ON SOME RELIABILITY ESTIMATION PROBLEMS IN RANDOM AND PARTITION TESTING

Markos Z. Tsoukalas
Computer Science Program
University of Texas at Dallas
Richardson, TX 75083–0688

Joe W. Duran
Video Telecom
1901 W.Braker Lane
Austin, TX 75858

Simeon C. Ntafos
Computer Science Program
University of Texas at Dallas
Richardson, TX 75083–0688

Abstract: Random testing is receiving increasing attention in recent years. Aside from its relative simplicity and low cost, studies have shown that random testing is an effective testing strategy. An advantage of random testing is that the reliability of the program can be estimated from the test outcomes. In this paper we extend the Thayer-Lipow-Nelson reliability model to account for the cost of errors. Also we compare random with partition testing by looking at upper confidence bounds for the cost weighted performance of the two strategies.

Index Terms: Reliability Estimation, Partition Testing, Random Testing.

I Introduction.

Program testing strategies attempt to increase our feelings of confidence on the reliability of a program by executing the program on a (usually small) subset of its inputs. The difference among testing strategies is on the criteria for selecting this subset. Most strategies use some kind of knowledge of the program, be that the program's structure, its functional characteristics, information on common errors or combinations thereof. In random testing the program is treated as a black box and test cases are generated by selecting random values for each of the input variables (these can be determined from the input specification and program documentation). There is a tendency to dismiss random testing as ineffective and expect that strategies based on knowledge are the only effective way to do testing. Still, a number of practitioners have proposed the use of random testing [GR73,TLN78], especially for the final testing of software. Mills [M70] reported that acceptance testing of programs developed using the Clean Room design methodology is done exclusively with random test cases and that this approach is successful. The trend in software methodologies is to perform validation at each stage of software development. With respect to final testing, this reduces the need and effectiveness of testing strategies that are based on knowledge of the program's structure and its functional properties since that knowledge has already been used in validating the early design phases. On the other hand, the value of random testing increases since it is more likely to test the program in novel ways and uncover problems that were missed during development. Also, studies reported in [DN84], [HT88], [Ha89] show that random testing is comparable to path testing with respect to inferences of the operational reliability of the program (the operational reliability of a program is a measure of the proportion of the run-time distribution of inputs that the program executes correctly). All this, together with its relative simplicity and lower cost, indicate that random testing is a valuable testing strategy that merits more attention.

An important advantage of random testing over most other testing strategies is that it is amenable to statistical analysis and allows us to estimate the reliability of a program from test outcomes. Here we take reliability to be a measure of the proportion of the inputs drawn from a given input distribution that the program treats correctly. This is closely related to alternative views of reliability such as Mean Time to Failure (MTTF) and availability, since these quantities can be estimated from the failure rate of the program. In [TLN78], a model for obtaining confidence bounds on the reliability of a program from the outcomes of random test cases was presented. This model treats all errors as if they had equal costs. In practice, the cost of errors varies widely from errors with negligible costs to catastrophic ones. In this paper we present some results on incorporating the cost of errors into the model described in [TLN78] . We also present theoretical support for the experimental comparisons between partition and random testing in [DN84, HT88, Ha89], taking into account the cost factors for the failures in the subdomains of the input data space. In section II we solve the problem of estimating the unknown failure rates of each subdomain so as to maximize the weighted cost with a prescribed confidence level in the cases of partition and random testing respectively. This is done for the general and some important special cases frequently encountered in practice. Also we give conditions under which partition

TH0336-5/91/0000/0194$01.00

testing outperforms random testing and we give theoretical justification why partition testing does not perform significantly better than random testing in practice. In section III we determine the optimal number of test cases based on the criterion of minimizing the cost estimates of section II in addition to the cost of performing the tests. Finally in section IV we present our conclusions.

