On the Asymptotic Behavior of Adaptive Testing Strategy for Software Reliability Assessment

Junpeng Lv, Bei-Bei Yin, and Kai-Yuan Cai

Abstract—In software reliability assessment, one problem of interest is how to minimize the variance of reliability estimator, which is often considered as an optimization goal. The basic idea is that an estimator with lower variance makes the estimates more predictable and accurate. Adaptive Testing (AT) is an online testing strategy, which can be adopted to minimize the variance of software reliability estimator. In order to reduce the computational overhead of decision-making, the implemented AT strategy in practice deviates from its theoretical design that guarantees AT's local optimality. This work aims to investigate the asymptotic behavior of AT to improve its global performance without losing the local optimality. To this end, a new AT strategy named Adaptive Testing with Gradient Descent method (AT-GD) is proposed. Theoretical analysis indicates that AT-GD, a locally optimal testing strategy, converges to the globally optimal solution as the assessment process proceeds. Simulation and experiments are set up to validate AT-GD's effectiveness and efficiency. Besides, sensitivity analysis of AT-GD is also conducted in this study.

Index Terms—Adaptive testing, operational profile, software reliability, testing strategy

1 Introduction

Mission-critical systems play an important role in daily life, such as avionics and flight safety, financial services, and so on. In such systems, more and more jobs are accomplished by computers and software. One problem is that, some software systems might not be as reliable as hardware systems, which can probably lead to business or property loss. Usually, high reliability is required in mission-critical systems, e.g., less than 1×10^{-9} probability of failure per hour in a flight control system. Thus, software reliability assessment is a critical and hard problem in such systems, since an estimate that significantly deviates from the true value of reliability might lead to overconfidence in the product, and unexpected loss after deploying the system.

In the theory of software reliability assessment, estimating the reliability value as well as the corresponding confidence interval is usually conducted based on the testing data collected during the assessment process [1], [2], [3], [4], [5], [6], [7]. Then, how to choose a proper estimator will be a major concern on providing an accurate and stable reliability estimate. In fact, four properties can be considered to construct a "good" estimator: unbiasedness, consistency, efficiency (e.g., minimum variance) and sufficiency [8]. Minimum variance is often considered as the principle for selecting the estimator since the other three are usually satisfied by many estimators. In statistics, estimator variance is related to how predictable the estimation result is, that is, how close the new estimate will be to the previous one when the assessment is rerun with the same setting. Lower variance often implies more predictable estimate. In addition, variance is also related to confidence interval, that is, lower variance usually implies a tighter confidence interval. Therefore, an unbiased estimator with minimum variance can improve the effectiveness of estimation.

On the other hand, lower variance can improve the efficiency of assessment. In theory, the estimation result can be considered as the sum of a random error and the true value. For an unbiased estimator, the distribution of this error has a mathematical expectation of zero. Lower variance means that this error is more likely to be close to zero. In order to guarantee an estimate accurate enough, the error needs to be restricted to a tight enough interval, which means that the variance of the corresponding estimator must be lowered down to some extent. Note that, when more tests are conducted, the variance of the corresponding estimator will decrease naturally. Therefore, an estimator with lower variance implies that the task of obtaining an accurate enough estimate can be accomplished with fewer tests. This is important for assessments with limited budget, especially for the assessments of mission-critical systems, e.g., flight control systems. It is well known that high assurance is required for flight control systems. However, the assessment of such systems needs a sophisticated simulation environment that is usually shared by multiple projects, and thus, the assessment is required to be as efficient as possible. Under this circumstance, an unbiased estimator with high effectiveness and efficiency is preferred.

In our previous work [9], [10], [11], [12], [13], Adaptive Testing (AT) is adopted to construct an efficient software testing strategy. AT is a software testing technique that results from the application of feedback and adaptive control principles in software testing [9], [12]. It can be treated as the software-testing counterpart of adaptive control. In [10], the optimal and adaptive testing strategies for reliability assessment are proposed. The performance of AT is

• *The authors are with the School of Automation Science and Electrical Engineering, Bejing, China.*
E-mail: {reallJp, yinbeibei, kycai}@buaa.edu.cn.

Manuscript received 15 May 2013; revised 15 Feb. 2014; accepted 23 Feb. 2014; date of publication 10 Mar. 2014; date of current version 1 May 2014.
Recommended for acceptance by M. Pezzã.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSE.2014.2310194

validated by simulations and experiments [11]. However, there are still some unsolved problems in AT.

The major concern is whether the optimality of AT can be guaranteed as it is designed. Although the theoretical foundation of AT for reliability assessment can guarantee its local optimality, the globally asymptotic behavior of AT is not yet analyzed. According to [14], [15], a globally optimal solution on how to choose the estimator with minimum variance can be obtained if the true failure rates are known. Although this solution is implausible in practice as it is quite difficult to get the true values of failure rates, this solution provides a theoretical upper bound for the global performance of AT. Besides, the locally optimal design of AT is obtained by recursive evaluation on all possible following (software) assessment states. After that, which test case should be executed is decided. However, this locally optimal design will lead to huge computational complexity when the number of available test cases is large. In order to reduce the computational complexity, one way is to limit the number of following states, which are forward from the current state and taken into account in the process of the recursive evaluation, in practice [11]. However, one problem still remains, will AT with the theoretical design converge to the globally optimal solution? More importantly, will AT in practice converge to this solution too?

Note that, if AT fails to converge to the globally optimal solution, it might converge to a suboptimal solution. Although these two solutions do not have much difference in many cases, difference of this kind might matter a lot in mission-critical systems if both assessment effectiveness and efficiency are highly required. On the other hand, although AT is believed to outperform Random Testing (RT) and some other testing strategies according to the experimental results, the problem is that the performances of AT under various circumstances cannot be exhaustively validated by experiments. If asymptotic global optimality is guaranteed in theory, the performance of AT can be more convincing, and thus AT can be utilized more widely in practice.

In order to explore the asymptotic behavior of AT without losing its local optimality, a new AT strategy with Gradient Descent method (AT-GD) is proposed in this study. Gradient is usually considered as the direction and steepness of the slopes. It is extensively used in deciding a search direction when a step-size choice is made to solve an optimization problem. In addition, gradient descent method is often adopted to construct a local optimization solution [16]. By introducing gradient descent method into the original AT framework, the asymptotic behavior of AT-GD can be investigated and the upper bound for the global performance of AT strategies can be explored.

The rest of this paper is organized as follows. Section 2 formulates the concerned problem of reliability assessment and gives the optimal allocation (OA) solution for this problem. Section 3 presents the AT-GD strategy with theoretical analysis of its asymptotic behavior and sensitivity analysis of its estimator variance. Simulation and experiments are set up to validate the effectiveness and efficiency of AT-GD in Section 4. Related work is presented in Section 5. Conclusions and future work are listed in Section 6.

2 Problem Formulation

For software reliability assessment, there are mainly two branches: the continuous-time base and the discrete-time base. The former focuses on the reliability behavior measured in terms of CPU execution time and so on. As pointed out by Cai in [17], although the continuous-time assumption is appropriate for a wide scope of systems, there are many systems that do not essentially satisfy this assumption. For example, reliability behavior of the software system for rocket control should be measured in terms of how many rockets are successfully launched, rather than of how long a rocket flies without failures. For such systems, the time base of reliability measurement is essentially discrete rather than continuous.

In fact, the reliability assessment for many mission-critical systems can be viewed as a discrete-time problem, such as, GPS system on an aircraft. GPS system can provide the position of an aircraft for landing and departure procedures. However, GPS data from an aircraft might not be accurate enough for these procedures. Thus, the Ground Based Augmentation System (GBAS) is built to correct the GPS data received by an aircraft. The GBAS system broadcasts the correction information to nearby aircrafts so that GPS system on an aircraft can correct its position and calculate the protection level according to its navigation solution. Since GPS system on an aircraft corrects the GPS data discretely, e.g., at a frequency of 2 Hz, this system can be viewed as a system in discrete-time domain. In practice, the correction of GPS data can be affected by several factors/inputs, such as, atmosphere condition, the status of satellites, the performance of GPS receivers, and even the fault mode and probability of the GBAS system. Usually, an operational profile [18] can be used to describe the occurrence probabilities of different scenarios in the above factors.

In many mission-critical systems, the reliability is required to be as high as possible. On the other hand, the reliability assessment process can be costly, and thus, only limited testing resources are available. This will lead to few failures being observed after the assessment, which will increase the difficulty of delivering a highly accurate estimate. Therefore, both high effectiveness and efficiency are required in the reliability assessment. Under such circumstance, a reliability estimator with minimum variance is preferred. Thus, the main target of this study is to minimize the variance of reliability estimator for such systems in discrete-time domain.

2.1 Assumptions

In order to formulate the concerned problem, some assumptions on the systems in discrete-time domain should first be presented.

1. The software under test or reliability assessment is frozen. The assessment aims to find out the current status of system reliability. Thus, during the assessment process, the system will not be modified even though it can be modified after the assessment.
2. The input domain of the software under test can be divided into m subdomains, denoted as $\{D_1, D_2, \ldots, D_m\}$. According to some criterion, these subdomains

are divided according to some criteria, e.g., functionality [19]. Note that, the test cases in the same subdomain should have some common properties. These properties are related to the software under test, and thus the choice of division can vary a lot for different software systems.

3. The operational profile of the software under test can be described as$\{< D_i, p_i >,\ i = 1, 2, \ldots, m\}$, where p_i denotes the probability that an input is selected from D_i in the phase of software operation, and $\sum_{i=1}^{m} p_i = 1$.
4. The output of the software under test is independent of the testing history. There are some cases where one test case is deemed failure-free but it actually leads to some fault status that cannot be observed due to limited knowledge of test oracle. In such situation, this test case is not considered to reveal a failure. However, this faulty status may cause a failure to be observed when some following test cases are executed even though no failure should be observed after executing these test cases. Then, the latter test cases are mistakenly considered to reveal a failure. This will lead to some error in reliability estimation, but this problem should be considered as a test oracle problem rather than a testing strategy one since it can be encountered by every testing strategy without a proper test oracle.

In addition, some assumptions on the assessment process are listed as follows.

5. A software test includes selecting a test case from the input domain/subdomain, executing the test case, collecting the testing data, and updating the estimates of necessary parameters. Moreover, a total of t test cases are allowed in the assessment, that is,

$$\sum_{i=1}^{m} n_i = t$$

where n_i denotes the number of test cases from subdomain D_i.

6. Each test case will lead the software under test to failure or success, and let

$$\Pr\{observing\ failures\ by\ test\ cases\ from\ D_i\} \equiv \theta_i$$

where $\theta_i \in [0, 1]$ and $i = 1, 2, \ldots, m$. After executing one test case, the test oracle should be utilized to verify the behavior of the software under test. As this study focuses on the performance of different reliability assessment strategies, a proper test oracle is assumed available in this study.

7. All actions or distinct software tests are admissible each time. Although there is some argument that this assumption cannot hold for some systems, the key point lies in how one test is defined. For example, in a flight control system, a test can be defined as a maneuver, e.g., rudder rotation. In this case, some test can actually affect the following maneuvers. If the rudder has been positively rotated to the maximum angle, it is forbidden to conduct some maneuver, e.g., positive rotation. Thus, the conduction of some test can rely on the previous one. However, if a test is defined as completing a task, e.g., a "cobra" maneuver, things are different. When the "cobra" maneuver is accomplished, the status of aircraft will be the same as that before the maneuver. Therefore, the next test should be independent of the previous one.

