IIE Transactions (1998) **30**, 1143–1151

Economic allocation of test times for subsystem-level reliability growth testing

DAVID W. COIT

Department of Industrial Engineering, Rutgers University, Piscataway, NJ 08854, USA
E-mail: coit@rci.rutgers.edu

Received June 1997 and accepted October 1998

In the development of new electronic systems the planning of reliability growth tests has become both more critical and more difficult as available testing budgets have diminished. Previously, system designers were able to plan and implement relatively lengthy reliability growth test plans to assist in the development of reliable systems. A new method is presented to allocate subsystem reliability growth test time in order to maximize the system mean time between failure (MTBF) or system reliability when designers are confronted with limited testing resources. For certain problems, the algorithm yields the same results as competing approaches to the problem but with significantly fewer required iterations. More significantly, the new algorithm applies to a larger problem domain compared to analogous algorithms. Much more realistic formulations of the problem can now be solved optimally. The algorithm is based on objective function gradient information projected onto a feasible region that consists of candidate test plans given the testing budget constraint. The algorithm is demonstrated on several examples with superior results.

1. Introduction

Integrated reliability testing programs for electronic systems development generally include reliability growth testing, reliability qualification testing and environmental stress screening (ESS). Reliability growth testing is particularly important when system reliability is a critical concern but the system design is new and unproven. In the past, the development budgets for electronic systems were sufficiently large so that allocation of resources for subsystem level testing was a relatively minor concern because each subsystem could be tested for extended time periods. However, in the current business climate involving shorter development schedules and tighter budgets, allocation of reliability growth test times becomes critical. System designers must allocate testing resources to effectively maximize the resultant benefits.

Reliability growth testing provides a systematic method to conduct developmental testing, to track the progress of reliability improvement efforts and to predict system reliability given the observed (or anticipated) rate of improvement. Reliability growth testing has been adopted by many industries, including defense [1], as an integral part of system design activities. Recent experiences using reliability growth testing in the automotive and cellular telephone industries have been described by Freind [2] and Bothwell *et al.* [3], respectively.

For any large system development effort, it is neither practical nor cost-effective to conduct reliability growth testing with the entire system in a fully configured state. At the early stage of system development when reliability growth testing is performed, it is more practical to conduct the testing at the subsystem level. Sometimes, the individual subsystems are not even being produced in the same facility. Additionally, it is very likely that the maturity of the design will be different for each subsystems, and thus, the benefit to be realized from reliability growth testing will naturally differ. Without proper allocation of test times, it is very likely that a limited testing budget will be consumed by testing individual subsystems which offer little potential for additional gains in system reliability.

Rajgopal and Mazumdar [4,5] have demonstrated that it is often less costly to conduct reliability qualification testing at the subsystem level, and they have developed optimal subsystem-level reliability qualification test plans. It is even more important to develop optimal subsystem reliability growth test plans because this testing is already often conducted at the subsystem level, and if done properly, it will yield the largest net return for testing expenditures.

2. Reliability growth testing

Reliability growth testing, also known as Test-Analyze-and-Fix (TAAF) testing, involves the testing of a system early in the development cycle when the design is immature and design changes can be implemented more readily. The design is still evolving at this point and system reliability is improving as compensatory design changes are made in response to observed failures.

This type of testing has many advantages compared to reliability qualification testing. One advantage is that it is not necessary to wait until all design efforts have been completed to initiate a reliability testing program. Another attractive feature of reliability growth testing is that the emphasis is on reliability *improvement* as opposed to reliability *measurement*.

The concept of reliability growth testing was introduced by Duane [6] who was involved in developmental testing of aircraft engines. Crow [7] has further studied reliability growth and proposed that the improvement in reliability can be modeled by a nonhomogeneous Poisson process. He formalized his findings with the popular Crow/AMSAA model with corresponding maximum likelihood estimators for model parameters and goodness-of-fit tests. Other reliability growth models have also been proposed including those of Lloyd [8], Robinson and Dietrich [9,10], and Crow [11–13] who has continued to extend his original findings.

The reliability growth philosophy is to begin reliability testing early in the design and development process, generally exposing the test item to environmental stresses simulating actual usage stresses. When failures are observed, failure analysis is conducted and the appropriate failure mechanism and cause are identified. Then, the design is physically changed to prevent or minimize future occurrence of the same failure mechanism.

