r} n_{Ai}$ for $r = 1,\ 2,\ \ldots,\ k-1$, where n_{Ai} is the number of test cases to be selected from subdomain $\mathcal{D}_i$. For two partitioning strategies A and B with the same partitioning scheme, they proved that if $\sum_{i=1}^{r} n_{Ai} \geq \sum_{i=1}^{r} n_{Bi}$ for all $r = 1,\ 2,\ \ldots,\ k-1$, then $P_P(n_{A1},\ n_{A2},\ \ldots,\ n_{Ak}) \geq P_P(n_{B1},\ n_{B2},\ \ldots,\ n_{Bk})$ and $E_P(n_{A1},\ n_{A2},\ \ldots,\ n_{Ak}) \geq E_P(n_{B1},\ n_{B2},\ \ldots,\ n_{Bk})$. In other words, if every partial sum of Strategy A is greater than or equal to the corresponding one in Strategy B, then Strategy A is better than Strategy B in terms of both the P-measure and E-measure.

2.2 Test Case Selection Without Replacement

There are a number of drawbacks if test cases are selected with replacement. Since test cases may be repeated, testing resources may then be wasted. Furthermore, even if the number of selected test cases exceeds the size of the input domain, there is no guarantee that all the failures will be revealed, because some failure-causing inputs may be missed while some other inputs may have been repeatedly selected as test cases.

Let us consider situations where test cases are selected without replacement. For random testing, the probability of detecting at least one failure (P-measure) is equal to

$$1 - \left(\frac{d-m}{d}\right)\left(\frac{d-m-1}{d-1}\right)\cdots\left(\frac{d-m-n+1}{d-n+1}\right)$$

or

$$1 - \frac{(d-m)!(d-n)!}{d!(d-m-n)!}$$

when $d > m+n$, and 1 otherwise. The expected number of failures detected (E-measure) is still equal to $n\theta$. It is because, in the derivation of this formula, we need not consider whether the failures are distinct.

For partition testing when test cases are selected with replacement, the P-measure is equal to $1 - \prod_{i=1}^{k} \frac{(d_i-m_i)!(d_i-n_i)!}{d_i!(d_i-m_i-n_i)!}$, and the E-measure is equal to $\sum_{i=1}^{k} n_i\theta_i$.

3 Probability of Detecting at Least One Failure when Test Cases are Selected Without Replacement

We note from the last section that the result of the P-measure when test cases are selected with replacement is different from that when test cases are selected without replacement In order to differentiate between them, the latter will be called the *Q-measure* in this paper.

Definition 1 (Q-Measure)
The Q-measure is defined as the probability of detecting at least one failure when test cases are selected without replacement.

The Q-measure for random testing will be denoted by $Q_R(n)$ and that for partition testing by $Q_P(n_1, n_2, \ldots, n_k)$.

The Q-measure better reflects the real life situation of software testing, since testers would prefer not to repeat test cases whenever possible. However, an analytical study of the Q-measure would be very complex. We shall propose in this section how the mathematical complexity can be alleviated by linking up the Q-measure when test cases are selected without replacement with the P-measure when test cases are selected with replacement, thus paving the way for the analysis of various testing strategies.

Before we do so, let us reiterate some of our major observations from the previous section:

(a) Given an integer n_r, n_d may not be an integer.

(b) Given an integer n_d, n_r may not be an integer.

(c) $Q_R(n)$ and $P_R(n)$ are the probabilities of detecting at least one failure only when n is an integer.

(d) The formula for $P_R(n)$ is still well-defined and in fact continuous when n is real. However, this does not apply to $Q_R(n)$.

In view of these points, we shall keep all the n_d's as integers in our analysis. The corresponding n_r's are computed using the formula in Corollary 1. We extend the domain of the function $P_R(n) = 1-(1-\theta)^n$ from positive integers to positive real numbers.

3.1 $P_R(n_r)$ as a Lower Bound of $Q_R(n_d)$

We have pointed out that the Q-measure is very complex for useful mathematical analysis and hence difficult to be applied in practice. It is obvious, however, that $Q_R(n) \geq P_R(n)$ because some of the test cases may be duplicated when they are selected with replacement. Given n_d test cases selected without replacement from an input domain of size d, and n_r as computed from the formula in Corollary 1, we are interested in the relationship between $Q_R(n_d)$ and $P_R(n_r)$. We find that $P_R(n_r)$ is a lower bound of $Q_R(n_d)$. In order to prove this property, we need the following lemmas:

Lemma 1
Consider an input domain of size d with m failure-causing instances. Let n_d be the number of test cases selected randomly without replacement and $n_r = \dfrac{\log\left(1-\frac{n_d}{d}\right)}{\log\left(1-\frac{1}{d}\right)}$.

$$\begin{aligned} P_R(n_r) &= 1 - \left(1-\frac{m}{d}\right)^{n_r} \\ &= 1 - \left(1-\frac{n_d}{d}\right)^{m'}, \text{ where } m' = \frac{\log\left(1-\frac{m}{d}\right)}{\log\left(1-\frac{1}{d}\right)} \end{aligned}$$

Proof

$$\begin{aligned} \log\left[\left(1-\frac{n_d}{d}\right)^{m'}\right] &= m' \log\left(1-\frac{n_d}{d}\right) \\ &= \frac{\log\left(1-\frac{m}{d}\right)}{\log\left(1-\frac{1}{d}\right)} \log\left(1-\frac{n_d}{d}\right) \\ &= \frac{\log\left(1-\frac{n_d}{d}\right)}{\log\left(1-\frac{1}{d}\right)} \log\left(1-\frac{m}{d}\right) \\ &= n_r \log\left(1-\frac{m}{d}\right) \\ &= \log\left[\left(1-\frac{m}{d}\right)^{n_r}\right] \end{aligned}$$

Therefore, $\left(1-\frac{m}{d}\right)^{n_r} = \left(1-\frac{n_d}{d}\right)^{m'}$ and the proof is done. ■

Corollary 2
If $m = 1$, then $P_R(n_r) = \dfrac{n_d}{d} = Q_R(n_d)$.