The above assumptions should be reasonable in many mission-critical systems. Let us consider GPS system on an aircraft. Before GPS system is deployed on an aircraft, the reliability of this system should be first evaluated on the emulator. During the assessment process on the emulator, the system will be fixed without modifications. Since the system takes the GPS information received by an aircraft and the correction data from the GBAS system as inputs, the input domain can be determined by factors that affect these two inputs. The GPS information obtained by an aircraft can be affected by the positions of the aircraft and the satellites, the atmosphere condition as well as the performance of GPS receiver. On the other hand, the correction data from GBAS system are supposed to be independent of the GPS information received by an aircraft. Therefore, how these factors vary in the actual environment can be investigated. After that, the input domain can be divided into subdomains according to some input factor. For example, the division can be conducted according to the atmosphere condition. As the GBAS system usually provides its service around the airport area (approximately a 20-30 mile radius), the atmosphere condition around the airport over a year can be recorded, statistically analyzed and categorized. Thus, the subdomains can be created based on the categories of atmosphere condition. Within each subdomain, the test cases can also be further generated according to various choices of other input factors, e.g., positions of the aircraft and the satellites, and types of receivers. Based on the statistical analysis of the atmosphere condition, the operational profile can be evaluated according to the occurrence probabilities of different atmosphere condition categories. Since GPS system discretely corrects its position and calculates the protection level, thus, the output of the system should be independent of the history as the correction and calculation only depends on the current real-time inputs. Usually, the total number of tests during the assessment process is predefined before the assessment. During the assessment process, the expected behavior of GPS system, which serves as the test oracle, can be determined by the tester since the assessment is conducted in a controllable environment where various inputs can be generated for testing.

2.2 Optimal Allocation Solution

Based on above assumptions, the software reliability should be $R = \sum_{i=1}^{m} p_i(1 - \theta_i)$. After the assessment, the testing data can be collected to assess the reliability. Let z_{ij} represents the execution result of the jth test case from subdomain D_i, i.e.,

$$z_{ij} = \begin{cases} 1 & if\ a\ failure\ is\ observed \\ 0 & if\ no\ failure\ is\ observed. \end{cases}$$

Thus, an unbiased point estimator of θ_i is $\hat{\theta}_i = \sum_{j=1}^{n_i} \frac{z_{ij}}{n_i}$.

Accordingly, the unbiased estimator of software reliability is

$$R_E = 1 - \sum_{i=1}^{m} p_i \hat{\theta}_i = 1 - \sum_{i=1}^{m} \sum_{j=1}^{n_i} p_i \frac{z_{ij}}{n_i}.$$

Note that, $\{z_{ij}\}$ is a series of independent random variables according to assumption 4. Based on assumption 3, it can be inferred that $z_{ij} = 1$ should be observed with probability θ_i. Therefore, z_{ij} follows a binominal distribution with parameter θ_i. The mathematical expectation and variance of above estimator are

$$E(R_E) = 1 - \sum_{i=1}^{m} p_i \theta_i, \tag{1}$$

$$Var(R_E) = \sum_{i=1}^{m} p_i^2 \frac{\theta_i(1-\theta_i)}{n_i}. \tag{2}$$

In order to explore the asymptotic behavior of AT, a globally optimal solution that minimizes (2) should be investigated. There are several methods to solve this problem [14], [15]. Maxim and Weed [14] used a first order Taylor series approximation to the system reliability R in order to get an approximation of its variance. Al-Maati and Rekab [15] considered a problem that is similar to the globally optimal problem in this work but their model has only two subdomains. In this study, the variance of reliability estimator is formulated based on the operational profile, which is not explicitly described in [14]. Moreover, compared with [15], a more generalized Optimal Allocation solution will be presented.

In order to solve this OA problem, one important premise is that all the subdomain failure rates are known. This might be unrealistic, because the reliability assessment will make no sense if the failure rates are known in advance. Note that, this premise is just for mathematical tractability. In this study, the OA solution only provides the theoretical minimum variance of the reliability estimator as an upper bound for the asymptotic performance of the AT strategies.

According to (1), the estimator is an unbiased one. Thus, minimizing (2) will provide an unbiased estimator with minimum variance, that is, the OA solution. In this study, the OA solution can be obtained by the Lagrange multiplier method.

Note that, the test resources are limited, that is,

$$g(n_1, n_2, \ldots, n_m) = \sum_{i=1}^{m} n_i - t \le 0.$$

In order to get the minimum value of (2), a variable λ called a Lagrange multiplier is introduced as follows

$$\begin{aligned}\Phi(n_1, n_2, \ldots, n_m, \lambda) &= Var(R_E) + \lambda g(n_1, n_2, \ldots, n_m), \\ &= \sum_{i=1}^{m} p_i^2 \frac{\theta_i(1-\theta_i)}{n_i} + \lambda \left(\sum_{i=1}^{m} n_i - t \right).\end{aligned}$$

One approach to solve above formula is to differentiate it by considering n_i as a continuous variable as follows

$$\begin{cases} \dfrac{\partial \Phi}{\partial n_i} = -p_i^2 \dfrac{\theta_i(1-\theta_i)}{n_i^2} + \lambda = 0 \\ \dfrac{\partial \Phi}{\partial \lambda} = \sum\limits_{i=1}^{m} n_i - t = 0 \end{cases} \quad i = 1, 2, \ldots, m$$

Thus, the OA solution is determined by the following equations

$$\begin{aligned} n_i &= \frac{\sqrt{p_i^2 \theta_i(1-\theta_i)}}{\sum_{j=1}^{m} \sqrt{p_j^2 \theta_j(1-\theta_j)}} \cdot t, \\ \lambda &= \frac{\left(\sum_{j=1}^{m} \sqrt{p_j^2 \theta_j(1-\theta_j)} \right)^2}{t^2}, \\ \min(Var(R_E)) &= \frac{\left(\sum_{j=1}^{m} \sqrt{p_j^2 \theta_j(1-\theta_j)} \right)^2}{t}. \end{aligned}$$

According to above solution, the relationship between n_i and n_j in OA can be obtained as follows

$$\frac{n_i}{n_j} = \frac{\sqrt{p_i^2 \theta_i(1-\theta_i)}}{\sqrt{p_j^2 \theta_j(1-\theta_j)}}. \tag{3}$$

Equation (3) describes how the tests are allocated to different subdomains in the OA solution, which facilitates the following analysis of AT strategies.

3 Adaptive Solution with Gradient Descent Method

Note that, the OA solution presented in Section 2 might not be directly utilized in practice, since it needs the true failure rates to allocate the tests. Thus, a natural idea is to estimate the failure rates and substitute the true failure rates with the estimates to implement the OA solution, e.g., two-stage design [15]. However, the deviation between the estimates and the true values in the first stage will affect the optimality of testing resource allocation scheme in the second stage. Moreover, the estimation accuracy might be unpredictable since the length of the first stage is set based on experience. In fact, according to the strong law of large number, the estimate will tend to be closer to the true value when more tests are conducted. Therefore, dynamically adjusting the testing strategy during the assessment process might be plausible. If such a testing strategy is properly constructed, e.g., AT for reliability assessment, the estimator variance can be lowered down.

3.1 Adaptive Testing Strategy

AT is a software testing technique that results from the application of feedback and adaptive control principles in software testing. As shown in Fig. 1 (proposed in [9]), there are two feedback loops in the AT strategy. The first feedback loop constitutes the software under test, the database (history of testing data) and the testing strategy, where the history of testing data is used to generate the next test case by a given testing policy or test adequacy criterion. The second

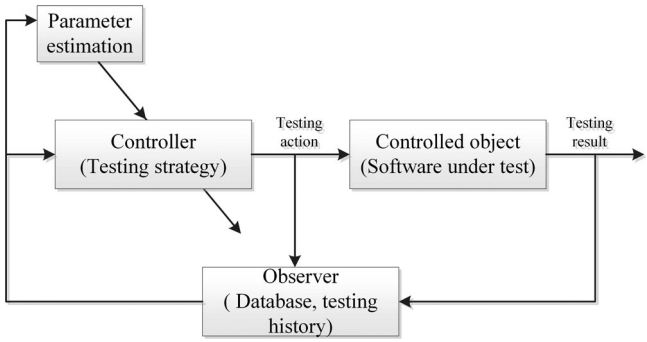

Fig. 1. Diagram of adaptive testing.

feedback loop constitutes the software under test, the database, the parameter estimation scheme and the testing strategy, where the history of testing data is used to improve or change the underlying testing policy or test adequacy criterion. The improvement may lead the random testing to switch from one test distribution (e.g., uniform distribution) to another test distribution (e.g., non-uniform distribution). It may lead partition testing to refine the partitioning of the input domain of the software under test. It may also lead data flow testing to perform boundary value testing. At the beginning of software testing, the software tester has limited knowledge of the software under test and the capability of the test suite. As testing proceeds, the understanding of the software under test as well as that of the test suite is improved, and thus the testing strategy should be improved. In this study, the improvement leads to better test case selection scheme due to better failure rate estimation.

The AT strategy for reliability assessment is constructed based on the optimal solution proposed in [10]. In order to obtain the optimal solution, software states are defined in terms of the numbers of applied tests and observed failures, whereas the ordering of applied tests can be ignored. Based on the states, the software under test for reliability assessment can be modeled as a Controlled Markov Chain (CMC) with a special cost structure. In this way, the problem of determining the optimal testing strategy becomes the first-passage problem of the CMC. Based on the theories of CMC [20], it can be concluded that there exists an optimal solution to this problem. Note that, this optimal testing strategy is designed for the testing process, and it is just locally optimal compared with the OA solution. Since the parameters, i.e., true failure rates, required by the optimal testing strategy are seldom known in practice, an AT strategy is proposed by substituting the true failure rates with the corresponding estimates that are updated during the assessment process. Furthermore, the optimal testing strategy is determined by recursive evaluation from the current state to the states where all available testing resources are exhausted, which highly increases the computational complexity. Thus, in practice, the number of following states that are taken into account in the process of recursive evaluation is limited to lower down the computational overhead of AT, and this compromise might lead AT to deviate from its locally optimal design.

Besides, one problem of AT is its asymptotic behavior, that is, whether the AT strategy converges to the OA solution with limited number of following states taken into account in the process of recursive evaluation. Inspired by the OA solution, the asymptotic behavior of AT is discussed in Observation 1.

Observation 1. Previous AT strategy might not converge to the OA solution with all possible numbers of following states taken into account in the process of recursive evaluation.

Explanation. *See Appendix A.*

Note that, one advantage of AT is that AT is not only a strategy but also a framework, which means that the implementation of components in AT can be various. In fact, the parameter estimation methods in AT can vary, and the modeling of software under test can be modified. Due to this advantage, the asymptotic optimality problem can be investigated by a new AT strategy with different test case selection scheme. In this study, this new AT strategy based on gradient descent method, namely AT-GD, is proposed. On one hand, this AT-GD strategy is a locally optimal solution that accomplishes the task of choosing the next test case to minimize the variance of reliability estimator; on the other hand, the asymptotic optimality of AT-GD can also be guaranteed.