In practice, it is generally not possible to implement design changes in response to 100% of observed failures. However, the reliability growth philosophy requires the investigation of all observed failures and the implementation of "fixes" for those failure modes which demonstrate a failure rate that does not support attainment of the system reliability goal. The Duane and Crow/AMSAA models do not require the assumption that the "design fixes" are always successful nor do they assume that the likelihood of an observed failure mechanism falls to zero once a design change has been made. Instead, the collective rate of improvement is used to estimate the reliability given that the design changes have been incorporated.

The Crow/AMSAA reliability growth model is as follows:

$$E(N(\tau)) = \lambda(\tau)^{\beta},$$
$$u(\tau) = \lambda\beta(\tau)^{\beta-1},$$

where,

$N(\tau)$ = number of observed failures in $(0, \tau)$;
$u(\tau)$ = failure intensity (sometimes called the instantaneous failure rate);
λ, β = model parameters ($\lambda > 0,\ \beta > 0$);
τ = development test time.

For $0 < \beta < 1$, failures during development testing occur as a nonhomogeneous Poisson process with a decreasing failure intensity. Then, when development testing is concluded at time τ', subsequent failures in actual (or field) conditions arrive in accordance with a homogeneous Poisson process at a constant rate of $u(\tau')$. In other words, failure intensity is decreasing during development testing and constant thereafter. This is logical because design fixes are being implemented to prevent or minimize the occurrences of observed failures during the testing program. Then, improvement ceases at the conclusion of testing because it is no longer practical or cost effective to incorporate design changes in response to each failure.

In accordance with the Crow/AMSAA model, the interarrival times are distributed according to the exponential distribution after completion of the reliability growth test. Ferdous *et al.* [14] have studied the same problem with Weibull interarrival times.

After reliability growth testing (for time τ'), the MTBF (i.e., the mean interarrival time) and the system reliability, $R(t)$, for time t, are given by the following equations. For most system development programs, the system-level objective is to maximize MTBF or $R(t)$ or to achieve a stated requirement. Since the failures occur as a homogeneous Poisson process after the reliability growth testing, MTBF is constant.

$$MTBF = \frac{1}{u(\tau')} = \frac{1}{\lambda\beta(\tau')^{\beta-1}},$$
$$R(t) = \exp(-u(\tau')t) = \exp(-\lambda\beta(\tau')^{\beta-1}t).$$

In practice, reliability growth testing is generally conducted for large systems at the subsystem level. It is generally not practical to fully assemble the system given the early stage of system development. Also, the potential benefits of reliability growth testing will usually not be the same for each subsystem. This is often known, or can be anticipated, in advance. If certain subsystems are nondevelopmental items (NDI) or they have very minor changes from a previous application, then their designs are likely to be mature and there will be very little benefit to be gained from a formal reliability growth test. To apply the Crow/AMSAA model at the subsystem level, the subsystems must be complex in the sense of having a large number of failure modes.

It is assumed that the individual subsystems are arranged in a series configuration. When reliability growth testing is conducted at the subsystem level, the system-level failure intensity in the working or field condition (after the conclusion of reliability growth testing) is the sum of the individual subsystem failure intensities. This is because the superposition of independent homogeneous Poisson processes forms another homogeneous Poisson process.

Although reliability growth tests are often conducted at the subsystem-level, the principal design objectives are still system-level objectives such as the system reliability (for time t) or the system MTBF. Therefore, allocation of testing resources for the subsystem-level reliability growth tests must recognize the system-level design objectives. The rate of the resulting system failure process depends on the individual subsystem processes. Since the subsystem failure processes are affected by the amount or allocation of reliability growth testing, it becomes imperative that the testing times be allocated intelligently when there is a limited testing budget.

Three common approaches used by practitioners to allocate subsystem-level test plans are:

1. Test for multiples of a predicted (i.e., analytical) subsystem MTBF.
2. Add test time based on incremental improvement in the lowest subsystem MTBF.
3. Equal allocation of test time across all subsystems.

The first approach involves multiplying a predetermined factor by an independently developed predicted MTBF. The logic being that the subsystems with the highest MTBF will require more testing so that failures can be observed and improvements made. The second approach involves allocating test time based on the rank order of the failure intensities. The third approach simply equally allocates available test time or resources to each subsystem. Each of these approaches is inefficient and can produce decidedly nonoptimal results.

Campbell [15] has proposed a more rigorous and efficient approach to the allocation of subsystem test time. He defined a test time increment, say $\Delta\tau$. Then, starting at time zero, test time increments are successively allocated to the individual subsystem offering the highest marginal decrease in failure intensity based on the Crow/AMSAA model. The procedure continues until all available test time has been allocated. This procedure is intuitive and offers significant improvements compared to other approaches. Additionally, as $\Delta\tau$ becomes small, Campbell's method approximates a gradient search. Deficiencies with this approach are that: (1) it does not accommodate constraints on testing cost, as opposed to testing time; and (2) to obtain near-optimal results, $\Delta\tau$ must be very small and then there will an excessively high number of iterations required.