Proof
When $m = 1$, we have $m' = 1$. Hence, it follows from Lemma 1 that $P_R(n_r) = 1 - \left(1-\frac{n_d}{d}\right) = \frac{n_d}{d}$. On the other hand,

$$
\begin{aligned}
Q_R(n_d) &= 1 - \left(\frac{d-m}{d}\right)\left(\frac{d-m-1}{d-1}\right)\cdots\left(\frac{d-m-n_d+1}{d-n_d+1}\right) \\
&= 1 - \left(\frac{d-1}{d}\right)\left(\frac{d-2}{d-1}\right)\cdots\left(\frac{d-n_d}{d-n_d+1}\right) \\
&= 1 - \frac{d-n_d}{d} \\
&= \frac{n_d}{d} \\
&= P_R(n_r)
\end{aligned}
$$

■

Lemma 2
Let $f(x) = x\,(x+1)\,\cdots\,(x+k)$ and $g(x) = (x+k)^{\alpha(k+1)}$, where $k > 1$ is an integer and $\alpha > 1$. Let $C > 0$. Then $Cf(x)$ and $g(x)$ do not meet at more than two points for $x \geq 0$.

Proof
Let $h(x) = \dfrac{f(x)}{g(x)}$. Note that both $h(x)$ and $h'(x) = \dfrac{\mathrm{d}}{\mathrm{d}x}h(x)$ are continuous and differentiable for $x > 0$. The lemma is proved if we can show that for $x > 0$, $h'(x) = 0$ occurs at exactly one point. Since $h'(x) = \dfrac{\mathrm{d}}{\mathrm{d}x}\dfrac{f(x)}{g(x)} = \dfrac{f'(x)g(x) - g'(x)f(x)}{g(x)^2}$, it is equivalent to argue that for $x > 0$, $\dfrac{f'(x)}{f(x)} = \dfrac{g'(x)}{g(x)}$ occurs at exactly one point. Next $\dfrac{f'(x)}{f(x)} = \dfrac{1}{x} + \dfrac{1}{x+1} + \cdots + \dfrac{1}{x+k}$ and $\dfrac{g'(x)}{g(x)} = \alpha\dfrac{k+1}{x+k}$. Let

$$
\begin{aligned}
p(x) &= \left(\frac{1}{x} + \frac{1}{x+1} + \cdots + \frac{1}{x+k}\right) \Big/ \frac{k+1}{x+k} \\
&= \left(\frac{1}{k+1}\right)\left(\frac{x+k}{x} + \frac{x+k}{x+1} + \cdots + \frac{x+k}{x+k-1} + \frac{x+k}{x+k}\right) \\
&= \left(\frac{1}{k+1}\right)\left(k+1 + \frac{k}{x} + \frac{k-1}{x+1} + \frac{k-2}{x+2} + \cdots + \frac{1}{x+k-1}\right) \\
&= 1 + \left(\frac{1}{k+1}\right)\left(\frac{k}{x} + \frac{k-1}{x+1} + \frac{k-2}{x+2} + \cdots + \frac{1}{x+k-1}\right)
\end{aligned}
$$

It is easy to see that, for $x > 0$, $p(x)$ is monotonically decreasing as x increases. Moreover, $\lim_{x\to 0} p(x) = \infty$ and $\lim_{x\to\infty} p(x) = 1$. Hence, $p(x)$ is a bijection from $(0,\ \infty)$ to $(1,\ \infty)$. Thus, $p(x) = \alpha$ occurs at exactly one point where $\alpha > 1$. ■

Lemma 3
For any $k = 0,\ 1,\ \ldots,\ d-1$, if $k' = \dfrac{\log\left(1 - \frac{k}{d}\right)}{\log\left(1 - \frac{1}{d}\right)}$, then $k' \geq k$.

Proof
From

$$
\log\left(1 - \frac{1}{d}\right)^{k'} = k'\,\log\left(1 - \frac{1}{d}\right) = \log\left(1 - \frac{k}{d}\right)
$$

we obtain

$$\left(1-\frac{1}{d}\right)^{k'} = 1-\frac{k}{d}$$

Hence,

$$k = d \ - \ d \ \left(1-\frac{1}{d}\right)^{k'}$$

Suppose k' is an integer. Then k can be considered to be that the expected number of distinct test cases obtained when k' random test cases are selected with replacement based on a uniform distribution from a domain of size d. Thus, $k' \geq k$. Suppose k' is not an integer. Let $k_1 = d \ - \ d \ \left(1-\frac{1}{d}\right)^{\lceil k' \rceil}$. Then $k < k_1 \leq \lceil k' \rceil$. Since k is an integer but k' is not an integer, we deduce that $k < k'$. ■

We are now ready to prove that $P_R(n_r)$ is a lower bound of $Q_R(n_d)$.

Theorem 1

Consider an input domain of size d with m failure-causing instances. Suppose n_d distinct test cases are selected randomly, where n_d is a non-negative integer smaller than d. Let $Q_R(n_d)$ be the probability of detecting at least one failure. If $n_r = \frac{\log(1-\frac{n_d}{d})}{\log(1-\frac{1}{d})}$ and $P_R(n_r) = 1 \ - \ \left(1-\frac{m}{d}\right)^{n_r}$, then $Q_R(n_d) \geq P_R(n_r)$.