3.2 AT-GD

Gradient descent method is a first-order optimization algorithm, which aims to find a local minimum value of a function. Usually, this approach makes choices based on the negative of the gradient at current state.

3.2.1 Test Case Selection Scheme

Inspired by the gradient descent method, the test case selection scheme for AT-GD is constructed. First, the gradient of the variance estimator function is calculated as follows:

$$-\frac{\partial Var(R_E)}{\partial n_i} = \frac{p_i^2 \hat{\theta}_i (1-\hat{\theta}_i)}{n_i^2} \quad i = 1, 2, \ldots, m. \tag{4}$$

These values can be considered as the change of variance if the number of test cases from subdomain D_i fluctuates, that is, how $Var(R_E)$ is reduced when n_i increases as a real number. After that, a greedy algorithm is adopted to select the subdomain D_i that reduces the variance most or maximizes (4).

Note that, in this algorithm, the number of test cases is considered as a continuous variable, which deviates from the discrete attribute of test case number. In fact, when the failure rates adopted in (4) are the true values, AT-GD will be locally optimal. In practice, these failure rates are substituted with the corresponding estimates. In this case, AT-GD can also be locally optimal, because the current estimates can be considered as the "true" failure rates when no further information is retrieved. If the discrete gradient is adopted, both the variance estimate of current state and the one after conducting the next test must be taken into account. Thus, there will be a new set of failure rates estimates after the next test. Since only one set of "true" failure rates can be adopted in the variance estimation to guarantee the local optimality, there will be a decision-making problem. On one hand, taking the current estimates as the "true"

values will ignore the possible results of the next test, and AT-GD will not be locally optimal since the updated estimates should be the "true" value as more information has been retrieved after the next test. On the other hand, choosing the estimates after the next test as the "true" values will make the estimation of the variance at current state implausible, since these "true" values cannot be obtained at the current state. As either choice has its own limitation, thus, the continuous functions are adopted in this study to avoid the above issues.

3.2.2 Parameter Estimation

Since the estimates of failure rates are adopted in the calculation of gradient, the estimation method should be chosen. Will the estimation method affect the asymptotic behavior of AT-GD? In fact, it can be proved in theory that as long as an unbiased parameter estimator is chosen, AT-GD will converge to the OA solution in Section 2.

After choosing the test case selection scheme and the parameter estimation method, the AT-GD strategy can be constructed. Note that, this new AT-GD strategy is locally optimal if only the next test needs to be determined. Besides, the AT-GD strategy also reduces the overhead for decision-making compared with previous AT strategy. In previous AT strategy, CMC indeed helps to construct a locally optimal testing strategy. However, CMC also increases the computational complexity of the decision-making process, which restricts the utilization of AT. With this new AT-GD strategy, the computational complexity can be effectively controlled since only the gradient needs to be calculated in the decision-making.

3.2.3 Asymptotic Analysis

According to Observation 1, previous AT strategy might not converge to the OA solution with all possible numbers of states taken into account in the process of recursive evaluation. Since AT-GD is already a locally optimal strategy, one more question is what the asymptotic behavior of AT-GD can be. According to Theorem 1, AT-GD converges to the OA solution so long as an unbiased parameter estimator is provided.

Theorem 1. $\lim_{t\to\infty}(Var(GAT,t) - Var(Optimal,t)) = 0$, *where t denotes the number of tests dedicated to the assessment process.*

Proof. See Appendix B. □

3.2.4 AT-GD Algorithm

The following process gives a full picture of the new AT-GD strategy:

1. Initialize $n_i = 1 \quad i = 1, 2, \ldots, m$;
2. Select one test case from each subdomain, and execute it against the software under test;
3. Observe the result of the execution of selected test case from each subdomain D_i. If a failure is observed after executing the test case from subdomain i, let $z_{i1} = 1$, else let $z_{i1} = 0$ $i = 1, 2, \ldots, m$;
4. Record the total number of observed failures in subdomain D_i, that is,

$$F_i = \sum_{j=1}^{n_i} Z_{ij} \quad i = 1, 2, \ldots, m;$$

5. Calculate

$$\hat{\theta}_i = \begin{cases} \dfrac{F_i + 1}{n_i + 1} & if\ F_i = 0 \\ \dfrac{F_i}{n_i} & else \end{cases} \qquad i = 1, 2, \ldots, m,$$

which is the estimate of θ_i. The biased parameter estimator $\hat{\theta}_i$ here is to avoid the case where all resources will be allocated to the subdomain that first reveals a failure. Note that, this biased estimator will switch to an unbiased one when a failure is observed. In fact, this biased estimator will not affect the asymptotic behavior of AT-GD since one premise of asymptotic behavior is sufficient testing and there will be failures eventually;

6. Check the values of

$$-\frac{\partial Var(R_E)}{\partial n_i} = \frac{p_i^2 \hat{\theta}_i (1 - \hat{\theta}_i)}{n_i^2} \quad i = 1, 2, \ldots, m,$$

if $\frac{p_i^2 \hat{\theta}_i (1-\hat{\theta}_i)}{(n_i^T)^2} > \frac{p_j^2 \hat{\theta}_j (1-\hat{\theta}_j)}{(n_j^T)^2}$ for $j = 1, 2, \ldots, m \ \& \ j \neq i$, select a test case from subdomain D_i, execute it and record the result. If there exist several subdomains that have the same $-\frac{\partial Var(R_E)}{\partial n_i}$ value, select one randomly or alternatively;

7. Check whether the stopping criterion is satisfied. If not, go back to Step 4, else go to Step 8;
8. Stop testing and evaluate the reliability

$$R_E = 1 - \sum_{i=1}^{m} p_i \hat{\theta}_i;$$

9. End.

3.3 Sensitivity Analysis of AT-GD

Note that, the reliability assessment in this study adopts an operational profile to characterize the field use of the system quantitatively. In practice, determining an operational profile can be difficult and it might introduce some error. Thus, sensitivity analysis is usually adopted to investigate the effect of an error in operational profile on the change of performance. In this study, sensitivity analysis is conducted to investigate the perturbation of reliability estimator variance with respect to an error in operational profile.

At the very beginning of AT-GD (step 2), one test is allocated to each subdomain. Thus, the estimated failure rate for each subdomain is usually the same since it is rare to observe a failure just after the first test when the failure rate is low. According to Step 6, when the estimated failure rates are the same, AT-GD should be the same as Operational Testing (OT), which is a testing strategy with the operational profile as its testing profile. In addition, AT-GD converges to the OA solution eventually. Therefore, the sensitivity of AT-GD should evolve from that of OT to that of OA when an error in the operational profile occurs. In order to determine the detailed sensitivity value on an error in the operational profile, the approach used by Musa [21] is adopted.

Let j be the operation whose probability is in error, and ε_j be the error in probability, that is, $\varepsilon_j = p_{Tj} - p_{Fj}$, where p_{Tj} is the testing operational profile for operation j and p_{Fj} is the true operational profile in field use. Thus, there is $-p_{Fj} < \varepsilon_j < 1 - p_{Fj}$ since probability only varies from zero to one. Let η_j be the relative error, that is, $\eta_j = \varepsilon_j / p_{Fj}$, and thus, there is $-1 \leq \eta_j \leq 1/p_{Fj} - 1$. Since no information for ε_k is known in any other operation, it is assumed that all ε_k are affected in the same relative way so that they have the same relative error η. As the sum of the occurrence probabilities of different operations equals to one, η can be calculated, that is, $\eta = -\eta_j p_{Fj}/(1 - p_{Fj})$. Note that, the sign of η is opposite to that of η_j, which means, "an error in one occurrence probability of the operational profile causes countervailing errors in other occurrence probabilities" [21].

Therefore, the sensitivity S_i of reliability estimator variance on an error in the occurrence probability of operation i can be defined as follows

$$S_i = \frac{\frac{Var_T - Var_F}{Var_F}}{\eta_i}.$$

This sensitivity is not an absolute value as what Musa defined in [21]. This is because neither positive nor negative error in Musa's sensitivity is acceptable, but smaller field variance is preferred in this study. Thus, the acceptable cases are $S_i > 0 \ if \ \eta_i > 0$ and $S_i < 0 \ if \ \eta_i < 0$.

In this case, the analytical solutions of sensitivity in OT and OA can be obtained as follows

$$\begin{aligned}
S_i^{OT} &= F_i^{OT}, \\
S_i^{OA} &= 2F_i^{OA} + \eta_i \cdot (F_i^{OA})^2, \\
F_i^{OT} &= -\frac{p_{Fi}}{1 - p_{Fi}} + \frac{1}{1 - p_{Fi}} \cdot \frac{p_{Fi}\theta_i(1 - \theta_i)}{\sum_{k=1}^{m} p_{Fk}\theta_k(1 - \theta_k)}, \\
F_i^{OA} &= -\frac{p_{Fi}}{1 - p_{Fi}} + \frac{1}{1 - p_{Fi}} \cdot \frac{p_{Fi}\sqrt{\theta_i(1 - \theta_i)}}{\sum_{k=1}^{m} p_{Fk}\sqrt{\theta_k(1 - \theta_k)}},
\end{aligned}$$

where S_i^{OT} denotes the sensitivity of OT when an error occurs in the occurrence probability of operation i and S_i^{OA} denotes the sensitivity of OA when an error occurs in the occurrence probability of operation i.

Note that, F_i^{OA} has the same structure as F_i^{OT}. The difference lies in the latter part of the expression, which can be considered as the contribution of operation j to the total variance of each testing strategy. By MATALB, the values of S_i^{OT} and S_i^{OA} with different F_i^{OT}, F_i^{OA} and η_i can be plotted in Fig. 2. For convenience, the subscript of $S_i^{OT}, S_i^{OA}, F_i^{OT}, F_i^{OA}$ and η_i are ignored in Fig. 2 and the following analysis.

Fig. 2 indicates that OA is more sensitive to the error in operational profile than OT. This is because the OA solution is calculated based on the precise operational profile to minimize the variance, and any deviation in the operational profile will make the solution no longer an optimal one and the variance fluctuates. Recall that, the cases where the field variance is lower than the testing variance are preferred, that is, $S_i > 0 \ if \ \eta_i > 0$ and $S_i < 0 \ if \ \eta_i < 0$ are acceptable. According to Fig. 2, the acceptable areas in OA are

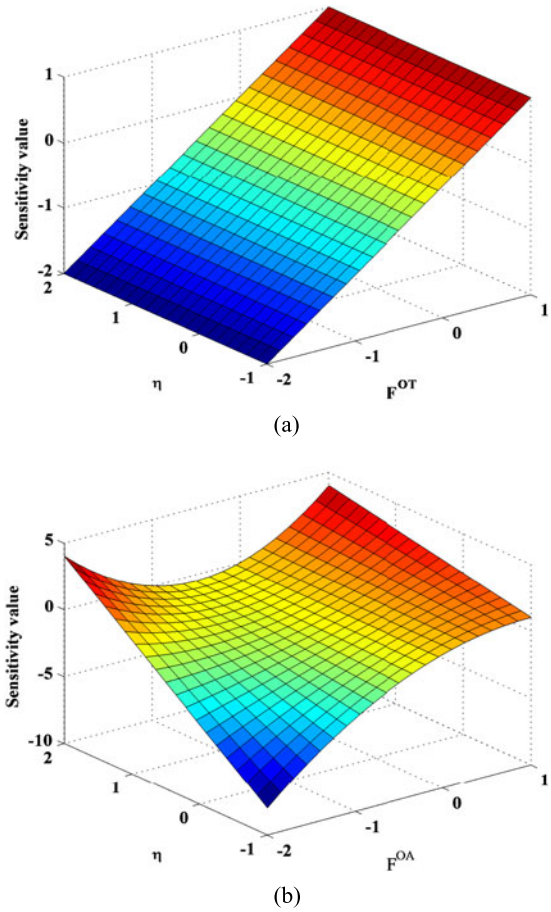

Fig. 2 (a). Sensitivity values of OT. (b) Sensitivity values of OA.

larger than in OT. Besides, in the acceptable areas OA guarantees larger variance reduction than OT. However, in the areas that are not preferred, the increase of variance for OA can usually be larger than that of OT. Note that, AT-GD evolves from OT to OA. Therefore, it can be inferred that the sensitivity of AT-GD should lie between the sensitivity of OT and that of OA.