3. Problem formulation

The problem is to select subsystem-level test times which maximize system MTBF or system reliability for a defined mission time t. The amount of testing is limited by a defined budget. For either MTBF or system reliability, this is accomplished by minimizing the failure intensity of the system, i.e., the rate (or parameter) of the system homogeneous Poisson process once testing has been concluded.

The primary test allocation problem is described initially followed by an extension that considers estimation variability. Each subsystem is assumed to be independent. Estimates of Crow/AMSAA model parameters are required for each subsystem.

Problem RG

$$\min \sum_{i=1}^{s} \lambda_i \beta_i (\tau_i + \tau_{0,i})^{\beta_i - 1},$$

$$\text{s.t.} \sum_{i=1}^{s} c_i \tau_i \le C,$$

$$\tau_i \ge 0 \text{ for } i = 1, \ldots, s,$$

where,

$s =$ number of subsystems;

$\lambda_i, \beta_i =$ Crow/AMSAA model parameters for subsystem i ($\lambda_i > 0, 0 < \beta_i \le 1$);

$\tau_i =$ developmental test time for subsystem i;

$\tau_{0,i} =$ developmental test time already accumulated for subsystem i ($\tau_{0,i} \ge 0$);

$c_i =$ cost of testing subsystem i per unit time;

$C =$ available budget for reliability growth testing (constraint).

It is assumed that $\beta_i \le 1$. This means that the system will not become less reliable as a result of the developmental testing and the corresponding design changes. This is a very reasonable assumption during the early stage of development when reliability growth testing occurs. Nevertheless, if this assumption concerning β_i is violated and there exists a subsystem with a β_i greater than one, then the optimal solution will allocate no test time to that particular subsystem and the analysis can continue as if the system had one fewer subsystem.

In many system design and development problems, there will already have been test time accumulated on some of the subsystems. This may be a byproduct of functional testing or other types of testing. If this is the case, it can be accommodated in the problem formulation by the $\tau_{0,i}$ term. The decision variable, τ_i, then is the additional amount of test time. If none of the systems have any test time previously accumulated, then each of the $\tau_{0,i}$ values is set to zero.

For many design problems, there is a limited amount of total test time available, and therefore, a more appropriate constraint is on the sum of the developmental test time. This is the problem described by Campbell [15]. This can be accommodated by setting all c_i values to one. The constraint limit, C, then becomes the amount of available test time.

Engineers planning a reliability growth test almost always treat their estimates of the Crow/AMSAA model parameters and the predicted failure intensity as exact values. Once the test is started, empirical data will be generated and more accurate estimates of model parameters will be obtained. Adjustments to the original planned test times can then be made at interim decision points if the additional data provides sufficient justification. While this strategy effectively minimizes some of the risk associated with uncertainty in the models, planning tools that explicitly recognize estimation variability will offer additional benefits and minimize the reliance on reactive strategies. A more appropriate strategy involves the explicit consideration of the variance of the estimated failure intensity.

Engineers use different methods to estimate Crow/AMSAA or Duane model parameters. Often there is preliminary data as a result of subsystem functional testing to estimate the model parameters. Alternatively, model parameters are estimated based on the completed test results of analogous system development testing programs. Bayesian approaches to reliability growth have also been used successfully, for example the cases reported by Mazzuchi and Soyer [16], Calabria *et al.* [17] and Robinson and Dietrich [18]. As a final resort, practitioners also use available rules-of-thumb to estimate parameters based on analytical reliability predictions and past experience.

Goodness-of-fit tests have been described by Crow [7] and Park and Seoh [19,20] to assess whether the estimated model parameters are satisfactory. Additionally, Crow and Basu [21] have presented estimation methods for cases where there is missing data.

It would be particularly beneficial to consider estimation variability of the subsystem failure intensities, $u_i(\tau)$, if the sources or quantity of data differ appreciably for the subsystems within the overall system. For example, consider a system where one subsystem shows great promise for improvement (as indicated by a high λ_i and low β_i), but there was significant estimation variability for the parameters, while the other subsystems indicate less possibility for improvement but the model parameters were known. A subsystem test allocation strategy which ignores the uncertainty may put unwarranted emphasis on the one subsystem with great potential for improvement. While this test allocation could be warranted, it could also be risky (if the potential for improvement was illusionary) and a more conservative plan could be developed which assured additional improvement for the other subsystems.