Proof

When $m = 0$, $Q_R(n_d) = 0 = P_R(n_r)$. When $m = 1$, $Q_R(n_d) = \frac{n_d}{d} = P_R(n_r)$. When $m = d$, $Q_R(n_d) = 1 = P_R(n_r)$. Therefore, we need only show that

$$1 \ - \ \left(\frac{d-m}{d}\right) \ \left(\frac{d-m-1}{d-1}\right) \ \cdots \ \left(\frac{d-m-n_d+1}{d-n_d+1}\right) \ \geq \ 1 \ - \ \left(1-\frac{m}{d}\right)^{n_r}$$

for $n_d = 0, \ 1, \ \ldots, \ d-1$ and $m = 2, \ 3, \ \ldots, \ d-1$. By Lemma 1, if $m' = \frac{\log\left(1-\frac{m}{d}\right)}{\log\left(1-\frac{1}{d}\right)}$, the problem can be transformed into

$$\left(\frac{d-m}{d}\right) \ \left(\frac{d-m-1}{d-1}\right) \ \cdots \ \left(\frac{d-m-n_d+1}{d-n_d+1}\right) \ \leq \ \left(1-\frac{m}{d}\right)^{n_r} \ = \ \left(1-\frac{n_d}{d}\right)^{m'}$$

for $n_d = 0, \ 1, \ \ldots, \ d-1$ and $m = 2, \ 3, \ \ldots, \ d-1$. Since

$$\begin{aligned}
&\left(\frac{d-m}{d}\right) \ \left(\frac{d-m-1}{d-1}\right) \ \cdots \ \left(\frac{d-m-n_d+1}{d-n_d+1}\right) \\
&= \ \frac{(d-m)!(d-n_d)!}{d!(d-m-n_d)!} \\
&= \ \frac{1}{d \ (d-1) \ \cdots \ (d-m+1)} \ (d-n_d) \ (d-n_d-1) \cdots (d-m-n_d+1)
\end{aligned}$$

and by letting $x = d-m-n_d+1$, it will be equivalent to show that

$$\frac{d^{m'}}{d \ (d-1) \ \cdots \ (d-m+1)} \ (x) \ (x+1) \ \cdots \ (x+m-1) \ \leq \ (x+m-1)^{m'}$$

for $x = -m+2,\ -m+3,\ \ldots,\ d-m+1$. Next, we observe that

$$\frac{d^{m'}}{d\ (d-1)\ \cdots\ (d-m+1)}\ (x)\ (x+1)\ \cdots\ (x+m-1)\ =\ 0$$

and $(x+m-1)^{m'} > 0$ for $x = -m+2,\ -m+3,\ \ldots,\ 0$. Thus, the proof is done if we can show that

$$\frac{d^{m'}}{d\ (d-1)\ \cdots\ (d-m+1)}\ (x)\ (x+1)\ \cdots\ (x+m-1)\ \leq\ (x+m-1)^{m'}$$

for all real values of x in the range $(0, d-m+1]$.

Let $C = \dfrac{d^{m'}}{d\ (d-1)\ \cdots\ (d-m+1)}$, $f(x) = x\ (x+1)\ \cdots\ (x+m-1)$ and $g(x) = (x+m-1)^{m'}$. Since $m' > m$ and $C > 0$, by Lemma 2, $Cf(x)$ and $g(x)$ do not meet at more than two points for $x \geq 0$. When $x = d-m$,

$$\begin{aligned}
Cf(x) &= \frac{d^{m'}}{d\ (d-1)\ \cdots\ (d-m+1)}\ (x)\ (x+1)\ \cdots\ (x+m-1) \\
&= d^{m'}\frac{d-m}{d} \\
&= \left(1-\frac{m}{d}\right)\ d^{m'} \\
&= \left(1-\frac{m}{d}\right)\ d^{\log(1-m/d)/\log(1\ -\ 1/d)} \\
&= \left(1-\frac{m}{d}\right)\ \left(1-\frac{m}{d}\right)^{\log(d)/\log(1\ -\ 1/d)} \\
&= \left(1-\frac{m}{d}\right)^{1+\log(d)/\log(1\ -\ 1/d)} \\
&= \left(1-\frac{m}{d}\right)^{(\log(1\ -\ 1/d)+\log(d))/\log(1\ -\ 1/d)} \\
&= \left(1-\frac{m}{d}\right)^{\log(d-1)/\log(1\ -\ 1/d)} \\
&= (d-1)^{\log(1-m/d)/\log(1\ -\ 1/d)} \\
&= (x+m-1)^{m'} \\
&= g(x)
\end{aligned}$$

When $x = d-m+1$,

$$\begin{aligned}
Cf(x) &= \frac{d^{m'}}{d\ (d-1)\ \cdots\ (d-m+1)}\ (x)\ (x+1)\ \cdots\ (x+m-1) \\
&= d^{m'} \\
&= (x+m-1)^{m'} \\
&= g(x)
\end{aligned}$$

That is, $Cf(x) = g(x)$ for $x = d-m$ and $x = d-m+1$. By the fact that $Cf(0) = 0 < g(0)$ since $m \geq 2$, we deduce that $Cf(x) < g(x)$ for $0 \leq x < d-m$. Together with the fact that $Cf(x) = g(x)$ when $x = d-m$ and $x = d-m+1$, the proof is done. ■

3.2 $P_R(n_r)$ as an Approximation for $Q_R(n_d)$

Theorem 1 also establishes the relationships $Q_R(n_d) \geq P_R(n_r) \geq P_R(n_d)$ as $n_r \geq n_d$. Obviously, $P_R(n_r)$ gives a better lower bound for $Q_R(n_d)$ than $P_R(n_d)$. The theorem reveals that, even after converting n_d to the corresponding n_r, the effectiveness of n_d test cases selected without replacement is still higher than that using n_r test cases selected with replacement. This means that test case selection without replacement outperforms that with replacement not just because the test cases are distinct.

A closer examination of test case selection without replacement reveals that it is a more effective approach to *hit* the failure-causing inputs than test case selection with replacement. When earlier selections of test cases do not hit any of the failure-causing inputs, the selected data are discarded from the selection set, and hence the failure rate will increase in subsequent selections. For example, if $d = 10$ and $m = 5$, the converging approach of test case selection without replacement guarantees that at least one failure-causing input will be selected when $n_d = 6$. There is no such guarantee for any value of n_r when test cases are selected with replacement.