4 Simulation and Experimental Validation

Note that, the theoretical analysis of AT-GD is conducted based on some assumptions, e.g., considering $Var(R_E)$ as a continuous function of n_i, and the assumptions might not always be reasonable in all mission-critical systems. Thus, in this section, simulation and experiments are set up to validate the effectiveness and efficiency of AT-GD.

4.1 Simulation

The main purpose of simulation is to validate the effectiveness of AT-GD with sufficient testing resources. Under such circumstance, the variance of AT-GD should converge to that of OA. In this simulation, MATLAB is utilized to generate ten subdomains, and each of them is constructed with an operational profile as well as a randomly generated failure rate that lies between 1.0e-07 and 1.0e-05.

During the simulation, the numbers of executed test cases and observed failures in each subdomain are recorded to estimate the failure rate of each subdomain. When the

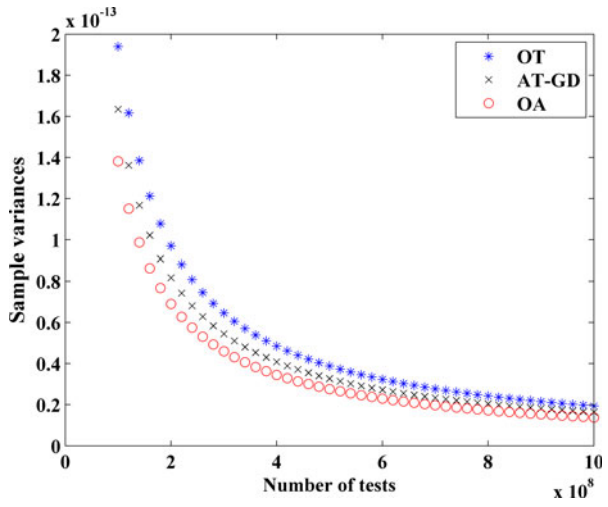

Fig. 3. Sample variances of different assessment strategies.

number of tests reaches 1.0e+09, the testing process is stopped as one trial.

In this simulation, three testing strategies are examined, that is, AT-GD, OA and OT. OT is a common approach for reliability assessment in practice. Besides, as all the failure rates are generated in the simulation, thus the OA strategy can be constructed precisely. Since these three testing strategies are randomized algorithms, thus, the simulation has 200 trials for each testing strategy to reduce possible statistical bias. The sample variances of these trials are calculated to measure the three testing strategies. The results of the sample variances are plotted in Fig. 3.

According to Fig. 3, it can be seen that AT-GD has a lower variance than OT throughout the testing process. Meanwhile, as the testing proceeds, the variance of AT-GD converges to that of OA, which confirms the asymptotic global optimality of AT-GD.

It should be emphasized that the simulation is just set up to validate the results of the theoretical analysis. Usually, only limited testing resources can be available during the assessment process for mission-critical systems. Under such circumstance, the observed failures are few. Therefore, the effectiveness and efficiency of AT-GD should be further validated with experiments on real-life subject programs.

4.2 Experimental Validation

In this section, two real-life subject programs, that is, GCC and gzip, are adopted to evaluate the effectiveness and efficiency of AT-GD.

4.2.1 Subject Programs

Version 3.4.1 of the GNU Compiler Collection (GCC) and six defects extracted from Bugzilla are used. Since all six defects are fixed in the subsequent version, version 3.4.5 of GCC is used as the test oracle with respect to these defects. The same test case is executed on version 3.4.5 and its output is compared with that from version 3.4.1 to determine whether a failure occurs. There are several ways to partition the test suite [19], [22], and a common way is based on functionality, which is adopted in this paper. To this end, we intentionally selected 6,435 test cases from the 8,137 test cases contained in the GCC test suite. Each of these 6,435 test cases is a C/C++ program designed to test some single functionality of GCC. Those test cases that cannot be identified to test some specific functionality of GCC are avoided. The 6,435 test cases are partitioned in to six disjoint subdomains on the basis of the functionalities they are intended to test, containing 624, 766, 729, 537, 1,890, and 1,889 test cases, respectively.

gzip (GNU zip) is a commonly used Unix command line utility developed by GNU. Several previously released versions of gzip are obtained from the Software-artifact Infrastructure Repository (SIR) (http://sir.unl.edu/portal/index.html) together with their test suites and injected defects. gzip is designed to be a replacement for *compress*. Its main advantages over *compress* are much better compression and freedom from patented algorithms. Defects are injected into these subject programs according to the bug history and the codes that are deleted from, inserted into, or modified between the versions. The original gzip test suite provided by SIR is created based on the test specification language (TSL) [22] and it contains 214 test cases, which does not seem sufficient to get a reliability estimate in a statistical sense. The TSL specifications include parameter dimension and environmental dimension. These 214 test cases have covered the parameter dimension but the environmental dimension is only partially covered since the choices for environmental dimension, e.g., input files, can be various when test cases are generated based on TSL specifications. Our intension of enlarging the test suite to comprise 8,653 test cases is to increase the coverage of environmental dimension by providing more choices in test case generation. Then, according to the functionalities determined in the TSL specifications, the test suite is partitioned into seven disjointed subdomains, containing 170, 4,238, 66, 475, 2,118, 1,058, and 528 test cases, respectively. As some defects are easy to detect, which is usually removed before the reliability assessment, only six relatively hard-to-detect defects along with the released versions provided by SIR are chosen and injected into the program.

4.2.2 Testing Strategies and Performance Metrics

In the experimental validation, four different testing strategies are examined: AT-GD, OT, OA and AT:

Adaptive testing strategy with gradient descent method. AT-GD selects the next test case to maximize the variance reduction for the next action and it converges to the OA solution.

Operational testing. OT is a common reliability assessment strategy with an operational profile as the testing profile. That is, the testing resource allocated in some subdomain is proportional to the occurrence probability in operational profile.

Optimal strategy. OA is based on the OA solution that aims to construct an unbiased estimator with minimum variance. As the true failure rates are needed in the OA solution, these true values for the two subject programs in this section are assumed known in advance. In this study, the true failure rates are obtained by executing all available test cases in each subdomain. Thus, the OA strategy can be implemented for the experiments.

Adaptive testing for reliability assessment. This is a previous AT strategy with the CMC modeling the software under

test. Note that, the number of following states that are taken into account in the process of recursive evaluation in this section is five, that is, only the possible testing results of next five tests are considered in the decision-making process of AT. In this experimental validation, the term AT refers to the previous AT strategy proposed in [10] but not the framework of adaptive testing.

As AT-GD aims to minimize the variance of reliability estimator, one possible way of measuring the examined testing strategies is to calculate their reliability estimator variances analytically. However, since it is hard to get the analytical variance of AT and the sample variance is an unbiased estimator of population variance, the sample variance of each testing strategy is adopted for performance measurement. As all four testing strategies are randomized, 1,000 trials are conducted in each experiment for each testing strategy to have a statistical analysis. A trial is defined as running out predefined testing resources for software reliability assessment, which is consuming t tests to get a reliability estimate. Besides, the mean value of reliability estimates in 1,000 trials is also recorded for each testing strategy. Since only limited tests are available for each experiment, there can be cases where no failure is observed for some subdomain. Therefore, an biased estimation of failure rate, i.e., Bayesian estimation with a beta distribution *Beta*(1,1) a priori [23], is adopted. Under this circumstance, the mean squared error (MSE) is adopted in this study. Since both estimation bias and variance of the estimator are involved in MSE, i.e., $MSE(\hat{\theta}) = Var(\hat{\theta}) + (Bias(\hat{\theta}, \theta))^2$, it can provide a fair comparison when the estimators are biased. The mean value (*Mean*), sample variance (*Var*) and *MSE* are calculated as follows:

$$Mean = \frac{1}{1,000}\sum_{i=1}^{1,000}\hat{R}_i,$$

$$Var = \frac{1}{999}\sum_{i=1}^{1,000}\left(\hat{R}_i - \frac{1}{1,000}\sum_{j=1}^{1,000}\hat{R}_j\right)^2,$$

$$MSE = \frac{1}{1,000}\sum_{i=1}^{1,000}(\hat{R}_i - R)^2,$$

where R is the true reliability that can be calculated based on the true failure rates, and $\hat{R}_i$ is the reliability estimate for trial i.

In this study, nine checkpoints for each experiment are set to collect data, that is, 200, 225, 250, 275, ..., 400 tests for each trial. Reliability estimates are recorded at these checkpoints for each testing strategy, and after 1,000 trials, the sample mean, variance and MSE are calculated to investigate the performance of AT-GD, OT, OA and AT.

4.2.3 Operational Profile

The operational profile in field use can vary, and thus, assumptions on that might cause bias. However, not every operational profile can be adopted in the experiments due to time and space limitation. Therefore, three typical operational profiles for each subject program are adopted to validate the effectiveness and efficiency of AT-GD, OT, OA, and AT. These profiles are listed in Table 1.

TABLE 1
Operational Profiles for Experiments

	Profile 1	Profile 2	Profile 3
GCC	0.167,0.167, 0.167,0.167, 0.167,0.165	0.115,0.095,0.115, 0.084,0.297,0.294	0.157,0.136, 0.192,0.128, 0.214,0.173
gzip	0.143,0.143,0.143,0.143,0.143, 0.143,0.142	0.020,0.490,0.008, 0.055,0.245,0.122, 0.061	0.10,0.18,0.12,0.15, 0.15,0.15,0.15

Profile 1 is a profile that the occurrence probability of each subdomain is nearly the same, whereas due to the rounding issue, the profile slightly deviates from the uniform one. Profile 2 emulates a scenario where the usage of a subdomain is proportional to its complexity that can be interpreted in many ways, e.g., the number of test cases in each subdomain in this study. With Profile 3 as the operational profile, the OA solution is allocating the test cases uniformly. Note that, these three profiles cannot represent all the scenarios in filed use and the selected profiles are just some typical or extreme ones in practice.

4.2.4 Experimental Results

The experimental results for GCC and gzip are depicted in Fig. 4.