The parameter $\hat{u}_i(\tau_i)$ is an estimate of failure intensity based on the estimated model parameters of the Crow/AMSAA model. The variance of the subsystem estimate of failure intensity, $\mathrm{Var}\big(\hat{u}_i(\tau_i)\big)$, can be defined as $\sigma_i^2(\tau_i)$. It is assumed that the variance changes with τ_i. If $\sigma_i^2(\tau_i) > 0$, then the problem formulation can be modified to incorporate the element of risk. Most designers and system users are risk-averse regarding system reliability. It is less important to maximize the expected value of system reliability (given possible uncertainties in failure intensity estimates). Instead, the more important objective is to assure that a very high percentage of all units produced will be as reliable as possible. This objective can be accomplished by formulating the problem to maximize a $(1-\alpha)\times 100\%$ lower-bound on system reliability or MTBF. This is equivalent to minimizing a $\alpha \times 100\%$ upper-bound on system failure intensity. α is a measure of user risk and is user-selected.

The following problem formulation is to maximize the previously described lower-bound on system reliability or MTBF. It involves finding the combination of testing times, τ_i, which collectively maximizes a $\alpha \times 100\%$ upper-bound on system failure intensity.

Problem RG(α).

$$\min \sum_{i=1}^{s} \lambda_i\beta_i\left(\tau_i + \tau_{0,i}\right)^{\beta_i-1} + z_\alpha\sqrt{\sum_{i=1}^{s}\sigma_i^2(\tau_i)},$$

$$\text{s.t.} \sum_{i=1}^{s} c_i\tau_i \le C,$$

$$\tau_i \ge 0 \text{ for } i = 1,\ldots,s,$$

where,

$$z_\alpha = (1-\alpha)\times 100\% \text{ percentile of a standard normal distribution}, Z \sim \mathrm{N}(0,1),$$

$$= \left\{ z \,\middle|\, \int_{z}^{\infty} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)\mathrm{d}x = \alpha \right\}.$$

Problem formulation RG(α) is only appropriate for a defined set of conditions. $\hat{u}_i(\tau_i)$ must be an unbiased estimate of $u_i(\tau_i)$, as provided by the Crow/AMSAA model. Furthermore, at least one of the two following conditions must be satisfied. If these conditions do not apply, then this formulation and the associated solution methodology are not appropriate.

- The estimate of failure intensity (at time τ_i) is normally distributed or can be closely approximated by a normal distribution.

or

- There is a sufficient number of subsystems, s, to invoke the central limit theory for system failure intensity.

Determination of the functional form of $\mathrm{Var}(\hat{u}_i(\tau_i))$, as it varies with τ_i, can be accomplished by analyzing reliability growth data on previously completed reliability growth testing programs or by characterizing the variance of the estimation error of the Crow/AMSAA model parameters, λ_i and β_i.

Realistically, it will often be difficult to estimate $\mathrm{Var}(\hat{u}_i(\tau_i))$ for system development programs without sufficient, representative data. Nevertheless, even in this cases, solution algorithms which consider failure intensity estimation variance will allow test planners to conduct trade-off analyses using assumed variances for subsystems with different relative degrees of variance. This may not provide "optimal solutions" but it will indicate the relative amount of test time to allocate to the various subsystems.

4. Solution algorithms

Algorithms have been developed to determine optimal solutions based on the gradient projection method developed by Rosen [22,23]. These algorithms are superior to other approaches to allocate reliability growth test times. They produce optimal solutions, require fewer iterations and can accommodate a larger range of problems.

If c_i equals one for all subsystems in Problem RG, the solution algorithm yields the same result as the algorithm proposed by Campbell [15] when Campbell's test time increment is set to a small value. In this case, however, the number of iterations is significantly less, as will be later demonstrated in an example.

The optimal solution to Problems RG and RG(α), with a minor condition for the second problem, will always occur with the constraint being tight. To demonstrate this, consider a supposed optimal solution with slack in the constraint. Then additional test time for an arbitrarily selected subsystem can be added. This will decrease the failure intensity for that subsystem, and thus, decrease the system failure intensity and become a superior solution.

The cost function is linear and the optimal solution must lie on the boundary of the feasible region. Therefore, the set of possible solutions can be considered as a $s-1$ dimensional hyperplane. The solution strategy is to project objective function gradient information onto that hyperplane and search the hyperplane using the projected gradients until optimality conditions are achieved.