We are interested in knowing how close the values of $Q_R(n_d)$ and $P_R(n_r)$ are. One way is to look at the ratio $\dfrac{P_R(n_r)}{Q_R(n_d)}$. For any given d, we define

$$closeness(d) = \min_{m,\ n_d} \frac{P_R(n_r)}{Q_R(n_d)} \text{ for all possible values of } m \text{ and } n_d$$

It provides a conservative estimate on how close the two values are to each other. Since $Q_R(n_d) \geq P_R(n_r)$, $closeness(d)$ should never exceed 1. For a given d, a $closeness(d)$ of 1 implies that the two values are always equal. From the proof of Theorem 1, we have seen that $Q_R(n_d) = P_R(n_r)$ for $m = 0$, $m = 1$, and $m = d$. Therefore, $closeness(d) = 1$ for $d = 1$ and $d = 2$.

Through simulations, we find that $closeness(d)$ is monotonically increasing for $d = 3$ to 300.

$$\begin{aligned}
&closeness(d) \geq 0.949 \text{ for } d \geq 3\\
&closeness(d) \geq 0.980 \text{ for } d \geq 7\\
&closeness(d) \geq 0.990 \text{ for } d \geq 14\\
&closeness(d) \geq 0.995 \text{ for } d \geq 29\\
&closeness(d) \geq 0.997 \text{ for } d \geq 49\\
&closeness(d) \geq 0.999 \text{ for } d \geq 153
\end{aligned}$$

A further checking on $d = 1000$ shows that $closeness(d) \geq 0.99984$. From these empirical observations, we propose the following conjecture.

Conjecture 1
Consider an input domain of size d with m failure-causing instances. Suppose n_d distinct test cases are selected randomly, where n_d is a non-negative integer smaller than d. Let $Q_R(n_d)$ be the probability of detecting at least one failure. If $n_r = \dfrac{\log(1-\frac{n_d}{d})}{\log(1-\frac{1}{d})}$, then $P_R(n_r) = 1 - \left(1-\dfrac{m}{d}\right)^{n_r}$ is a close approximation of $Q_R(n_d)$ when d is sufficiently large.

We note that the value of n_r expressed in Conjecture 1 may not always be an integer, and hence $1 - \left(1-\dfrac{m}{d}\right)^{n_r}$ cannot always be interpreted as the probability of detecting at least one failure when

n_r test cases are selected with replacement. Since $Q_R(n_d)$ is difficult to evaluate, we use the generalized function $P_R(x) = 1 - (1-\theta)^x$ as its approximation and substitute x by n_r, which is the expected number of test cases selected randomly with replacement in order to find n_d of them to be distinct.

One may ask whether rounding down or rounding up n_r will be a better way to handle the case when n_r does not take an integral value. We repeated similar simulations by rounding down n_r to an integral value and used $P_R(\lfloor n_r \rfloor)$ instead of $P_R(n_r)$. As expected, $P_R(\lfloor n_r \rfloor)$ is not as good an approximation as $P_R(n_r)$. The values of *closeness* are not approaching 1 as fast. For example, *closeness*(200) using $P_R(\lfloor n_r \rfloor)$ is 0.9803619, while *closeness*(200) using $P_R(n_r)$ is 0.9992303. On the other hand, we have also tried rounding up n_r, but found that $P_R(\lceil n_r \rceil)$ is not a good approximation for $Q_R(n_d)$ either. Its values may be greater than $Q_R(n_d)$. For example, when $d = 10$, $m = 2$, and $n_d = 2$, $\frac{P_R(\lceil n_r \rceil)}{Q_R(n_d)} = 1.2918$ whereas $\frac{P_R(n_r)}{Q_R(n_d)} = 0.9969$. Hence, $P_R(\lceil n_r \rceil)$ is not a lower bound of $Q_R(n_d)$. The relative difference between $P_R(\lceil n_r \rceil)$ and the corresponding $Q_R(n_d)$ can sometimes be quite large when compared with the case when $P_R(n_r)$ is used.

If we substitute $Q_R(n_d)$ by $P_R(n_r)$ as a close approximation, we have a much simpler mathematical model for the analysis of the effectiveness of partition testing when test cases are selected without replacement. An important implication for software testers is that the above conjecture provides an effectiveness conversion formula for comparing test case selection with and without replacement. Roughly speaking, $(n_r - n_d) \times$ *cost of executing a test case* gives an estimate of the execution cost of test case selection with replacement over that without replacement. Software testers can compare this difference in cost with the additional overheads required to screen out duplications in test cases. They can then decide whether test cases should be selected with or without replacement.

Table 1 may provide some idea on the relation between n_r and n_d for various domain sizes. Based on these observations, when n_d is approximately one tenth of d, n_r exceeds n_d by about 5%. When n_d is around 20% of d, n_r exceeds n_d by about 11%. When n_d is half of d, n_r is more than 38% of n_d. The closer n_d is to d, the greater n_r exceeds n_d. When n_d approaches d, n_r approaches ∞. From another point of view, we find that when n_r is approximately one tenth of d, n_d is more than 95% of n_r. When n_r is around half of d, n_d is around 80% of n_r. When n_r is equal to d, n_d is around 63% of n_r.

3.3 A Review of Existing Strategies in the Context of Test Case Selection Without Replacement

Previous studies on the P-measure of partition testing centered on test cases with replacement. Since the assumptions have been changed when test cases are selected without replacement, and since the Q-measure is different from the P-measure, we would like to re-examine some of the previous findings and see whether they are still valid under the new assumptions.

Leung and Chen [13] found that the proportional sampling strategy ($\sigma_1 = \sigma_2 = \cdots = \sigma_k$) does not always guarantee $Q_P(n_{1_d}, n_{2_d}, \ldots, n_{k_d}) \geq Q_R(n_d)$ when test cases are selected without replacement. We use the following example to illustrate this.