Figs. 4a and 4b depict the sample means and variances for GCC and gzip, respectively. The left columns of Figs. 4a and 4b denote the means of the reliability estimates and the dashed lines are the true reliability values. As the number of tests is limited, which leads few failures to be observed during the assessment, the mean values of reliability estimates are not so close to the true values. However, when more tests are conducted, the means get closer to the true values. According to the experimental results, the means of AT-GD are closer to the true values than those of AT. Both AT-GD and AT provide more accurate means than OA and OT. The right columns in Figs. 4a and 4b are the sample variances of these four testing strategies. AT-GD and AT outperform OT and OA in almost all the experiments by providing lower sample variance, which indicates that the AT strategies provide more stable estimates than OT and OA when few testing resources are available.

Since the biased estimator is adopted when no failure is observed for one subdomain, MSE is utilized to compare the effectiveness of the testing strategies. Fig. 4c depicts the MSE values for the examined testing strategies. It can be seen that AT-GD has lower MSEs than AT, which means that AT-GD provides more accurate and stable estimates than AT. In addition, both AT-GD and AT outperform OA and OT by providing lower MSE values.

As the four testing strategies are all randomized algorithms, the differences in sample means and variances should be confirmed by statistical test. In this study, the Mann-Whitney U test and Levene's test are utilized to test the significances in the mean value and variance. The Mann-Whitney U test is a nonparametric test and Levene's test is less sensitive to departure from normality. These two tests are selected in order to alleviate the effect of data distribution on the statistical test result

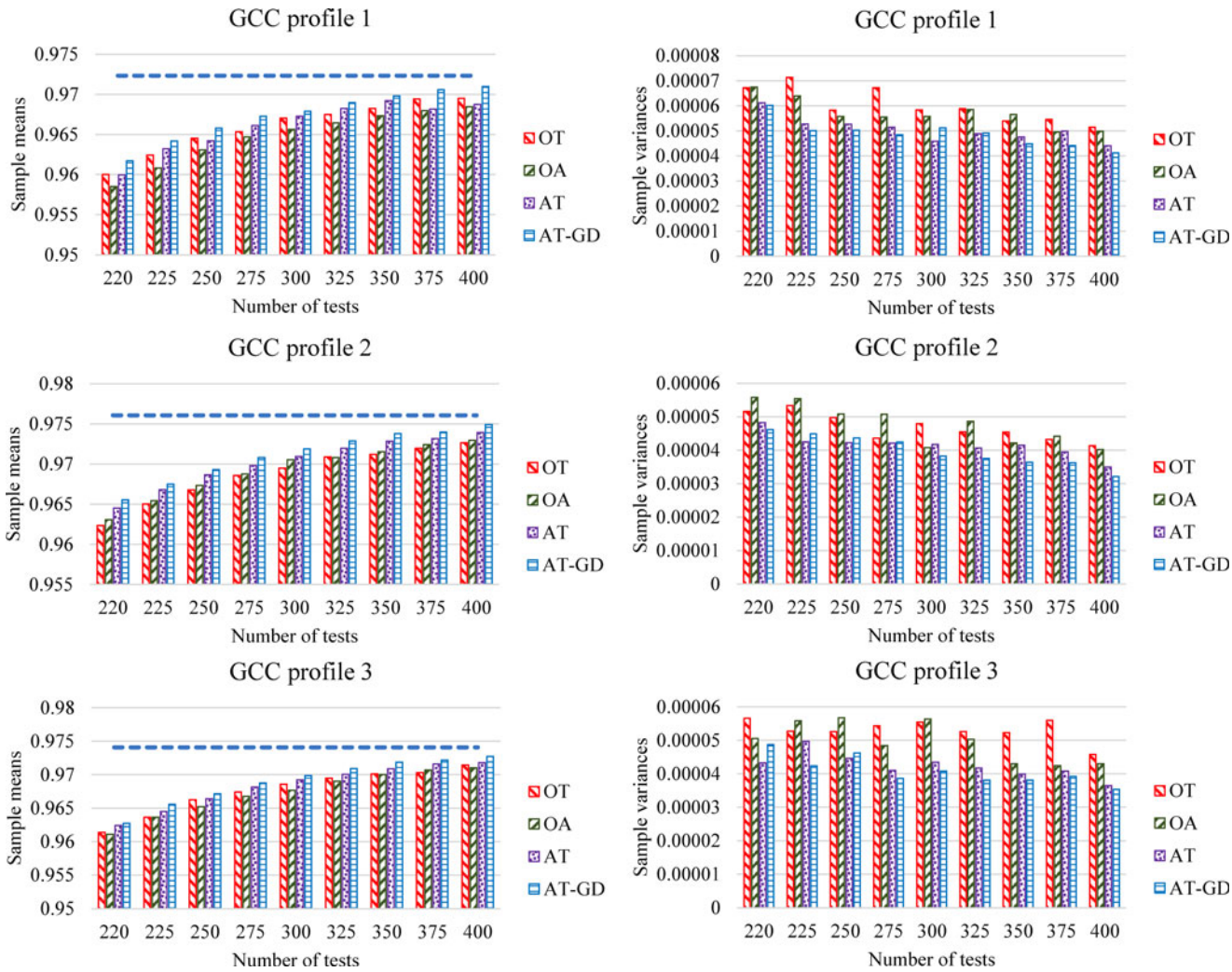

Fig. 4. (a) The sample means and variances for GCC with different operational profiles.

since no evidence indicates the distributions of the experimental results. Due to space limitation, the p values of the statistical tests are not listed separately. Instead, a summary is presented in Table 2. For example, the third row of column 3 in Table 2 is 9/9, which indicates that the mean of AT-GD has significant difference with that of OT for 9 out of 9 checkpoints. Based on the mean values plotted in Fig. 4a, it can be confirmed that AT-GD significantly outperforms OT by providing estimates closer to the true reliability when few test cases are available.

According to the statistical tests and the corresponding sample values in Fig. 4, it can be seen that the advantages of AT-GD and AT over OT and OA can be statistically confirmed. Besides, AT-GD can significantly improve the accuracy of reliability estimates compared with AT, meanwhile, AT-GD's advantage over AT on providing more stable reliability estimates are also confirmed by the statistical test results.

Note that, according to the experimental results, OA did not show its advantage over OT on lowering down estimator variance. Although OA has theoretically lower variance than OT, yet the difference cannot be confirmed by the results of 1,000 trials in most experiment scenarios. In fact, the biased estimation of failure rates might affect the performance of OA since OA has minimum variance with the premise of unbiased the estimation. However, it should be noted that when no failure is observed for some operation, it is better to estimate the failure rate with a biased estimator rather than to be zero. Under such circumstance, it seems that there is not enough evidence to guarantee that OA has better performance than OT with limited testing resources being available.

In addition, since the computational overhead is one obstacle that might prevent AT from being utilized more widely in practice, the cost incurred in decision-making of AT-GD should be analyzed. According to the recursive evaluation process in AT, it can be inferred that the variances of the next $2m$ states need to be calculated in order to obtain the variance of each current state. Thus, if the number of following states that are taken into account in the process of the recursive evaluation is n, the total cost incurred decision-making should be $\mathrm{O}(m^n)$. As for AT-GD, only the gradient needs to be calculated to decide the next test, thus, the computational overhead should be $\mathrm{O}(m)$. Under this circumstance, the computational

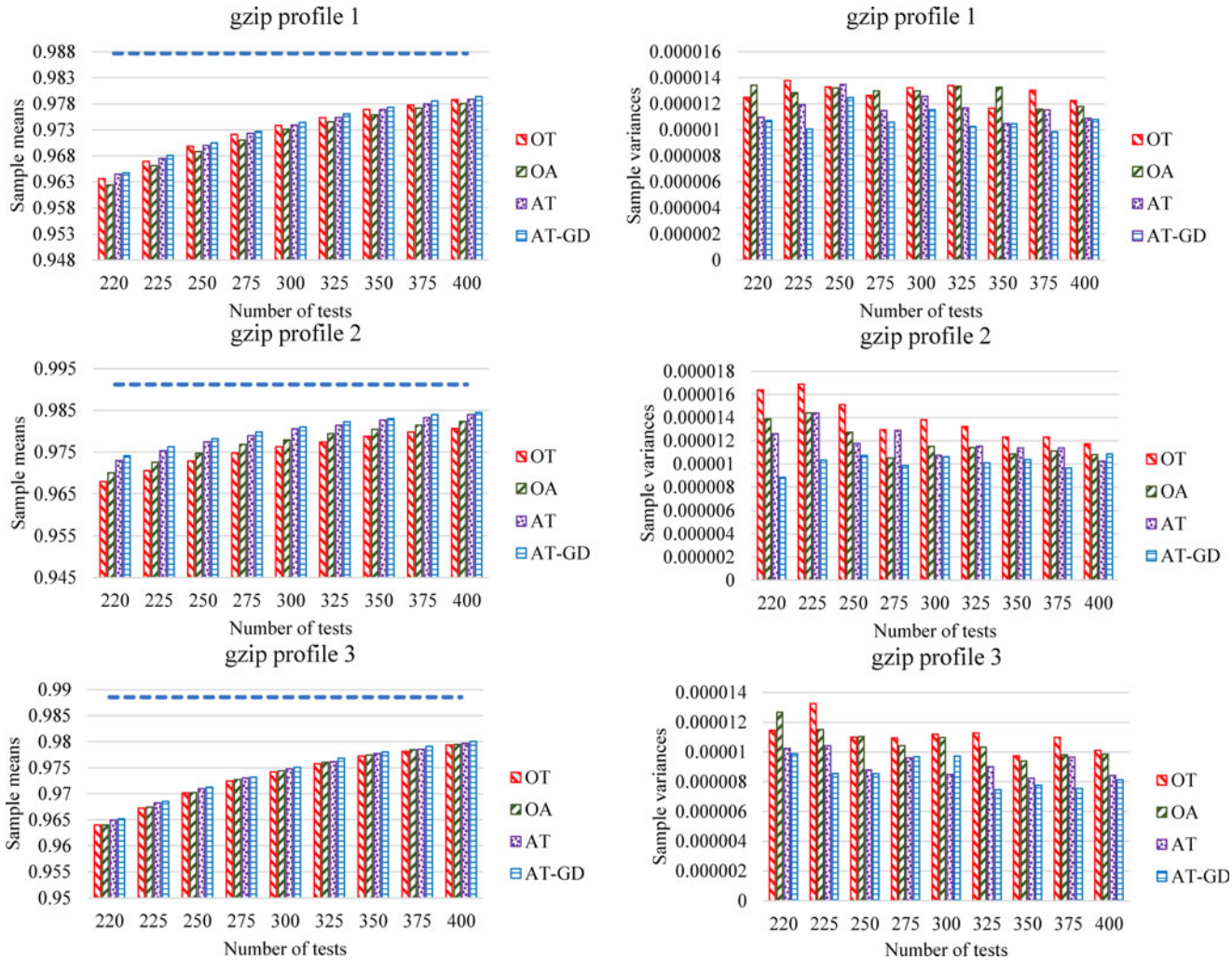

Fig. 4. (b) The sample means and variances for gzip with different operational profiles.

complexity of AT-GD is more controllable compared with that of AT. Besides, the averaged time cost incurred for decision-making in each testing strategy is recorded. Since this overhead should be independent of the operational profile and it does so according to the experimental results, only the experimental results with Profile 1 are tabulated in Table 3 to save space. Note that, this recorded time just represents the average time needed to decide the next test, and it does not include the time cost incurred for executing the test case.