The objective function for problems RG or RG(α) are nonlinear but differentiable. The gradient vector, $\nabla f(\tau)$, can be determined and the negative gradient used to define the direction of steepest descent where $f(\tau)$ is the objective function. However, moving successively along the negative gradient will lead to infeasible solutions. Rosen developed the methodology to project gradient information onto the feasible region, in this case, the $s-1$ dimensional hyperplane. This method involves the definition of a projection matrix, **P**. The following projection matrix, **P**, is applicable for the reliability growth test allocation problem. **P** is an $s \times s$ matrix.

$$\mathbf{P} = \frac{1}{\sum_{j=1}^{s} c_j^2} \begin{bmatrix} \sum_{j=1}^{s} c_j^2 - c_1^2 & -c_1 c_2 & \cdots & -c_1 c_s \\ -c_1 c_2 & \sum_{j=1}^{s} c_j^2 - c_2^2 & & \vdots \\ \vdots & & \ddots & -c_{s-1} c_s \\ -c_1 c_s & \cdots & -c_{s-1} c_s & \sum_{j=1}^{s} c_j^2 - c_s^2 \end{bmatrix}.$$

The direction of the gradient projected onto the feasible region is then given by,

$$\mathbf{d} = -\mathbf{P}\nabla f(\tau)$$

where,

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_s \end{bmatrix}, \quad \nabla f(\boldsymbol{\tau}) = \begin{bmatrix} \frac{\partial f}{\partial \tau_1} \\ \frac{\partial f}{\partial \tau_2} \\ \vdots \\ \frac{\partial f}{\partial \tau_s} \end{bmatrix},$$

$$\boldsymbol{\tau} \in R^s{}_+, \quad \mathbf{d} \in R^s, \quad \nabla f(\boldsymbol{\tau}) \in R^s.$$

Movement in the direction of **d** will decrease the objective function but always maintain feasibility providing that the search is initiated at some point on the feasible region. The search then involves successively moving, and updating **d**, until optimality conditions are satisfied. When $\mathbf{P}\nabla f(\boldsymbol{\tau})$ equals zero, the Kuhn–Tucker optimality conditions have been met and the optimal solutions found as described by Bazaraa and Shetty [24].

The problem formulation presented here is for a single constraint. However, the methodology could be easily extended to accommodate multiple linear constraints. If nonlinear constraints are considered, then the gradient could be projected onto a tangent hyperplane and a correction made at each iteration to move the current solution in the search to the feasible region.

4.1. *Reliability growth model with known model parameters*

For Problem RG, the projected gradient search can be implemented using the following algorithm. Calculation of **P** and $\nabla f(\tau)$ have been incorporated into the algorithm. q represents the current iteration during the search.

Algorithm RG

Step 0. Initialize $q = 0$; $\tau_i^0 = C/sc_i$ for $i = 1, \ldots, s$; select δ

Step 1. Compute

$$\left.\frac{\partial f}{\partial \tau_i}\right|_q = \lambda_i \beta_i (\beta_i - 1)\left(\tau_i^q + \tau_{0,i}\right)^{\beta_i - 2}, \quad i = 1, \ldots, s$$

$$\Lambda^q = \left(\sum_{j=1}^{s} c_j \frac{\partial f}{\partial \tau_j}\bigg|_q\right) \Bigg/ \sum_{j=1}^{s} c_j^2$$

$$d_i^q = -\frac{\partial f}{\partial \tau_i}\bigg|_q + c_i \Lambda^q, \quad i = 1, \ldots, s$$

Step 2. Update test times $\tau_i^{q+1} = \tau_i^q + \delta d_i^q, \quad i = 1, \ldots, s$

Step 3. Compute

$$f(\boldsymbol{\tau}^{q+1}) = \sum_{i=1}^{s} \lambda_i \beta_i \left(\tau_i^{q+1} + \tau_{0,i}\right)^{\beta_i - 1}$$

Step 4. Check optimality conditions

If $|\mathbf{d}| < \varepsilon$, then the optimal solution, $\boldsymbol{\tau}^*$, is equal to $\boldsymbol{\tau}^{q+1}$ and the minimum system failure intensity is $f(\boldsymbol{\tau}^*)$

If $|\mathbf{d}| > \varepsilon$, then set $q = q + 1$ and return to Step 1

ε is an arbitrarily chosen, very small number (e.g., 10^{-10}). $\delta \times |\mathbf{d}|$ is the distance to be traveled along the hyperplane in the direction of $\mathbf{d}$. There are two strategies for the selection of δ. For each iteration, there will some optimal value of δ which can be found via a line search. This will result in the fewest number of iterations of the algorithm until convergence to the optimal solution. Also, if a unique δ is found for each iteration, then the algorithm can be used without any tunable or problem-specific parameters. Alternatively, a fixed δ can be selected and used throughout the search algorithm. This will likely result in more iterations but the line searches will not be required. With this strategy, the particular choice of δ is problem-specific and it may be necessary to try several different values until the algorithm converges. If δ is too low, the algorithm may take many iterations until convergence, and conversely, if δ is too high, the algorithm may not converge. Any convergence problems can be remedied by lowering δ.