Example 1
Consider $d = 1000$, $k = 2$, $d_1 = 400$, $d_2 = 600$, $m_1 = 2$, $m_2 = 2$, and $n = 10$. The proportional sampling strategy imposes that $n_{1_d} = 4$ and $n_{2_d} = 6$. We would then have $Q_R(10) = 0.394623 > 0.394445 = Q_P(4, 6)$. ■

$d =$	n_d	2	3	4	5	8	9			
10	n_r	2.12	3.39	4.85	6.58	15.28	21.85			
$d =$	n_d	2	5	10	20	25	30	40	49	
50	n_r	2.02	5.22	11.05	25.29	34.31	45.35	79.66	193.64	
$d =$	n_d	2	5	10	20	40	50	60	80	99
100	n_r	2.01	5.10	10.48	22.20	50.83	68.97	91.17	160.14	458.21
$d =$	n_d	5	10	100	200	300	500	700	900	999
1000	n_r	5.01	10.05	105.31	223.03	356.45	692.80	1203.37	2301.43	6904.30
$d =$	n_d	50	100	500	1000	2000	2500	3000	4000	4999
5000	n_r	50.25	101.00	526.75	1115.61	2553.87	3465.39	4581.00	8046.38	42581.7

Table 1

Since the proportional sampling strategy is a special case of the follow-the-crowd strategies, we can also conclude that the follow-the-crowd strategies do not guarantee $Q_P(n_{1_d},\ n_{2_d},\ \ldots,\ n_{k_d}) \geq Q_R(n_d)$ when test cases are selected without replacement.

When test cases are selected with replacement, the ORPS strategy guarantees a higher probability of finding at least one failure than random testing. When test cases are selected without replacement, however, this guarantee is no longer valid. Again, let us use an example to illustrate this.

Example 2
Consider $d = 10$ and $n = 2$. Applying the optimally refined proportional sampling strategy, we have $k = 2$, $d_1 = d_2 = 5$, and $n_1 = n_2 = 1$. If $m_1 = 1$ and $m_2 = 1$, then $Q_R(2) = 0.377778$ and $Q_P(1,\ 1) = 0.360000$. ■

Consider two partition testing strategies A and B with the same partitioning scheme but different test case allocation schemes such that the test cases are selected without replacement. The partial sums condition no longer guarantees that one strategy has a better Q-measure than the other. In other words, given $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ and $\sum_{i=1}^{k} n_{Ai} = \sum_{i=1}^{k} n_{Bi}$, the condition $\sum_{i=1}^{r} n_{Ai} \geq \sum_{i=1}^{r} n_{Bi}$ for all $r = 1,\ 2,\ \ldots,\ k-1$ does not guarantee that $Q_P(n_{A1},\ n_{A2},\ \ldots,\ n_{Ak}) \geq Q_P(n_{B1},\ n_{B2},\ \ldots,\ n_{Bk})$. The following example substantiates our claim.

Example 3
Suppose $k = 2$, $d_1 = 100$, $m_1 = 10$, $d_2 = 100$, and $m_2 = 9$. In Strategy A, take $n_{A1} = 10$ and $n_{A2} = 15$. In Strategy B, take $n_{B1} = 5$ and $n_{B2} = 20$. Then

$$\begin{aligned} Q_P(10,\ 15) &= 0.928469 \\ Q_P(5,\ 20) &= 0.928835 \geq Q_P(10,\ 15) \end{aligned}$$

■

4 Expected Number of Distinct Failures Detected

4.1 The D-Measure

The expected number of failures detected (E-measure) is an effectiveness metric aimed at measuring how many failures a test suite is capable of detecting. $E_R(n_r)$ is not a true reflection of this capability when test cases are selected with replacement. It is because the same failure-causing input may be selected as test cases more than once, and hence may be counted repeatedly towards the E-measure. The following example illustrates such a weakness.

Example 4
Consider $d_1 = 2$, $m_1 = 1$, $d_2 = 10$, and $m_2 = 4$.
In Strategy A, take $n_{1_r} = 2$ and $n_{2_r} = 10$. We have E-measure $= 5$.
In Strategy B, take $n_{1_r} = 12$ and $n_{2_r} = 0$. We have E-measure $= 6$. ■

In this example, the five failure-causing inputs may be related to different program errors. Based on the E-measure, Strategy B above gives the false impression that it is a better test case allocation scheme than Strategy A. However, it is obvious that the value of its E-measure, 6, comes from the same single failure-causing input in $\mathcal{D}_1$. On the other hand, for Strategy A, the expected number of failures detected, 5, is related to all five failure-causing inputs. Since Strategy A can detect more distinct failures, it is in fact the better choice.

A more practical effectiveness measure of software testing is now defined:

Definition 2 (D-Measure)
The D-measure is defined as the expected number of distinct failures detected.

When test cases are selected without replacement, for random testing, $D_R(n_d) = E_R(n_d) = n_d\theta$, while for partition testing, $D_P(n_{1_d},\ n_{2_d},\ \ldots,\ n_{k_d}) = E_P(n_{1_d},\ n_{2_d},\ \ldots,\ n_{k_d}) = \sum_{i=1}^{k} n_{i_d}\theta_i$. When test cases are selected with replacement, however, we have

$$\begin{aligned} D_R(n_r) &= d\theta\left[1 - \left(1 - \frac{1}{d}\right)^{n_r}\right] \\ &= m\left[1 - \left(1 - \frac{1}{d}\right)^{n_r}\right] \end{aligned}$$

and

$$\begin{aligned} D_P(n_{1_d},\ n_{2_d},\ \ldots,\ n_{k_d}) &= \sum_{i=1}^{k} d_i\theta_i\left[1 - \left(1 - \frac{1}{d_i}\right)^{n_{ir}}\right] \\ &= \sum_{i=1}^{k} m_i\left[1 - \left(1 - \frac{1}{d_i}\right)^{n_{ir}}\right] \end{aligned}$$

The D-measure and E-measure are identical when test cases are selected without replacement. We shall, therefore, confine our discussion below to the situation when test cases are selected with replacement, where the D-measure and E-measure are different. Let us return to the previous example. The D-measure for Strategy A is 3.3553 and that for Strategy B is 0.99975. It can be seen that the D-measure is a better indicator of testing effectiveness than the E-measure.