Experimental results in Table 3 indicate that the computational overhead of AT-GD, OT and OA can have the same order of magnitude, whereas AT has a much higher cost incurred in decision-making. Based on the entire experimental results that examine both the effectiveness and efficiency of the testing strategies, AT-GD are preferred compared with the other three testing strategies, since AT-GD can improve the effectiveness of reliability estimation compared with OT, OA and even AT, meanwhile, the cost incurred in decision-making of AT-GD is acceptable.

In this section, the simulation results confirmed the asymptotic optimality of AT-GD. In addition, the experimental results shed light on the probability of adopting AT-GD for the reliability assessment of mission-critical systems where the testing resources are limited. The advantages of AT-GD over other testing strategies when few testing resources are available can be attributed to the locally optimal design in AT-GD. Based on the above results, more confidence can be obtained on the utilization of AT-GD in practice.

4.2.5 *Threats to Validity*

The followings are several potential threats to the validity with respect to the experiments, and provide details on how these threats are addressed.

1. Threats to *construct validity* in this study are on how the performance of a testing strategy is defined. The primary focus is on the variance minimization and decision-making efficiency. In addition, since there are few observed failures in the experiments, the estimates cannot be that accurate and even biased due to the Bayesian estimation. Therefore, besides the sample variance, the mean value of reliability estimates as well as the MSE of each testing strategy is recorded and compared in this study. Based on these three metrics and the time cost incurred for decision-making, a cost/benefit analysis can be carried out to compare the effectiveness and efficiency of the testing strategies.

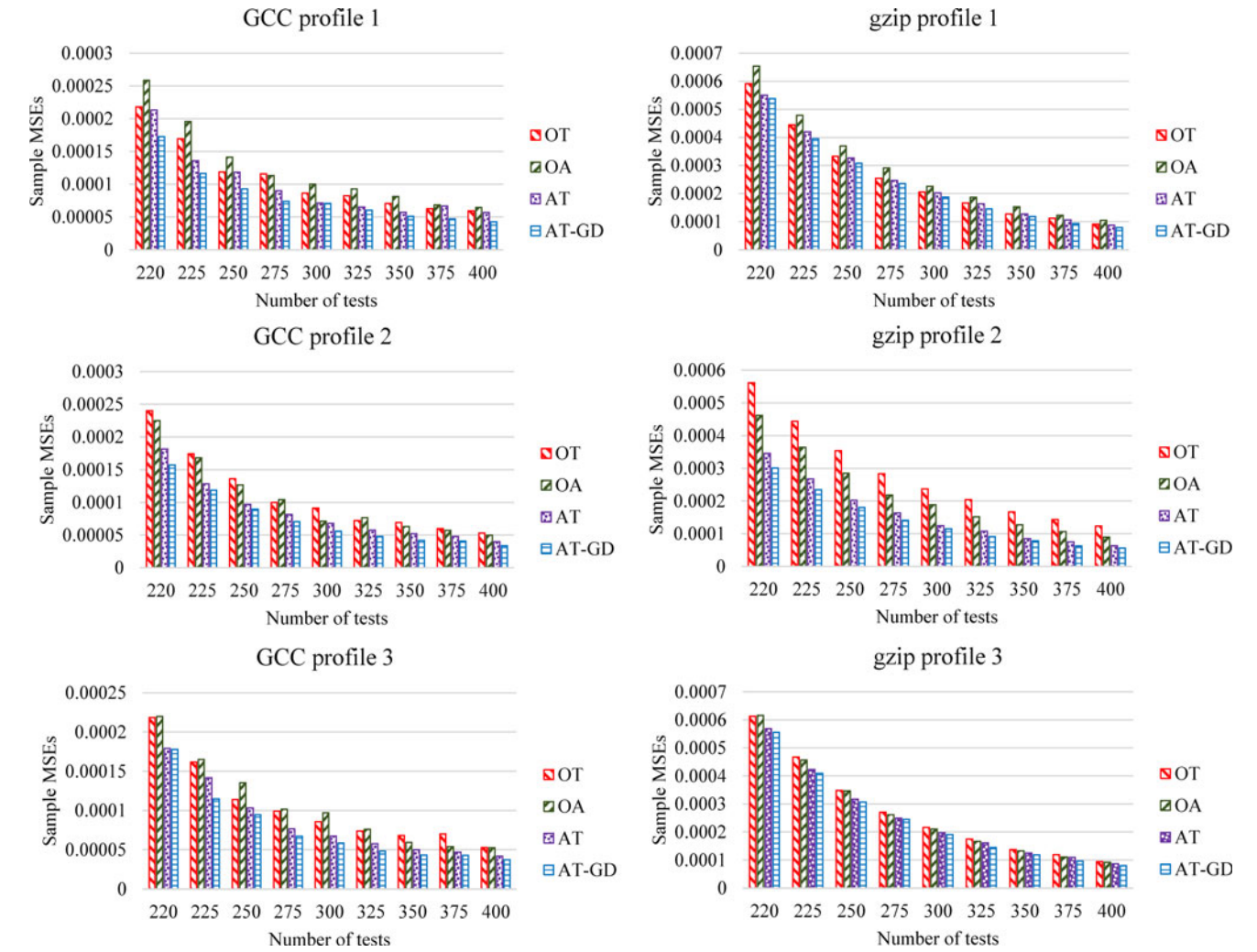

Fig. 4. (c) The sample MSEs for GCC and gzip with different operational profiles.

TABLE 2
Statistical Test Results of Variance and Mean

Scenarios		Variance			Mean		
		AT-GD	AT	OA	AT-GD	AT	OA
GCC profile 1	OT	9/9	7/9	2/9	9/9	6/9	9/9
	OA	5/9	6/9	-	9/9	7/9	-
	AT	1/9	-	-	9/9	-	-
GCC profile 2	OT	8/9	6/9	3/9	9/9	9/9	3/9
	OA	7/9	6/9	-	9/9	8/9	-
	AT	0/9	-	-	9/9	-	-
GCC profile 3	OT	9/9	8/9	3/9	9/9	5/9	5/9
	OA	6/9	6/9	-	9/9	9/9	-
	AT	1/9	-	-	7/9	-	-
gzip profile 1	OT	7/9	2/9	4/9	9/9	9/9	9/9
	OA	6/9	3/9	-	9/9	9/9	-
	AT	4/9	-	-	9/9	-	-
gzip profile 2	OT	9/9	6/9	7/9	9/9	9/9	9/9
	OA	4/9	1/9	-	9/9	9/9	-
	AT	5/9	-	-	9/9	-	-
gzip profile 3	OT	8/9	3/9	0/9	9/9	6/9	9/9
	OA	6/9	6/9	-	9/9	9/9	-
	AT	4/9	-	-	9/9	-	-

2. Threats to *internal validity* may come from how the experiments are designed without bias. In this study, all the experiments are automatically conducted by own-developed testing platform with the same subject programs and test suites. In order to avoid possible defects in the testing platform itself, the platform as well as the implemented testing strategies has been carefully tested. For each testing strategy, 1,000 trails are conducted in each experiment to get enough data for performance comparison. In order to avoid possible statistical bias during the comparison, the Mann-Whitney U test and Levene's test are utilized to test the significances in mean and

TABLE 3
Averaged Time Cost Incurred in Decision-Making in Different Testing Strategies (Seconds)

	AT-GD	AT	OT	OA
GCC	5.75*E-5	9.50	1.52*E-5	1.52*E-5
gzip	6.00*E-4	22.05	1.53*E-5	1.53*E-5

variance, respectively. These statistical tests are independent of/less sensitive to the data distribution, thus, the assumption on the data distribution can be avoided. In fact, not all the experimental data can pass the normal distribution test according to our statistical test results that are not listed in this study due to space limitation.

3. The *external validity* for any empirical study is the generalization of the obtained results. The AT-GD strategy aims to provide more accurate and stable reliability estimates, and it is supposed to be utilized in the reliability assessment of key systems, e.g., mission-critical systems. However, in the experimental validation, only two open source programs are chosen. It should be noticed that although AT-GD can be utilized in mission-critical systems, yet it is not limited to such systems. The asymptotic convergence of AT-GD can guarantee its performance when sufficient testing resources are provided. Meanwhile, the experimental results have also confirmed the effectiveness and efficiency of AT-GD for the cases where limited testing resources are available. As mentioned in Section 2.1, the assumptions proposed in this paper can be reasonable in many systems, e.g., GPS system on an aircraft. It should be noted that the verification of the rationality of assumptions in mission-critical systems could take much effort. For example, in GPS system, the input factors should first be confirmed before the division on the input domain. In order to obtain the operational profile, much effort is needed since the data of atmosphere condition should be collected and statistically analyzed in order to make a proper division of the input domain before constructing the operational profile.

Note that, the generalization of the obtained results should depend on the attributes of the system. For systems that are similar to GPS system on an aircraft and the subject programs in this study, the generalization can be more convincing but still care should be taken before the assumptions are verified to be reasonable in the system under assessment.

As with all experimental studies, further experiments are needed to replicate the results. However, both the theoretical analysis and the experimental results show that AT-GD strategy can improve the testing effectiveness without incurring too much computational overhead, which brings more confidence on wider application of AT-GD strategy.

5 Related Work

This section recalls several strategies on reliability testing and assessment, which are related to the proposed AT-GD strategy in this study.

First, this work is related to the studies on approaches for improving the software reliability.

The purposes of software testing can be divided into two categories by the purposes of testing, that is, for reliability improvement and for reliability assessment. During the reliability improvement process, the test cases are executed against the software under test to detect and remove defects.

Cotroneo et al. [24] proposed a method to select testing techniques according to the features of the software to test. The method is based on two basic steps: first, models are empirically constructed to characterize the software to test in terms of fault types that it is more prone to contain; second, the testing techniques are characterized with respect to types of fault that are more prone to be detected by these testing techniques. Based on such steps, the goal of maximizing the effectiveness of testing process can be achieved. In [25], Cotroneo et al. addressed the challenge of reliability-driven testing, i.e., of testing software systems with the objective of increasing its operational reliability. Consequently, a new testing strategy that combines operational testing with a debug testing approach is proposed to improve delivered reliability.

The above strategies mainly aim to test the software for improving its reliability. By constructing a proper testing strategy, the testing effectiveness can be improved so that more defects can be detected and removed in order to improve the software reliability. In this study, different testing strategies are examined and compared with respect to their abilities to deliver a better estimate of reliability. The proposed AT-GD also improves the testing effectiveness to deliver more accurate and stable reliability estimate. However, during the reliability assessment process, the software under test is fixed without modifications.

Second, this paper is also related to the studies on solving the problem of testing resource allocation that aims to achieve the predefined system requirements, such as, reliability requirement with lowest costs.