II. Cost Weighted Reliability Confidence Bounds.

In this section we extend the reliability model of Thayer-Lipow-Nelson [TLN78] to account for the cost of errors. Let θ be the probability that the program will fail to execute correctly on some input case from a given input distribution. If random testing is performed, that is the program is executed a large number of times with random inputs from this distribution, the observed failure rate will converge to θ. Thus, we refer to θ as the failure rate of the program. If the input distribution corresponds to the operational profile of the program, θ becomes a measure of the operational reliability of the program. In [TLN78], it was shown that if n random test cases are run and f failures are discovered, then θ^*, the $1-a$ upper confidence bound on θ is the largest θ such that:

$$\sum_{j=0}^{f} \binom{n}{j} \theta^j (1-\theta)^{n-j} \geq a \tag{1}$$

This means that the proportion of times that the confidence bound estimating process will yield a θ^* which exceeds the actual θ of the program is $1-a$, or $(1-a)100\%$. If no errors are detected by the n random test cases, we have that

$$\theta^* = 1 - a^{1/n} \tag{2}$$

When it can be obtained, θ^* is a useful measure of software reliability. However, note that this measure does not allow for the fact that some failures are more costly than others. In [TLN78], the failure rate of a program is described in terms of the failure probabilities over the subsets of some partition of its input domain. We refer to this as partition testing to distinguish it from random testing. Since the consequences of possible program failures can vary greatly, it is desirable to extend the partition testing model further to account for failure cost variations. Let the input domain D of a program to be tested be partitioned into k subsets so that $D = D_1 \cup D_2 \cup \ldots \cup D_k$. A partition might be chosen, for example, to ensure path testing or to ensure the testing of the various functions the program is required to perform. Let c_i be the cost penalty which would be incurred by the program's failure to execute properly on input from D_i (estimation of the cost factors is an important and difficult problem that we do not address here). Note that the cost penalties themselves could be used to obtain the partition of D. Let p_i be the probability that a randomly selected input will belong to D_i. (Any desired input distribution may be used but the resulting inferences will of course be valid only with respect to that input distribution.) The cost weighted failure rate for the program as a whole is:

$$C = \sum_{i=1}^{k} c_i \, p_i \, \theta_i \tag{3}$$

This also represents the average failure cost per run. Since the θ_i's are not usually known in advance, we would like to estimate C from test outcomes. Suppose that a series of n test runs is executed, with n_i input cases chosen randomly from D_i, $0 \leq i \leq k$, so that $\sum_{i=1}^{k} n_i = n$. Let f_i be the number of failures observed over D_i. The maximum likelihood estimate for C is:

$$\hat{C} = \sum_{i=1}^{k} c_i \, p_i \, f_i / n_i \tag{4}$$

In many cases one would like to have an upper confidence bound, C^* for C (as with θ^* for θ). Since partition testing (selecting $n_i > 0$ test cases from each D_i) is a lot of work, and since random testing seems quite often to be cost effective, it is of interest to look at C^* for random testing which we denote C_r^*. We have that:

$$\sum_{i=1}^{k} c_i \, p_i \, \theta_i \leq \sum_{i=1}^{k} c_{max} \, p_i \, \theta_i = c_{max} \sum_{i=1}^{k} p_i \, \theta_i \tag{5}$$

where c_{max} is the largest c_i. Under random testing, if we have found f errors in all subdomains we want the following relationships to be satisfied :

$$G(f, n, \theta) = \sum_{j=0}^{f} \binom{n}{j} \theta^j (1-\theta^j) \geq a \tag{6}$$

where $\theta = \sum_{i=1}^{k} p_i \, \theta_i$.

Without loss of generality we assume, that $c_1 \geq c_2 \geq \ldots \geq c_k$ (we can always rearrange the indices). From the discussion above we arrive at:

Theorem 1: If we perform random testing and find f errors in n test cases then C_r^* is the solution of the linear programming problem:

$$max\left\{\sum_{i=1}^{k} c_i\, p_i\, \theta_i\right\} \text{ subject to} \tag{7}$$

$$\sum_{i=1}^{k} p_i\, \theta_i \leq \theta^* \,,\quad 0 \leq \theta_i \leq 1 \,,\ 1 \leq i \leq k \tag{8}$$

where $\theta^* = \theta^*(f, n, a)$ is determined from the relation

$$G(f, n, \theta^*) = \sum_{j=0}^{f} \binom{n}{j} \theta^{*\,j} (1 - \theta^{*\,j}) = a \tag{9}$$