The D-measure and P-measure, however, represent two different effectiveness metrics for software testing. When one test case allocation scheme has a higher P-measure than another scheme, it does not necessarily mean that the former scheme has a higher D-measure also. The following example illustrates such a situation. Hence, the final decision on the test case allocation scheme should depend on which effectiveness measure is considered more important by the software tester.

Example 5
Consider two test case allocation strategies such that test cases are selected with replacement.

In Strategy A, take $d_1 = 1000$, $\theta_1 = 0.4$, $n_{1_r} = 5$, $d_2 = 10$, $\theta_2 = 0.6$, and $n_{2_r} = 5$. We have P-measure $= 0.999204$ and D-measure $= 4.453064$.

In Strategy B, take $d_1 = 1000$, $\theta_1 = 0.4$, $n_{1_r} = 0$, $d_2 = 10$, $\theta_2 = 0.6$, and $n_{2_r} = 10$. We have P-measure $= 0.999895$ and D-measure $= 3.907929$.

Strategy A is better than Strategy B with respect to the D-measure, but Strategy B is better with respect to the P-measure. ■

Since the D-measure is not monotonically increasing with the E-measure or P-measure, we would like to re-examine the existing partition testing strategies in the light of the D-measure and see how far they are still applicable to this new effectiveness metric.

4.2 A Review of Existing Strategies in the Light of the D-measure

4.2.1 Proportional Sampling Strategy

The proportional sampling strategy was proved to be a sufficient condition for $P_P(n_1,\ n_2,\ \ldots,\ n_k) \geq P_R(n_d)$ and $E_P(n_1,\ n_2,\ \ldots,\ n_k) = E_R(n_d)$ for test case selection with replacement. Coincidentally, we find that the proportional sampling strategy also favors partition testing over random testing with respect to the D-measure.

Theorem 2
Let $D_R(n_r)$ be the D-Measure for random testing with n_r test cases selected with replacement, and $D_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r})$ be the corresponding measure for partition testing, where $\sum_{i=1}^{k} n_{i_r} = n_r$. Under the proportional sampling strategy, $D_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) > D_R(n_r)$.

Proof

$$\begin{aligned} D_R(n_r) &= m\left[1 - \left(1-\frac{1}{d}\right)^{n_r}\right] \\ &= m - m\left(1-\frac{1}{d}\right)^{n_r} \end{aligned}$$

$$\begin{aligned} D_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) &= \sum_{i=1}^{k} m_i\left[1 - \left(1-\frac{1}{d_i}\right)^{n_{i_r}}\right] \\ &= \sum_{i=1}^{k} m_i - \sum_{i=1}^{k} m_i\left(1-\frac{1}{d_i}\right)^{n_{i_r}} \\ &= m - \sum_{i=1}^{k} m_i\left(1-\frac{1}{d_i}\right)^{n_{i_r}} \end{aligned}$$

where $k \geq 2$, $m_i \geq 0$, and $d_i \geq 1$.

Under the proportional sampling strategy, $\frac{n_{i_r}}{d_i} = \frac{n_r}{d} = c$ for $i = 1,\ 2,\ \ldots,\ k$, where $c > 0$ is a constant. Hence

$$\begin{aligned} D_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) \ -\ D_R(n_r) \ &=\ m\ \left(1-\frac{1}{d}\right)^{n_r} \ -\ \sum_{i=1}^{k} m_i\ \left(1-\frac{1}{d_i}\right)^{n_{i_r}} \\ &=\ \sum_{i=1}^{k} m_i\ \left[\left(1-\frac{1}{d}\right)^{n_r} \ -\ \left(1-\frac{1}{d_i}\right)^{n_{i_r}}\right] \\ &=\ \sum_{i=1}^{k} m_i\ \left[\left(1-\frac{1}{d}\right)^{cd} \ -\ \left(1-\frac{1}{d_i}\right)^{cd_i}\right] \\ &=\ \sum_{i=1}^{k} m_i\ \left[\left(\left(1-\frac{1}{d}\right)^{d}\right)^{c} \ -\ \left(\left(1-\frac{1}{d_i}\right)^{d_i}\right)^{c}\ \right] \end{aligned}$$

Since $(1-\frac{1}{x})^x$ increases as x increases and $d > d_i$,

$$\left(1-\frac{1}{d}\right)^{d} > \left(1-\frac{1}{d_i}\right)^{d_i} \geq 0$$

$$\left[\left(1-\frac{1}{d}\right)^{d}\right]^{c} \ -\ \left[\left(1-\frac{1}{d_i}\right)^{d_i}\right]^{c} > 0$$

Thus, $D_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) - D_R(n_r) > 0$. ■

4.2.2 Optimally Refined Proportional Sampling Strategy

Following the same arguments as proposed by Chan *et al.* [11], it can be easily shown that the optimally refined proportional sampling (ORPS) strategy can be applied to refine the proportional sampling strategy to give a higher D-measure. Consider a partitioning scheme P_1 consisting of two subdomains $\mathcal{D}_1$ and $\mathcal{D}_2$ of sizes d_1 and d_2, respectively. Let n_1 and n_2 test cases be selected from $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Suppose we apply the proportional sampling strategy in the test case selection, so that $\frac{n_1}{n_2} = \frac{d_1}{d_2}$. Let D_{P_1} be the resulting D-measure.