4.2. *Reliability growth model with variance estimates*

Problem RG(α) incorporates the variances of the subsystem failure intensities, which are a function of developmental test time, τ_i. This formulation and the corresponding solution algorithm is intended for risk-averse system designers and users, and system design problems where there are estimates of $\sigma_i^2(\tau_i)$ which are greater than zero. This solution algorithm is the same except that the objective function and partial derivatives incorporate an additional term.

Algorithm RG(α)

Steps 0, 2 and 4, and calculations for Λ^q and d_i^q in *Step* 1 are the same as in Algorithm RG. Changes to Steps 1 and 3 are as follows.

Step 1. Compute

$$\frac{\partial f}{\partial \tau_i}\bigg|_q = \lambda_i \beta_i \left(\beta_i - 1\right)\left(\tau_i^q + \tau_{0,i}\right)^{\beta_i - 2} + \frac{z_\alpha}{2}\left(\left[\frac{\mathrm{d}}{\mathrm{d}\tau_i}\sigma_i^2(\tau_i)\right]_{\tau_i = \tau_i^q} \Bigg/ \sqrt{\sum_{j=1}^{s} \sigma_j^2(\tau_j^q)}\right),$$

$$i = 1, \ldots, s$$

Calculations for Λ^q and d_i^q are the same as for Algorithm RG.

Step 3. Compute

$$f(\boldsymbol{\tau}^{q+1}) = \sum_{i=1}^{s} \lambda_i \beta_i \left(\tau_i^{q+1} + \tau_{0,i}\right)^{\beta_i - 1} + z_\alpha \sqrt{\sum_{i=1}^{s} \sigma_i^2(\tau_i)}$$

Algorithm RG(α) determines the optimal solution providing that the following condition is met. The condition simply states that additional testing never increases the upper-bound on subsystem failure intensity. This will be met for any practical engineering design problem.

$$\frac{\mathrm{d}}{\mathrm{d}\tau_i}\left(\lambda_i \beta_i (\tau_i + \tau_{0,i})^{\beta_i - 1} + z_\alpha \sigma_i(\tau_i)\right) \leq 0 \text{ for } 0 < \tau_i \leq \tau_{\max}$$

and for $i = 1, \ldots, s$,

where

$\tau_{\max}$ = maximum limit on possible test time for a subsystem.

Bishop and Bloomfield [25] have presented some analogous research for the testing of software. Angus [26] and Crow [12] have also described approaches to estimate a bound on predictions derived from reliability growth test results. This becomes increasingly important as the results are projected ahead for planning purposes.

5. Examples

Two example problems will be used to demonstrate the allocation of reliability growth test times. The first example is that solved by Campbell using his approach of incrementally allocating test time based on the marginal decrease in subsystem failure intensity. The system is comprised of five subsystems, 44 000 hours of cumulative

Table 1. Example 1 subsystem parameters

Parameter	*Subsystem number*				
	1	*2*	*3*	*4*	*5*
λ_i	0.500	0.308	0.190	0.0998	0.0614
β_i	0.7	0.7	0.7	0.7	0.7
c_i	1	1	1	1	1
$\tau_{0,i}$	25	50	100	250	500