The solution is of the form:

$$\theta_i = 1,\ 1 \leq i \leq s, \tag{10}$$

$$\theta_{s+1} = \left(\theta^* - \sum_{j=1}^{s} p_j\right) / p_{s+1}\,, \tag{11}$$

$$\theta_i = 0,\ s+2 \leq i \leq k\,. \tag{12}$$

$$C_r^* = \sum_{i=1}^{s} c_i\, p_i + c_{s+1}\left(\theta^* - \sum_{i=1}^{s} p_i\right) \tag{13}$$

where s is the minimum integer such that the following relations are satisfied:

$$p_{s+1} \geq \theta^* - \sum_{i=1}^{s} p_i, \quad p_{s+1} > 0,\ 0 \leq s \leq k-1. \tag{14}$$

Proof: Since c_1 is the maximum cost factor, if $p_1 < \theta$ then we take $\theta_1 = 1$ and we try to maximize:

$$\sum_{i=2}^{k} c_i\, p_i\, \theta_i \,, \text{ where now we have} \tag{15}$$

$$\sum_{i=2}^{k} p_i\, \theta_i \leq \theta^* - p_1 \tag{16}$$

Now the dimension of the problem has decreased by 1. We continue in this manner to reduce the dimension of the linear programming problem until we find the minimum s such that (14) holds. From the formulation of the procedure it is clear that it terminates in at most k steps (since $p_k \geq \theta^* - \sum_{i=1}^{k-1} p_i$ as $\sum_{i=1}^{k} p_i = 1 \geq \theta^*(f, n, a)$). For $\theta^* = \theta^*(f, n, a)$ we use an approximation of the binomial distribution by the gaussian distribution (see [KJ70,PR82,TDN90]). If the cost factors c_i , $1 \leq i \leq k$, are not all distinct then we may have more than one optimal solutions. ■

Next we obtain exact solution for the important special case when no failures are detected. If one performs n randomly selected test runs with no failures detected, then

$$\sum_{i=1}^{k} p_i\, \theta_i \leq 1 - a^{1/n} \text{ and } C_r^* \leq c_{max}\,(1 - a^{1/n}).$$

This upper bound can be achieved as shown in the following corollary:

Corollary 1: If n random tests are performed without discovering any failures, we obtain exact expressions for C_r^* by replacing θ^* with $1 - a^{1/n}$ in the formulas in theorem 1.

Now we turn our attention to partition testing. In what follows we will find corresponding theorems for C^* whenever it is possible. We denote C^* as C_p^* when it refers to partition testing in the sequel. Again we want to maximize $C = \sum_{i=1}^{k} c_i\, p_i\, \theta_i$ but the conditions are different than those for random testing. The constraint

$$\prod_{i=1}^{k} \sum_{j=1}^{f_i} \binom{n_i}{j} \theta_i^j (1 - \theta_i)^{n_i - j} \geq a \tag{17}$$

must be satisfied. This leads us to a system of equations that can be solved using a Newton-Raphson procedure as it is shown in [TDN90].

Software Engineering practice emphasizes testing in the early stages of the software lifecycle. Also random testing has been especially recommended for the final testing of software. Thus it is natural to assume that the testing process has reached to a point where the number of errors expected in each subdomain is fairly small compared to the test cases per subdomain (in the context of using random testing for the final testing of software). In this case we choose n_i proportional to $c_i\, p_i$, so as to achieve best results with the given number of test cases. We anticipate it to be close to the optimal policy for selecting the n_i's because the small f_i's will introduce relatively negligible deviations from the optimal selection. Choosing f_i distributed uniformly in the range [0, 2] gave results (see figure 1) similar to the case when no errors are detected (which is discussed later, see figure 2). The better performance of partition testing is attributed to the fact that in partition testing we choose n_i's taking into account both the cost of a subdomain and the frequency with which the paths pertaining to this subdomain are executed. In contrast, in random testing it appears that the n_i's are selected proportional to the execution frequencies p_i, $1 \leq i \leq k$, and so the variable cost of the subdomains is not taken into account.