If $n_1 > 1$, we can refine P_1 by partitioning $\mathcal{D}_1$ into two disjoint subsets, $\mathcal{D}_{11}$ and $\mathcal{D}_{12}$, of sizes d_{11} and d_{12}, respectively, such that $d_{11} + d_{12} = d_1$. Following the proportional sampling strategy, we select n_{11} and n_{12} test cases such that $n_{11} + n_{12} = n_1$ and $\frac{n_{11}}{n_{12}} = \frac{d_{11}}{d_{12}}$. The refined partitioning scheme, denoted by P_2, consists of three partitions, namely, $\mathcal{D}_{11}$, $\mathcal{D}_{12}$, and $\mathcal{D}_2$. Since $\frac{n_{11}}{d_{11}} = \frac{n_{12}}{d_{12}} = \frac{n_2}{d_2}$, the condition for the proportional sampling strategy has been satisfied. By Theorem 2, the D-measure with n_1 test cases from $\mathcal{D}_1$ is less than that with n_{11} and n_{12} test cases selected from $\mathcal{D}_{11}$ and $\mathcal{D}_{12}$, Hence, we have $D_{P_2} > D_{P_1}$.

Such a refinement process can be repeated until the input domain is divided into n equal-sized subdomains with one and only one test case being selected from each subdomain. In each refinement, the D-measure increases. Thus, if there is no preferred partitioning scheme, the ORPS strategy should be used instead of random testing when the division of the input domain into equal-sized subdomains is straightforward.

4.2.3 Follow-the-Crowd Strategies

Chen and Yu [7] stated that, when test cases are selected with replacement, and when $\sum_{i=1}^{k} n_{i_r} = n_r$, if $(\theta_i - \theta_j)(\sigma_i - \sigma_j) \geq 0$ for all $i,\ j = 1,\ 2,\ \ldots,\ k$, then $E_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) \geq E_R(n_r)$, and if $(\theta_i - \theta)(\sigma_i - \sigma) \geq 0$ for all $i,\ j = 1,\ 2,\ \ldots,\ k$, then $E_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) \geq E_R(n_r)$. These sufficient conditions no longer hold with respect to the expected number of distinct failures detected. In fact, as illustrated in the following example, $D_P(n_{1_r},\ \ldots,\ n_{k_r})$ can be greater than or less than $D_R(n_r)$.

Example 6
Consider $d = 99$, $\theta = \frac{10}{99}$, and $n = 20$; $d_1 = 20$, $d_2 = 79$, $\theta_1 = \frac{2}{20}$, $\theta_2 = \frac{8}{79}$, $n_{1_r} = 4$, and $n_{2_r} = 16$. We have $\theta_1 < \theta < \theta_2$ and $\sigma_1 < \sigma < \sigma_2$.

$D_R(20) = 1.837605$
$D_P(4,\ 16) = 1.846143 > D_R(20)$

Consider $d = 101$, $\theta = \frac{10}{101}$, and $n = 20$; $d_1 = 20$, $d_2 = 81$, $\theta_1 = \frac{2}{20}$, $\theta_2 = \frac{8}{81}$, $n_{1_r} = 8$, and $n_{2_r} = 12$. We have $\theta_1 > \theta > \theta_2$ and $\sigma_1 > \sigma > \sigma_2$.

$D_R(20) = 1.804555$
$D_P(8,\ 12) = 1.781090 < D_R(20)$ ■

In the same article, Chen and Yu further stated that, for partition testing such that test cases are selected with replacement, if $\theta_1 = \theta_2 = \cdots = \theta_k$, then $E_P(n_{1_r},\ n_{2_r},\ \ldots,\ n_{k_r}) = E_R(n_r)$. This is not necessarily true when the expected number of distinct failures is used as the effectiveness metric. The following example can be used to illustrate this.

Example 7
Consider $d = 100$, $d_1 = 20$, $d_2 = 80$, $m_1 = 2$, $m_2 = 8$, and $n = 10$. We have $D_R(10) = 0.956179$.

Case (A)
$n_{1_r} = 1$ and $n_{2_r} = 9$. $D_P(1,\ 9) = 0.956288$.

Case (B)
$n_{1_r} = 4$ and $n_{2_r} = 6$. $D_P(4,\ 6) = 0.952547$. ■

Hence, $D_P(n_{1_r},\ \ldots,\ n_{k_r})$ can be greater than or less than $D_R(n_r)$ when $\theta_1 = \theta_2 = \cdots = \theta_k$, depending on the values of n_{i_r}.

4.2.4 Partial Sums Condition

Consider two partition testing strategies A and B with the same partitioning scheme but different test case allocation schemes such that test cases are selected with replacement. As illustrated by the following example, we observe that the partial sums condition no longer guarantees that one strategy exposes more distinct failures than the other. In other words, given $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$ and $\sum_{i=1}^{k} n_{Ai} = \sum_{i=1}^{k} n_{Bi}$, the condition $\sum_{i=1}^{r} n_{Ai} \geq \sum_{i=1}^{r} n_{Bi}$ for all $r = 1,\ 2,\ \ldots,\ k-1$ does not guarantee that $D_P(n_{A1},\ n_{A2},\ \ldots,\ n_{Ak}) \geq D_P(n_{B1},\ n_{B2},\ \ldots,\ n_{Bk})$.

Example 8
Suppose $d_1 = 10$, $\theta_1 = 0.1$, $d_2 = 100$, and $\theta_2 = 0.08$.
For Strategy A, take $n_1 = 4$ and $n_2 = 1$. We have $D_P(4,\ 1) = 0.4239$.
For Strategy B, take $n_1 = 3$ and $n_2 = 2$. We have $D_P(3,\ 2) = 0.4302 > D_P(4,\ 1)$. ■

5 Application Guidelines

As a result of this study, we can provide software testers with a number of useful guidelines to help them compare the potential effectiveness of test case selection and allocation strategies, and hence choose the appropriate ones.

(a) When test cases are selected without replacement, the D-measure and E-measure are equivalent. When test cases are selected with replacement, however the "expected number of distinct failures detected" gives more precise information than the "expected number of failures detected". Hence, the D-measure should replace the E-measure as the effectiveness indicator for the capability of detecting failures.