In order to guarantee the system a required reliability level with minimum verification costs, Pietrantuono et al. [26] proposed an architecture-based optimization model to allocate the testing resources to different system components. This model can provide flexible solutions according to the information provided by the user. In their work, an architecture-based model (specifically, a DTMC) was constructed to describe the software architecture. In [27], two kinds of software testing-resource allocation problems are considered by Huang and Lyu. The first is to minimize the number of remaining faults given a fixed amount of testing-effort and a reliability objective. The second problem is to minimize the amount of testing effort given the number of remaining faults and a reliability objective. Several strategies for module testing are proposed to help software project managers solve these two problems, and make the best decisions. In [28], Wang et al. suggested solving optimal testing resource allocation problems with Multi-Objective Evolutionary Algorithms (MOEAs). Specifically, they formulated optimal testing resource allocation problems as two types of multi-objective problems. In the first problem, the reliability of the system and the testing cost are considered as two objectives. In the second problem, the total consumed testing resource is taken into account as the third objective.

Both this work and the studies on solving the testing resources allocation problem investigate how to allocate the testing resources optimally. However, the AT-GD focuses on reliability assessment to deliver more accurate and stable reliability estimates, whereas the studies on solving the

problem of testing resource allocation aim to meet the predefined system requirements, such as, reliability requirement, with lowest cost.

This study is also related to the strategies that focus on reliability assessment.

Maxim and Weed [14] concerned the problem of determining the optimal allocation of test effort among individual components of a system. By using knowledge of the relationship between component uncertainty/test costs and system uncertainty/test costs, the test allocation was determined to minimize the variance of an estimator of the system reliability. The optimal allocation solutions for a series system and a parallel system were examined as special cases. The sensitivity of the optimal allocation was investigated with respect to different system configurations. Given a fixed number of test cases, Al-Maati and Rekab [15] aimed to determine how to allocate these test cases among the partitions of the software to minimize the variance incurred by the maximum likelihood estimator of the software reliability. Therefore, a two-stage sampling model was proposed to make allocation decisions during the assessment process.

Our work shares a similar idea with the above two studies when solving the OA problem. However, as mentioned in Section 3, the OA solution in this study is utilized just to examine the asymptotic behavior of the AT-GD strategy, and it is not involved in the decision-making process of AT-GD. Besides, the AT-GD strategy decides the next test after each test, which makes it different from stage-based strategy where a batch of tests is determined and conducted without adjustment until the execution of this batch of tests has been accomplished.

There are also studies on the strategies for the reliability assessment of the series-parallel systems, i.e., reliability sequential sampling (R-SS) [29], [30], [31]. Rekab [29] aimed to estimate the reliability of a series system when the component reliabilities are unknown. Thus, a scheme called "reliability sequential sampling" was proposed and compared with both balanced allocation and optimal allocation. Benkamra et al. [30] extended Rekab's work by proposing an efficient sampling scheme to estimate the reliability of series-parallel systems. Then, they investigated the R-SS for parallel system and even for a parallel–series system, and showed the first order asymptotic optimality of their approach. Benkamra et al. [31] also gave a risk-averse solution based on Bayesian analysis to the problem of estimating the reliability of a parallel-series system.

This work shares similar spirit with the R-SS scheme, that is, the next test case is selected right after the execution of each test case and the needed parameters are dynamically updated during the assessment process. The difference lies in that the AT-GD approach is an instance of adaptive testing in the context of software cybernetics, which aims to explore the interplay between software theory/engineering and control theory/engineering. Specifically, AT-GD provides an explicit function based on the gradient descent method as a test case selection scheme, which guarantees both the local and asymptotic global optimality of AT-GD. However, R-SS provides an implicit function to get a global optimization goal without explicitly explaining the physical significance the function. Thus, no local optimality issue is discussed in the above R-SS studies. Moreover, the R-SS scheme aims to provide a solution for estimating the reliability of series-parallel systems, whereas this work pays more attention on operational-profile based reliability assessment of mission-critical systems.

This work is also related to the studies that aim to solve software-testing problems under the guidance of control theory. Cai et al. [12] proposed an adaptive testing strategy that uses the testing data collected on-line to estimate the required parameters and selects next test cases with testing resources constraints. To further validate the effectiveness of AT for software reliability assessment, an experimental study is conducted [11]. Both [32] and this paper adopt gradient method, but the purposes are quite different: in [32] the gradient method is used for online parameter estimation, whereas in this paper gradient descent method is used directly for test case selection. In [33], Cangussu et al. modeled the system test phase of the software life cycle. Their approach is based on concepts and techniques from control theory and is shown to be useful in computing the effort required to reduce the number of errors and the schedule slippage under a changing process environment. In [34], the sensitivity analysis of the state variable model proposed in [33] was investigated. All the above studies, including this work, try to provide better solutions for software testing problems by utilizing methods and principles adopted in control theory.

6 Conclusions and Future Work

In this study, a new AT strategy based on gradient descent method, namely AT-GD, is proposed. This new AT-GD strategy enhances the previous AT strategy such that both local and asymptotic global optimality are guaranteed. Specifically, a local optimization scheme is adopted in AT-GD as the test case selection scheme and this leads to a desirable result that AT-GD can converge to globally optimal solution as the assessment proceeds. Compared with the previous AT strategy, the computational overhead of AT-GD is also reduced to an acceptable level.

Both simulation and experiments of real-life software systems are set up to validate the effectiveness and efficiency of AT-GD. The simulation results confirm the asymptotic global optimality of AT-GD. In those experiments with limited testing resources being available, it can be seen that AT-GD and previous AT strategy outperform OT and OA by providing more accurate and stable reliability estimates. It can also be observed that AT-GD provides more accurate and stable reliability estimates than previous AT strategy. The time cost incurred in decision-making of AT-GD can even be of the same order of magnitude with those incurred by OT and OA.

Future studies on this work should include further validation on the effectiveness of AT-GD. More specifically, more real-life mission-critical systems should be involved in the validation; besides, how to relax the assumptions adopted in this study should also be further investigated to improve the flexibility of the testing framework.

APPENDIX A

Observation 1. Previous AT strategy might not converge to the OA solution with all possible numbers of following states taken into account in the process of recursive evaluation.

Explanation. Let us show one example to confirm this observation. In previous AT strategy [11], the next test is chosen according to the results of recursive evaluation on the following states. That is, the next test that will be conducted is determined as follows

$$\begin{aligned}&\arg\min(v(x,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m))\\&=\arg\min_{1\le i\le m}(\theta_i v(x-1,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_i+1,\\&\quad Y_i+1,\ldots,\eta_m,Y_m)\\&\quad+(1-\theta_i)v(x-1,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_i+1,Y_i,\ldots,\eta_m,Y_m)).\end{aligned}$$

If all the following states are taken into account in the process of recursive evaluation, there is

$$v(0,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m)=\sum_{i=1}^{m}\frac{p_i^2Y_i(\eta_i-Y_i)}{(\eta_i-1)\eta_i^2}.$$

And if the number of following states taken into account in the process of recursive evaluation is l, there is

$$v(x-l,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m)=\sum_{i=1}^{m}\frac{p_i^2Y_i(\eta_i-Y_i)}{(\eta_i-1)\eta_i^2}.$$

Note that, the purpose of the asymptotic analysis is to find out the test distribution among different operations when the testing number is large enough. Under this circumstance, the estimated failure rates can be as close to the true values as possible. Suppose the number of following states taken into account for the process of recursive evaluation is one. Under such circumstance, the selection of the next test is equivalent to the following strategy

$$\begin{aligned}&\arg\min(v(x,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m))\\&=\arg\max_{1\le i\le m}(v(x,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m)\\&\quad-\theta_i v(x-1,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_i+1,Y_i+1,\ldots,\eta_m,Y_m)\\&\quad-(1-\theta_i)v(x-1,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_i\\&\quad+1,Y_i,\ldots,\eta_m,Y_m))\\&=\arg\max_{1\le i\le m}\Bigg(\sum_{j=1}^{m}\frac{p_j^2Y_j(\eta_j-Y_j)}{(\eta_j-1)\eta_j^2}-\theta_i\sum_{\substack{j=1\\j\ne i}}^{m}\frac{p_j^2Y_j(\eta_j-Y_j)}{(\eta_j-1)\eta_j^2}\\&\quad-\theta_i\frac{p_i^2(Y_i+1)(\eta_i-Y_i)}{\eta_i(\eta_i+1)^2}-(1-\theta_i)\sum_{\substack{j=1\\j\ne i}}^{m}\frac{p_j^2Y_j(\eta_j-Y_j)}{(\eta_j-1)\eta_j^2}\\&\quad-(1-\theta_i)\frac{p_i^2Y_i(\eta_i+1-Y_i)}{\eta_i(\eta_i+1)^2}\Bigg).\end{aligned}$$

Since the failure rates can be accurate enough, thus, the true failure rate θ_i can be considered as Y_i/η_i. In this case, the selection policy should be

$$\begin{aligned}&\arg\min(v(x,\eta_1,Y_1,\eta_2,Y_2,\ldots,\eta_m,Y_m))\\&=\arg\max_{1\le i\le m}\left(\frac{p_i^2(Y_i)(\eta_i-Y_i)}{(\eta_i-1)\eta_i^2}-\left(\theta_i\frac{p_i^2(Y_i+1)(\eta_i-Y_i)}{\eta_i(\eta_i+1)^2}\right.\right.\\&\quad\left.\left.+(1-\theta_i)\frac{p_i^2Y_i(\eta_i+1-Y_i)}{\eta_i(\eta_i+1)^2}\right)\right).\\&=\arg\max_{1\le i\le m}\left(\frac{p_i^2\theta_i(1-\theta_i)}{(\eta_i-1)}-\frac{p_i^2\theta_i(1-\theta_i)(\eta_i+2)}{(\eta_i+1)^2}\right)\\&=\arg\max_{1\le i\le m}\left(p_i^2\theta_i(1-\theta_i)\frac{\eta_i+3}{(\eta_i-1)(\eta_i+1)^2}\right).\end{aligned}$$

The proportion of above strategy can be solved by substituting n_i^2 with $\frac{(\eta_i-1)(\eta_i+1)^2}{\eta_i+3}$ in (4). According to Theorem 1 in Appendix B, previous AT will converge to the following solution

$$\frac{\frac{(\eta_i^t-1)(\eta_i^t+1)^2}{\eta_i^t+3}}{\frac{(\eta_j^t-1)(\eta_j^t+1)^2}{\eta_j^t+3}}\to\frac{p_i^2\theta_i(1-\theta_i)}{p_j^2\theta_j(1-\theta_j)}\quad\text{as } t\to\infty,$$

where t denotes the number of executed test cases. Note that, this solution is different with the OA solution. Thus, this observation is confirmed.

APPENDIX B

Theorem 1. $\lim_{t\to\infty}(Var(GAT,t)-Var(Optimal,t))=0$, *where t denotes the number of tests dedicated to the assessment process.*

Proof. Note that, theorem 1 equals that the variance incurred by AT-GD converges to the variance incurred by OA as $t\to\infty$.

It easily follows that

$$\begin{aligned}&Var(GAT,t)-Var(Optimal,t)\\&=\sum_{i=1}^{m}p_i^2\theta_i(1-\theta_i)\left(\frac{1}{n_i^t}-\frac{1}{n_i}\right).\end{aligned}$$

where n_i^t denotes the number of test cases selected from subdomain D_i by AT-GD when t test cases have been executed.