An interesting special case is when all of the f_i are 0. In this case, C_p^* is the maximum value of $C = \sum_{i=1}^{k} c_i\, p_i\, \theta_i$ subject to the constraints,

$$\prod_{i=1}^{k} (1 - \theta_i)^{n_i} \geq a \qquad \text{and} \qquad 0 \leq \theta_i \leq 1. \tag{18}$$

Thus, determining C_p^* requires solving a non-linear programming problem. Using the Kuhn-Tucker conditions for this problem we have:

Theorem 2: If n_i test cases are executed randomly from each D_i without discovering any errors, then the exact solution for C_p^* is:

$$C_p^* = \sum_{i=1}^{k} c_i\, p_i \left[1 - \frac{n_i}{c_i\, p_i} \prod_{i=1}^{k} \left(\frac{c_i\, p_i}{n_i} \right)^{n_i/n} a^{1/n} \right] \tag{19}$$

Now we are in a position to compare partition and random testing for the case when no errors are detected. We have:

Theorem 3: If no errors are detected and all cost factors c_i are equal (we take them to be 1 by normalizing them) we have that $C_p^* \leq C_r^*$ the following relation holds:

$$\prod_{i=1}^{k} \left(\frac{p_i}{n_i} \right)^{n_i} \geq n^{-n} \quad . \tag{20}$$

This inequality is true only as an equality provided that each n_i=n p_i. If this is not true then the inequality is reversed and we have that random testing is better than partition testing ($C_p^* > C_r^*$).

Proof: If we manipulate algebraically the expressions for C_p^* and C_r^* using the assumption c_i=1 we arrive at (20). If we test each subdomain with n_i test cases, the left side of the inequality is minimized if we choose n_i=n p_i. This is proved if we take the Kuhn-Tucker conditions for the left side of the inequality using $\sum_{i=1}^{k} n_i = n$ as a constraint. Then using the optimal n_i=n p_i the inequality becomes the equality n^{-n}=n^{-n}. So under these conditions partition testing gives us exactly the same results as random testing. If we do not use the optimal n_i for each subdomain we have the reverse relation: $C_p^* > C_r^*$ which means that random testing is better in this case. ■

Next we consider another important special case. It seems that in order to concentrate our effort in the more costly subdomains we should choose n_i proportional to $c_i\, p_i$ to take account of the cost and frequency of execution concurrently. It is proved also that this policy is the optimal selection of the test cases for each subsdomain. If we do so we have:

Theorem 4: If no errors are detected and each n_i is chosen to be proportional to $c_i\, p_i$, then $C_p^* = \left(1 - a^{1/n}\right) \sum_{i=1}^{k} c_i\, p_i$ and this selection minimizes the cost C_p^* under the constraint $\sum_{i=1}^{k} n_i = n$

Proof: Choose $n_i = c_i\, p_i\, n / \sum_{j=1}^{k} c_j\, p_j$, where $n = \sum_{j=1}^{k} n_j$. Substituting this in the equation for C_p^* in theorem 2 we get the expression in the statement of the theorem. The optimality of this selection is proved by considering the Kuhn-Tucker conditions for C_p^* and the constraint in the statement of the theorem. ■