Table 2. Example 1 results

q	τ_1	τ_2	τ_3	τ_4	τ_5	Λ^q	$f(\boldsymbol{\tau})$	*MTBF*	$\lvert\mathbf{d}\rvert$
0	8 800	8 800	8800	8800	8800	-3.6×10^{-7}	0.053 0377	18.854 51	5.6×10^{-7}
1	13 018	10 002	8142	6725	6114	-3.1×10^{-7}	0.050 8420	19.668 79	2.7×10^{-7}
2	14 614	10 959	8275	5741	4411	-3.1×10^{-7}	0.050 2554	19.898 36	1.6×10^{-7}
3	15 567	11 474	8352	5230	3377	-3.2×10^{-7}	0.050 0680	19.972 83	7.6×10^{-8}
4	16 104	11 683	8293	4936	2983	-3.3×10^{-7}	0.050 0266	19.989 36	3.6×10^{-8}
5	16 396	11 730	8179	4773	2922	-3.3×10^{-7}	0.050 0164	19.993 46	2.1×10^{-8}
6	16 574	11 726	8089	4709	2903	-3.3×10^{-7}	0.050 0128	19.994 89	1.3×10^{-8}
7	16 687	11 708	8029	4684	2892	-3.3×10^{-7}	0.050 0113	19.995 46	8.7×10^{-9}
8	16 762	11 689	7991	4672	2886	-3.3×10^{-7}	0.050 0107	19.995 72	5.8×10^{-9}
9	16 812	11 671	7969	4665	2883	-3.3×10^{-7}	0.050 0104	19.995 83	3.9×10^{-9}
10	16 846	11 657	7955	4662	2881	-3.3×10^{-7}	0.050 0103	19.995 88	2.7×10^{-9}
11	16 869	11 646	7947	4659	2879	-3.3×10^{-7}	0.050 0102	19.995 91	1.9×10^{-9}
12	16 884	11 638	7942	4658	2878	-3.3×10^{-7}	0.050 0102	19.995 92	1.3×10^{-9}
13	16 895	11 632	7939	4656	2878	-3.3×10^{-7}	0.050 0102	19.995 92	9.0×10^{-10}
25*	16 920	11 617	7933	4654	2877	-3.3×10^{-7}	0.050 0102	19.995 93	1.2×10^{-11}

* optimal solution (to closest hour)

test time were allocated and the constraint is for total test time (so c_i is set to one for each subsystem). The subsystem parameters are presented in Table 1, and the algorithm performance is presented in Table 2. An excessive number of significant digits was intentionally left in several columns in the table to demonstrate convergence behavior. In additon δ was set to 10^{10}.

The results in Table 2 show that within three iterations, Algorithm RG identified a solution very close to the optimal solution in terms of the objective function. The optimal solution to the nearest hour of testing was found in the 25th iteration, but changes beyond approximately the tenth iteration are very minor. Campbell's solution approach also produced optimal solutions (actually within 10^{-6}% of optimal) but required 4400 iterations. Thus, the proposed algorithm produces the same result but with significantly reduced effort (25 versus 4400 iterations).

The first example was useful to compare the performance of the algorithm to alternative approaches, however, it does not sufficiently demonstrate the difficulty of this planning problem or the ability of the proposed algorithm to develop superior solutions. In the first example, the β_i and c_i values were the same for each subsystem and it is somewhat obvious which subsystems required more developmental testing.

A second example is presented in Table 3. This is for a system with three subsystems. For this system, the appropriate allocation of reliability growth test times is not obvious and the need for the algorithm is more evident. The constraint is 1000 cost units. Also, for two of the subsystems, the reliability growth parameters are known and the variance is approximately zero. For the third subsystem, there is less certainty regarding the failure intensity function and it has been found that the variance is proportional to the amount of additional testing. In other words, the further ahead the failure intensity is projected, the higher the relative degree of estimation variability.

The example was solved twice. First, the variance information was not considered. The results are presented in Table 4. The optimal solution was found at the eighth iteration (δ was set to 3×10^7). This problem could not have been solved using Campbell's approach because the constraint was on cost, and not total test time. The table also includes a 90% upper-bound on the objective function, $f_{0.10}(\boldsymbol{\tau})$ for informational purposes. This was not a consideration in the initial analysis but it can be used to compare to the results when the variance is considered.

The problem was then solved to minimize $f_{0.10}(\boldsymbol{\tau})$. Subsystem 3 has the greatest estimated rate of improvement (as indicated by β_i) but it also has a higher estimation variance, and therefore, relying on anticipated improvements for this subsystem is risky. It could be argued that the test time should now be increased higher than 125 for subsystem 3 to assure some minimal level of reliability growth in the event that the failure intensity is higher than estimated. However, total cost is constrained and this would mean proportionally less test time for subsystems 1 and 2, where reliability growth can be ac-

Table 3. Example 2 subsystem parameters

Parameter	*Subsystem number*		
	1	*2*	*3*
λ_i	0.02	0.20	0.20
β_i	0.7	0.8	0.6
c_i	1	2	3
$\tau_{0,i}$	0	0	0
$\sigma_i^2(\tau_i)$	≈ 0	≈ 0	$\tau_i/10\,000^2$