Denote that $\Delta_i=p_i^2\theta_i(1-\theta_i)(\frac{1}{n_i^t}-\frac{1}{n_i})$. It comes to

$$\begin{aligned}m\cdot\inf_t\{\Delta_i\}\;&<\;Var(GAT,t)-Var(Optimal,t)\\&<\;m\cdot sup_t\{\Delta_i\}\end{aligned}$$

If $\frac{1}{n_i^t}-\frac{1}{n_i}\to 0$ as $t\to\infty$, then there holds

$$\lim_{t\to\infty}(Var(GAT,t)-Var(Optimal,t))=0$$

Actually, $\frac{n_i^t}{n_j^t}\to\frac{\sqrt{p_i^2\theta_i(1-\theta_i)}}{\sqrt{p_j^2\theta_j(1-\theta_j)}}$ guarantees $\frac{1}{n_i^t}-\frac{1}{n_i}\to 0$ as $t\to\infty$. Inspired by [35], the proof is presented as follows.

For an enough large t, there exists

$$t^i = \sup\left\{ l < t : \frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{\left(n_i^l\right)^2} > \frac{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}{\left(n_j^l\right)^2}, \textit{for all } j \neq i \right\},$$

and$t^i \to \infty, n_i^t \to \infty \; as \; t \to \infty$. Then

$$\frac{n_i^t}{n_j^t} \leq \sqrt{\frac{\left(n_i^{t^i}+1\right)^2}{\left(n_j^{t^i}\right)^2}} \leq \sqrt{\left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^i} + \frac{2n_i^{t^i}+1}{\left(n_j^{t^i}\right)^2}}.$$

Thus,

$$\frac{n_i^t}{n_j^t} \leq \sqrt{\left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^i} + \frac{1}{\left(n_j^{t^i}\right)^2} + 2\cdot\frac{\sqrt{\left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^i}}}{n_j^{t^i}}} \tag{B.1}$$

And

$$\frac{n_i^t}{n_j^t} \geq \sqrt{\frac{\left(n_i^{t^j}\right)^2}{\left(n_j^{t^j}+1\right)^2}} \geq \sqrt{\left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^j} - \frac{\left(n_i^{t^j}\right)^2\left(2n_i^{t^j}+1\right)}{\left(n_j^{t^j}\right)^2\left(n_j^{t^j}+1\right)^2}},$$

Thus,

$$\frac{n_i^t}{n_j^t} \geq \sqrt{\left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^j} - \left\{\frac{p_i^2\hat{\theta}_i(1-\hat{\theta}_i)}{p_j^2\hat{\theta}_j(1-\hat{\theta}_j)}\right\}_{t^i}\frac{2n_i^{t^j}+1}{\left(n_j^{t^j}+1\right)^2}} \tag{B.2}$$

Note that, the right part of inequality in (B.1) and (B.2) converges to

$$\frac{\sqrt{p_i^2\theta_i(1-\theta_i)}}{\sqrt{p_j^2\theta_j(1-\theta_j)}}$$

as $t \to \infty$. That is, by the strong law, there is

$$\frac{n_i^t}{n_j^t} \to \frac{\sqrt{p_i^2\theta_i(1-\theta_i)}}{\sqrt{p_j^2\theta_j(1-\theta_j)}} \quad as \; t \to \infty.$$

□

ACKNOWLEDGMENTS

The authors are most grateful to the associate editor and four anonymous reviewers for their constructive and helpful comments on improving the quality of this paper. This work was financially supported by the National Natural Science Foundation of China (Grant Number 61272164).

REFERENCES

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. sixth ed. New York, NY, USA: McGraw-Hill, 2004.

[2] R. V. Binder, *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Reading, MA, USA: Addison-Wesley, 1999.

[3] P. G. Frankl, R. G. Hamlet, B. Littlewood, and L. Strigini, "Evaluating testing methods by delivered reliability," *IEEE Trans. Softw. Eng.*, vol. 24, no. 8, pp. 586–601, Aug. 1998.

[4] B. Beizer, *Software Testing Techniques*. second ed. New York, NY, USA: Van Nostrand, 1990.

[5] A. Podgurski, W. Masri, Y. McCleese, F. G. Wolff, and C. Yang, "Estimation of software reliability by stratified sampling," *ACM Trans. Softw. Eng. Methodology*, vol. 8, no. 3, pp. 263–283, Jul. 1999.

[6] *Handbook of Software Reliability Engineering*, M. R. Lyu, ed., New York, NY, USA: McGraw-Hill, 1996.

[7] J. A. Whittaker and J. H. Poore, "Markov analysis of software specifications," *ACM Trans. Softw. Eng. Methodology*, vol. 2, no. 1, pp. 93–106, Jan. 1993.

[8] H. Pham, *Software System Reliability*. New York, NY, USA: Springer-Verlag, 2006.

[9] K. Y. Cai, "Optimal software testing and adaptive software testing in the context of software cybernetics," *Inform. Softw. Technol.*, vol. 44, no. 14, pp. 841–855, Nov. 2002.

[10] K. Y. Cai, Y. C. Li, and K. Liu, "Optimal and adaptive testing for software reliability assessment," *Inform. Softw. Technol.*, vol. 46, no. 15, pp. 989–1000, Dec. 2004.

[11] K. Y. Cai, C. H. Jiang, H. Hu, and C. G. Bai, "An experimental study of adaptive testing for software reliability assessment," *J. Syst. Softw.*, vol. 81, no. 8, pp. 1406–1429, Aug. 2008.

[12] K. Y. Cai, Y. C. Li, and W. Y. Ning, "Optimal software testing in the setting of controlled Markov chains," *Eur. J. Oper. Res.*, vol. 162, no. 2, pp. 552–579, Apr. 2005.

[13] H. Hu, C. H. Jiang, K. Y. Cai, W. E. Wong, and A. P. Mathur, "Enhancing software reliability estimates using modified adaptive testing," *Inform. Softw. Technol.*, vol. 55, no. 2, pp. 288–300, Feb. 2013.

[14] L. D. Maxim and H. D. Weed, "Allocation of test effort for minimum variance of reliability," *IEEE Trans. Rel.*, vol. R-26, no. 2, pp. 111–115, Jun. 1977.

[15] S. A. Al-Maati and K. Rekab, "Dynamic test allocation model for software reliability," in *Proc. Int. Conf. Quality Softw.*, Nov. 2003, pp. 26–31.

[16] P. Pedregal, *Introduction to Optimization*. New York, NY, USA: Springer-Verlag, 2003.

[17] K. Y. Cai, "Towards a conceptual framework of software run reliability modeling," *Inform. Sciences*, vol. 126, no. 1-4, pp. 137–163, Jul. 2000.

[18] J. D. Musa, "Operational profiles in software-reliability engineering," *IEEE Softw.*, vol. 10, no. 2, pp. 14–32, Mar. 1993.

[19] T. Y. Chen, P. L. Poon, and T. H. Tse, "A choice relation framework for supporting category-partition test case generation," *IEEE Trans. Softw. Eng.*, vol. 29, no. 7, pp. 577–593, Jul. 2003.

[20] C. Derman, *Finite State Markovian Decision Processes*. New York, NY, USA: Academic, 1970.

[21] J. D. Musa, "Sensitivity of field failure intensity to operational profile errors," in *Proc. Int. Symp. Softw. Rel. Eng.*, Nov. 1994, pp. 334–337.

[22] T. J. Ostrand and M. J. Balcer, "The category-partition method for specifying and generating functional tests," *Commun. ACM*, vol. 31, no. 6, pp. 676–686, Jun. 1988.

[23] K. W. Miller, L. J. Morell, R. E. Noonan, S. K. Park, D. M. Nicol, B. W. Murrill, and M. Voas, "Estimating the probability of failure when testing reveals no failures," *IEEE Trans. Softw. Eng.*, vol. 18, no. 1, pp. 33–43, Jan. 1992.

[24] D. Cotroneo, R. Pietrantuono, and S. Russo, "Testing techniques selection based on odc fault types and software metrics," *J. Syst. Softw.*, vol. 86, no. 6, pp. 1613–1637, Jun. 2013.

[25] D. Cotroneo, R. Pietrantuono, and S. Russo, "Combining operational and debug testing for improving reliability," *IEEE Trans. Rel.*, vol. 62, no. 2, pp. 408–423, Jun. 2013.

[26] R. Pietrantuono, S. Russo, and K. S. Trivedi, "Software reliability and testing time allocation: an architecture-based approach," *IEEE Trans. Softw. Eng.*, vol. 36, no. 3, pp. 323–337, May/Jun 2010.

[27] C. Y. Huang and M. R. Lyu, "Optimal testing resource allocation, and sensitivity analysis in software development," *IEEE Trans. Rel.*, vol. 54, no. 4, pp. 592–603, Dec. 2005.

[28] Z. Wang, K. Tang, and X. Yao, "Multi-objective approaches to optimal testing resource allocation in modular software systems," *IEEE Trans. Rel.*, vol. 59, no. 3, pp. 563–575, Sep. 2010.

[29] K. Rekab, "A sampling scheme for estimating the reliability of a series system," *IEEE Trans. Rel.*, vol. 44, no. 2, pp. 287–290, Jun. 1993.
[30] Z. Benkamra, M. Terbecheb, and M. Tlemcani, "Tow stage design for estimating the reliability of series/parallel systems," *Math. Comput. Simul.*, vol. 81, no. 10, pp. 2062–2072, Jun. 2011.
[31] Z. Benkamra, T. Mekki, and T. Mounir, "Bayesian sequential estimation of the reliability of a parallel-series system," *Appl. Math. Comput.*, vol. 219, no. 23, pp. 10842–10852, Aug. 2013.
[32] K. Y. Cai, T. Y. Chen, Y. C. Li, Y. T. Yu, and L. Zhao, "On the online parameter estimation problem in adaptive software testing," *Int. J. Softw. Eng. Knowl.*, vol. 18, no. 3, pp. 357–381, May 2008.
[33] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "A formal model of the software test process," *IEEE Trans. Softw. Eng.*, vol. 28, no. 8, pp. 782–796, Aug. 2002.
[34] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "Using sensitivity analysis to validate a state variable model of the software test process," *IEEE Trans. Softw. Eng*, vol. 29, no. 5, pp. 430–443, May 2003.
[35] K. Rekab, "Asymptotic efficiency in sequential designs for estimation in the exponential family case," *Sequential Analysis: Design Methods and Applications*, vol. 9, no. 3, pp. 305–315, 1990.

Junpeng Lv received the BS degree in 2009 from Beihang University, Beijing, China. He is working toward the doctoral degree in Beihang University. His current research interests include software testing and software reliability assessment.

Bei-Bei Yin received the PhD degrees from Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China, in 2010. She has been a lecturer at Beihang University since 2010. Her main research interests include software testing, software reliability, and software cybernetics.

Kai-Yuan Cai received the BS, MS, and PhD degrees from Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China, in 1984, 1987, and 1991, respectively. He has been a full professor at Beihang University since 1995. He is a Cheung Kong Scholar (chair professor), jointly appointed by the Ministry of Education of China and the Li Ka Shing Foundation of Hong Kong in 1999. His main research interests include software testing, software reliability, reliable flight control, and software cybernetics.