Numerical calculations for C_p^* and C_r^* were done using c_i, $1 \leq i \leq k$ distributed uniformly over the range [0, 1000] and p_i, $1 \leq i \leq k$ distributed uniformly on [0, 1] and then normalized so that they are a probability measure. As can be seen from Fig.2 partition testing clearly outperforms random testing in this case. This seems at first to come into disagreement with the results in [DN84,HT88,Ha89]. But if we take into account that in those studies no variable failure costs were assumed we can easily explain the difference in performance. Because the cost is distributed uniformly and since $p_i = O(1/k)$ for most $1 \leq i \leq k$, C_p^* tends to acquire values around the average of c_i, $1 \leq i \leq k$ multiplied with the upper bound $1 - a^{1/n}$ for θ^* (with confidence level 1–a). In contrast as n becomes larger and larger $p_1 \geq 1 - a^{1/n}$ and as a result C_r^* takes the value $c_1\left(1 - a^{1/n}\right)$. As the average of c_i, $1 \leq i \leq k$, is expected to be around the middle of the distribution range the ratio 2:1 for C_r^*/C_p^* is anticipated and is indeed observed when k is greater than 10 (it is in fact the ratio c_1/c_{av} when n becomes large enough and where $c_{av} = 1/k \sum_{i=1}^{k} c_i$ under any distribution of c_i, p_i, $1 \leq i \leq k$). Calculations using a distribution where 20% of the p_i, $1 \leq i \leq k$ θ_i, where chosen to be 10 times bigger than the rest showed small difference from the first set of calculations. The situation improved slightly for random testing but clearly partition testing was superior.

In partition testing it is common practice to select one test case for each subdomain, i.e. n_i=1, $1 \leq i \leq k$. In this case C_p^* can be determined as follows: Let $k_1 = k - \sum_{i=1}^{k} f_i = k - k_2$ where f_i are all 0 or 1. Without loss of generality we assume that in the first k_1 subdomains we do not discover any errors and we discover one error in each one of the other subdomains (we can always rearrange the indices). The new constraints are :

$$\prod_{i=1}^{k_1} (1-\theta_i) \geq a \qquad (21)$$

$$\text{and} \qquad 0 \leq \theta_i \leq 1, \qquad 1 \leq i \leq k. \qquad (22)$$

So we have the following theorems :

Theorem 6: If we perform partition testing with only one test input for each subdomain D_i then

$$C_p^* = \sum_{i=1}^{k_1} c_i\, p_i \left[1 - \frac{1}{c_i\, p_i} \left(\prod_{j=1}^{k_1} (c_j\, p_j) \right)^{1/k_1} a^{1/k_1} \right] \qquad (23)$$

$$+ \sum_{i=k_1+1}^{k} c_i\, p_i \qquad (24)$$

$$= \sum_{i=1}^{k} c_i\, p_i - k_1 \left(\prod_{j=1}^{k_1} (c_j\, p_j) \right)^{1/k_1} a^{1/k_1} \qquad (25)$$

Proof: Using the Kuhn-Tucker conditions for this non-linear programming problem we have a system of non-linear equations for the θ_i's. As the only bounds for θ_i, $k_1 \leq i \leq k$, are $0 \leq \theta_i \leq 1$ we take them all equal to 1 to maximize $C = \sum_{i=1}^{k} c_i\, p_i\, \theta_i$. Substituting these values in the equation for C, we have the equality stated above. ∎

By manipulating algebraically expression (14) for C_r^* from theorem 1 and (23–25) for C_p^* in theorem 6 we have the following theorem for the case we have equal cost factors:

Theorem 7: If we have that c_i=1, n_i=1, $1 \leq i \leq k$, we have that $C_p^* \leq C_r^*$ iff we have

$$C_p^* = 1 - a^{1/k_1} \left(\prod_{i=1}^{k_1} p_i \right)^{1/k_1} \leq \theta^* = \theta^*(a, k_1, k) = C_r^* \qquad (26)$$

The significant difference between random and partition testing found in our calculations when we run only one test case per subdomain does not contradict the experimental findings of [DN84,HT88,Ha89] where it was found that, contrary to intuitive expectations, partition testing is not much better than random testing in practice. In their experimental findings [DN84] found this result under the assumption that there are no more than one kind of error in each subdomain and also they assumed a known distribution for θ_i, $1 \leq i \leq k$, where in our model we assume them to be variables under the constraints to ensure meaningful maxima to $C = \sum_{i=1}^{k} c_i\, p_i\, \theta_i$. Our calculations are in agreement with theorem 7 which gives necessary and sufficient conditions so that $C_p^* \geq C_r^*$. In the case that no errors are detected theorem 7 when $k_1 = k$ clearly reduces to the proposition $C_p^* \geq C_r^*$ as $C_p^* = 1 - \prod_{i=1}^{k} p_i^{1/k}\, a^{1/k} \geq \theta^* = \theta^*(a, k, k) = 1 - a^{1/k} = C_r^*$. When we incorporate the variable costs in our calculations the result remained in favor of random testing. This demonstrates the enormous benefits to partition testing when we optimally select the optimal selection of n_i to be proportional to $c_i\, p_i$, $0 \leq i \leq k$.