(b) Consider the situation when test cases are selected with replacement. Among the existing sufficient conditions that favor partition testing over random testing, the proportional sampling strategy, and the optimally refined proportional sampling strategy remain the most useful and practical strategies. They guarantee that partition testing is better than random testing with respect to both the P-measure and D-measure. The other strategies fail to do so when we use the D-measure as the yardstick. We therefore propose that, given a preferred partitioning scheme but no special preferences for test case allocation, the proportional sampling strategy should be used as the test case allocation scheme. When there is no preferred partitioning scheme and the division of the input domain into equal-sized subdomains is easy, the optimally refined proportional sampling strategy should be used instead of random testing.

(c) Consider the situation when test cases are selected without replacement. When the probability of detecting at least one failure (the Q-measure) is considered, none of the existing sufficient conditions that favor partition testing over random testing remain applicable. In other words, the proportional sampling strategy, optimally refined proportional sampling strategy and follow-the-crowd strategies do not guarantee that partition testing is better than random testing.

Nevertheless, given n_d distinct test cases and $n_r = \dfrac{\log(1-\frac{n_d}{d})}{\log(1-\frac{1}{d})}$, we find $P_R(n_r) = 1 - \left(1-\dfrac{m}{d}\right)^{n_r}$ to be a close approximation of $Q_R(n_d)$. This approximation provides an indication on how many additional test cases are required for test case selection with replacement in order to give more or less the equivalent testing effectiveness as test case selection without replacement. The additional overheads in test execution can be estimated by $(n_r - n_d) \times$ *cost of running a test case*. On the other hand, there are payoffs for selecting test cases with replacement. It is easier to apply because there is no need to check the duplication of test cases. Software testers can use the above reasoning to decide whether it is better to select test cases with or without replacement, depending on the testing situation they are facing.

6 Conclusion

So far, studies on the effectiveness of software testing have overwhelmingly assumed that test cases are selected with replacement. Our present study aims at enhancing the understanding of test case selection without replacement. This scenario is mathematically much more complex, and hence few research results have been reported. We have found a lower bound for the probability of detecting at least one failure when n_d test cases are selected without replacement (which we call the Q-measure or $Q_R(n_d)$). We have proved that if $n_r = \dfrac{\log(1-\frac{n_d}{d})}{\log(1-\frac{1}{d})}$ and $P_R(n_r) = 1 - \left(1-\dfrac{m}{d}\right)^{n_r}$, then $Q_R(n_d) \geq P_R(n_r)$. Furthermore, we conjecture that $P_R(n_r)$ is a close approximation of $Q_R(n_d)$ after extensive simulations.

The use of $P_R(n_r)$ as an approximation to $Q_R(n_d)$ can help simplify the analysis of testing effectiveness with respect to the probability of detecting at least one failure when test cases are selected without replacement. This approximation also provides a way for software testers to compare the cost effectiveness between test case selection with and without replacement.

We have introduced a new effectiveness metric, namely the expected number of distinct failures detected (which we call the D-measure). It is a better effectiveness indicator than the expected number of failures detected (E-measure) used in other literature, since in the latter metric, the failures caused by the same input may be counted more than once when test cases are selected with replacement.

Recently, a number of partition testing strategies that outperform random testing were suggested. We have reviewed these strategies in the contexts of the D-measure and Q-measure, and proposed application guidelines on the use of these strategies.

When test cases are selected with replacement, we have concluded that the proportional sampling strategy, including the optimally refined proportional sampling strategy as a special case, is the only strategy that guarantees to outperform random testing in both the contexts of P-measure and D-measure. Using this strategy, the P-measure, or the probability of detecting at least one failure, is no less than the P-measure of random testing. Furthermore, the D-measure, or the expected number of distinct failures detected, is strictly greater than the D-measure of random testing. The other strategies that favor the P-measure of partition testing over random testing fail to guarantee similar results for the D-measure.

References

[1] R. Hamlet, Random testing, *Encyclopedia of Software Engineering*, J. J. Marciniak, Ed., John Wiley, New York, 1994, pp. 970–978.

[2] R. M. Hierons and M. P. Wiper, Estimation of failure rate using random and partition testing, *Software Testing, Verification and Reliability* **7** (3) (1997) 153–164.

[3] P. S. Loo and W. K. Tsai, Random testing revisited, *Information and Software Technology* **30** (7) (1988) 402–417.

[4] G. J. Myers, *The Art of Software Testing*, John Wiley, New York, 1979.

[5] J. W. Duran and S. C. Ntafos, An evaluation of random testing, *IEEE Transactions on Software Engineering* **10** (4) (1984) 438–444.

[6] R. Hamlet and R. N. Taylor, Partition testing does not inspire confidence, *IEEE Transactions on Software Engineering* **16** (12) (1990) 1402–1411.

[7] T. Y. Chen and Y. T. Yu, On the expected number of failures detected by subdomain testing and random testing, *IEEE Transactions on Software Engineering* **22** (2) (1996) 109–119.

[8] E. J. Weyuker and B. Jeng, Analyzing partition testing strategies, *IEEE Transactions on Software Engineering* **17** (7) (1991) 703–711.

[9] F. T. Chan, T. Y. Chen, and T. H. Tse, On the effectiveness of test case allocation schemes in partition testing, *Information and Software Technology* **39** (10) (1997) 719–726.

[10] T. Y. Chen and Y. T. Yu, On the relationship between partition and random testing, *IEEE Transactions on Software Engineering* **20** (12) (1994) 977–980.

[11] F. T. Chan, T. Y. Chen, I. K. Mak, and Y. T. Yu, Proportional sampling strategy: guidelines for software testing practitioners, *Information and Software Technology* **38** (12) (1996) 775–782.

[12] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York, 1996.

[13] H. Leung and T. Y. Chen, A new perspective of the proportional sampling strategy, *The Computer Journal* **42** (8) (1999) 693–698.