III. Selection of the Optimal Number of Test Cases

Now we are in a position to determine the optimal number of test cases so as to minimize the total cost. Many factors contribute to the total cost and many of them are hard to quantify. We consider two principal components, the cost of developing and running the test cases and the cost associated with the errors that remain in the program after testing (and the associated debugging). We make the simplifying assumption that the first component grows linearly with the number of tests. The second component is the weighted cost C due to the errors remaining in some subdomains of the program. So we can express the total cost C_t as the sum of the two cost components, that is:

$$C_t = w\, n + C = w\, n + \sum_{i=1}^{k} c_i\, p_i\, \theta_i \qquad (27)$$

where w is the cost of developing and running one test case. We can handle the case when the cost of a test depends on the subdomain is intended for. We allocate the number of test cases according to some proportionality constants: That is we select $n_i = \nu_i\, n$, $0 \leq \nu_i \leq 1$, $1 \leq i \leq k$. and so if $0 \leq w_i$, $1 \leq i \leq k$, are the costs pertaining to the cost of a test case at the corresponding subdomains D_i $1 \leq i \leq k$, we have $w = \sum_{i=1}^{k} \nu_i\, w_i$. Now we want to find the particular *n* that minimizes the total cost C_t. This depends on the unknown subdomain failure rates θ_i, $1 \leq i \leq k$. So we choose to minimize with respect to *n* the maximum weighted cost C_t^* which can be expressed as

$$C_t^* = w\, n + C^* = w\, n + \sum_{i=1}^{k} c_i\, p_i\, \theta_i' \qquad (28)$$

where θ_i', $1 \leq i \leq k$, are the values that maximize C (giving C_p^* or C_r^* depending on whether we perform partition or random testing). We denote the total cost for partition testing as $C_{p,t}^* = w_p\, n + C_p^*$ and for random testing as $C_{r,t}^* = w_r\, n + C_r^*$ where w_p and w_r is the cost of one test case for partition and random testing respectively. Using the expressions we have found in section II for C_p^* and C_r^* we arrive at the following theorems:

Theorem 8: Assuming that no errors are found in any subdomain and n_i is chosen proportional to $c_i\, p_i$ then the optimal number of test cases that minimizes $C_{p,t}^*$ is:

$$n_p^* = \sqrt{\frac{\pi}{2}}\,|\ln a| \left[\sum_{j=1}^{+\infty} \left(\frac{1}{2} \sqrt{|\ln a| \frac{w_p}{Q}} \right)^k \frac{k^{k-1}}{k!} \right]^{-1} \quad (29)$$

where $Q = \sum_{i=1}^{k} c_i\, p_i$

Proof: Taking the derivative with respect to n of $C_{p,t}^* = w_p\, n + C_p^*$ and using the expression for C_p^* from theorem 4 we arrive at the equation:

$$n_p^* = \sqrt{|\ln a| \frac{Q}{w_p}}\; a^{1/(2n_p^*)} \quad (30)$$

This equation can be solved using the Lagrange Inversion formula giving (29) above. ■
Similar equations arise for n_r^* where c_{s+1} using the notation of theorem 1 replaces Q in the equations for n_p^*.

We found that partition testing under the assumption that the cost factors vary over some range in a uniform manner, performs much better than random testing if we consider the costs C_p^* and C_r^* as a criterion. Now we are in a position to take into account the testing costs as well. We use the ratio $C_{r,t}^*/C_{p,t}^*$ to compare the two testing strategies. Since the testing costs depend on the values of $\omega = w_p/w_r$ it would be of interest to look at the cost ratio $C_{p,t}^*/C_{r,t}^*$ as a function of ω.

We plotted $C_{r,t}^*/C_{p,t}^*$ versus $\omega = w_p/w_r$ for $a = 0.01$ and $a = 0.1$ in figure 3. We used $k = 100$ and costs uniformly distributed in [0, 1000]. Figure 3 reflects the case when the cost of a random test case is relatively high (50 times smaller than the average cost of an undetected failure). Figure 4 reflects the case when the cost of a random test case is low (5000 times smaller than the average cost of an undetected failure). We also tried non uniform cost distributions for the c_i, $1 \leq i \leq k$, using $c_{av} = 1/k \sum_{i=1}^{k} c_i \approx 100$ and w_r = 0.01, 0.1, 1, 10, obtaining similar results.

IV. Conclusions

We have presented an extension of the Thayer-Lipow-Nelson reliability model [TLN78] that allows us to account for errors of varying costs. We obtain approximate solutions for the general case and exact solutions for many important special cases. In comparing random and partition testing we show that the optimum way to distribute the test cases to the subdomains is to select n_i proportional to $c_i\, p_i$, $0 \leq i \leq k$. Then partition testing outperforms random testing. However if partition testing is done with n_i=1, $0 \leq i \leq k$, (as is often the case in practice), random testing performs much better.

We have also incorporated the cost of developing and running the test cases and obtain the optimal number of test cases for both strategies. Evaluations of the ratio $C_{r,t}^*/C_{p,t}^*$ show that random testing outperforms partition testing for relatively small values of $\omega = w_p/w_r$. Figures 3 and 4 indicate that the relative performance of random testing improves as the ratio ω increases (i.e. random tests are cheaper). It is important to note that random testing surpasses partition testing for relatively small values of ω (e.g. random test cases are 10 – 50 times cheaper than partition test cases). Also note that even if ω is low (<10), the performance of random testing is still within 20 – 40% of that for partition testing.

We are currently investigating various extensions of the model to incorporate prior information for the failure rates θ_i and uncertainty for the costs c_i and the frequencies p_i, $1 \leq i \leq k$.

References.

[AB79] A. Acree, T. Budd, R. DeMillo and R. Sayward, "Mutation Analysis," GIT-ICS-79/08, 1979.

[BM83] D. Bird and C. Munoz, "Automatic Generation of Random Self-Checking Test Cases," IBM Systems Journal, 22(3), 1983, pp. 229-245.

[BU83] T. Budd, "The Portable Mutation Testing Suite," TR83-8, University of Arizona, March 1983.

[CLR90] T.H.Cormen, C.E.Leiserson and R.L.Rivest, "Introduction to Algorithms, " The MIT Press, Massachusetts, 1990.

[CR83] L. Clarke and D. Richardson, "A Rigorous Approach to Error-Sensitive Testing," Proc. 16th HICSS, pp. 197-206, Jan. 1983

[DN84] J. Duran and S. Ntafos, "An Evaluation of Random Testing," IEEE Trans. on Software

Engineering, SE-10(4), July 1984, pp. 438-444.
[DW80] J. Duran and J. Wiorkowski, "Quantifying Software Validity by Sampling," IEEE Trans. on Reliability, R-29, 1980, pp. 141-144.
[GR73] E. Girard and J.-C. Rault, "A Programming Technique for Software Reliability," Proc. 1973 IEEE Symp. on Computer Software Reliability, pp. 45-50.
[HL77] M. Halstead, Elements of Software Science, Elsevier, New York, 1977.
[Ha89] R. Hamlet "Theoretical Comparison of Testing Methods," Proc. 3rd Symp. on Testing, Analysis and Verification, Florida, Dec.1989, pp.28–37.
[HL67] F. Hillier and G. Lieberman, Introduction to Operations Research, Holden-Day, San Francisco, 1967, pp. 515-518.
[HT88] R.Hamlet and R.Taylor "Partition Testing does not Inspire Confidence", Proc. 2nd Workshop on software Testing Verification and Analysis, Banff, Canada, July 1988, pp.206–215.
[KJ70] S. Kotz and N.L.Johnson, Distributions in Statistics: Discrete Distributions, Boston, Houghton Mifflin Company, 1970.
[KP78] B. Kernighan and P. Plauger, The Elements of Programming Style, New York, McGraw-Hill, 1978.
[MC76] T. McCabe, "A Complexity Measure," IEEE Trans. on Software Engineering, Vol. SE-2, No. 4, Dec. 1976, pp. 308-320.
[MI70] H. Mills, "On the Statistical Validation of Computer Programs," IBM FSD Rep., July 1970.
[MI84] H. Mills, Panel Discussion, 7th ICSE, Orlando, March 1984.
[ND85] S. Ntafos and J. Duran, "An Investigation of Stopping Rules for Random Testing," Proc. 18th Hawaii Int'l Conference on System Sciences, Jan. 1985, pp. 684-691.
[PR82] J. K. Patel and G. B. Read, "Handbook of the Normal Distribution," New York and Basel , Mariel Dekker Inc., 1982.
[TLN78] R. Thayer, M. Lipow and E. Nelson, Software Reliability, North-Holland, Amsterdam, 1978.
[TDN90] M.Z.Tsoukalas, J.W.Duran and S.Ntafos, "On Some Reliability Estimation Problems in Random and Partition Testing," UTD Technical Report, October 1990.

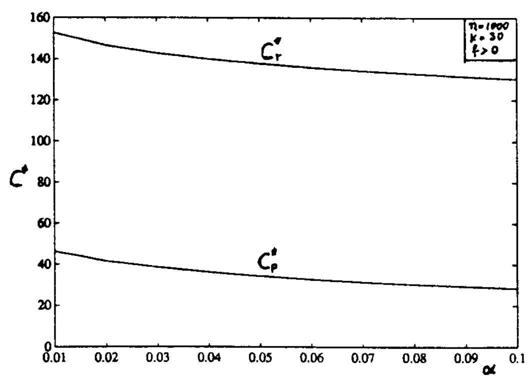

Figure 1

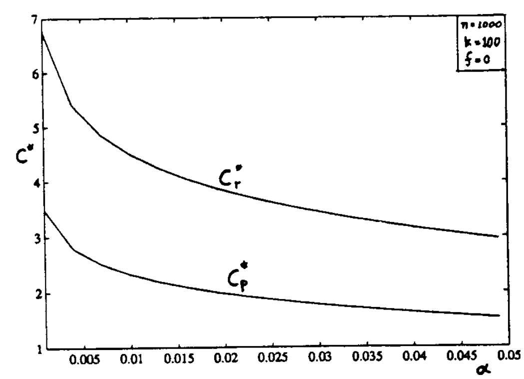

Figure 2

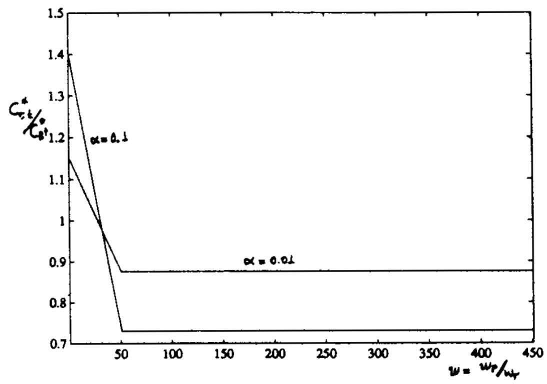

Figure 3

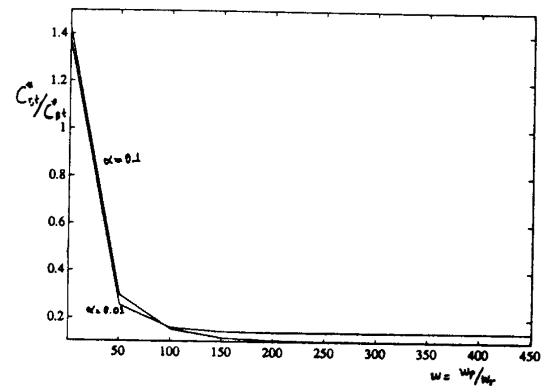

Figure 4