Table 4. Example 2 results (without estimation variance)

q	τ_1	τ_2	τ_3	$f(\tau)$	$MTBF$	$f_{0.10}(\tau)$	Λq	$\|\mathbf{d}\|$
0	333	167	111	0.007 820	127.8849	0.009 169	-2.4×10^{-6}	3.1×10^{-6}
1	268	229	91	0.007 630	131.0584	0.008 853	-2.5×10^{-6}	2.5×10^{-6}
2	200	218	121	0.007 497	133.3891	0.008 908	-2.0×10^{-6}	1.9×10^{-6}
3	153	249	116	0.007 408	134.9953	0.008 789	-2.0×10^{-6}	1.4×10^{-6}
4	112	259	124	0.007 355	135.9619	0.008 778	-1.9×10^{-6}	1.0×10^{-6}
5	84	269	126	0.007 330	136.4195	0.008 767	-1.8×10^{-6}	5.5×10^{-7}
6	69	276	127	0.007 324	136.5303	0.008 764	-1.8×10^{-6}	1.6×10^{-7}
7	65	279	126	0.007 324	136.5384	0.008 759	-1.8×10^{-6}	3.1×10^{-8}
8*	65	280	125	0.007 324	136.5387	0.008 756	-1.9×10^{-6}	6.0×10^{-9}

* optimal solution (to closest hour)

curately estimated. A more conservative, risk-averse strategy is actually to decrease the test time on subsystem 3 to obtain more certain benefits on the other two subsystems.

The algorithm converged to the optimal solution at the 21st iteration with a δ of 2×10^7. The resulting test times are 80, 350 and 74, which vary by a relatively large margin from the original test times (which did not consider the variance). As expected, the optimal solution to Problem RG(α) included less developmental testing time for subsystem 3. By allocating 51 fewer time units to subsystem 3, it was possible to increase the amount of testing time by 15 and 70 time units for subsystems 1 and 2 respectively. The minimum $f_{0.10}(\tau)$ was 0.008 583. Therefore, by considering the estimation variance, testing times are recommended that do not lead to the maximum MTBF but provide more assurances that the MTBF will consistently be high given uncontrollable variation.

6. Conclusions

Allocation of reliability growth test time is becoming increasingly important as design schedules become compressed. There may not be enough time to perform as much testing as was done previously. New designs must be transformed into competitive products in a timely manner, however, the products must also be reliable. Therefore, these compressed design schedules cause reliability testing in general, and reliability growth testing in particular, to have greater importance if they can be accomplished efficiently and effectively.

The new algorithm is based on gradient projection. It applies to a greater range of test time allocation problems compared to previously documented solution approaches. Furthermore, when being used with a constraint on total test time (rather than testing cost), the new algorithm produces the same result of competing algorithms, i.e., Campbell [15], but with significantly fewer iterations required.

In the solution algorithm, the gradient of the system failure intensity is projected onto a hyperplane defined by a single, linear constraint on available testing budget. A search over the hyperplane is conducted until optimality conditions are met. In sample problems, it is demonstrated to converge to the optimal solution quickly.

A problem formulation was also presented to minimize an upper-bound estimate for system failure intensity. This approach is suitable for system designers and users who are risk averse. More effort is required to characterize the variance of the estimated system failure intensity.

There is significant ongoing research in reliability growth testing. For example, Benski and Cabau [27] have described experimental designs for reliability growth testing; Ebrahimi [28] has described the practice of developing unique models for each successive design change; and Xie and Zhao [29] have presented a new graphical approach to reliability growth prediction. Whatever new advances are made, the particular problem of allocating test time must be continually addressed to reflect new models and testing philosophies. Furthermore, it becomes more important as design schedules are further compressed and budgets for testing further reduced.

References

[1] Anon. (1981) Military Standard MIL-HDBK-189, Reliability Growth Management. US Government printing office, Washington, DC.

[2] Freind, H. (1995) Reliability growth test planning, in *Proceedings of the 41st Technical Meeting, Institute of Environmental Sciences*, IES, pp. 104–110. Prospect, IL.

[3] Bothwell, R., Donthamsetty, R., Kania, Z. and Wesoloski, R. (1996), Reliability evaluation: field experience from Motorola's cellular base transceiver systems, in *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, pp. 348–359. Piscataway, NJ.

[4] Rajgopal, J. and Mazumdar, M. (1996) A system based component test plan for a series system with type-II censoring. *IEEE Transactions on Reliability*, **45**, 375–378.

[5] Rajgopal, J. and Mazumdar, M. (1995) Designing component test plans for series system reliability via mathematical programming. *Technometrics*, **37**, 195–212.

[6] Duane, J.T. (1964) Learning curve approach to reliability monitoring. *IEEE Transactions on Aerospace*, **2**, 563–566.

[7] Crow, L.H. (1974) Reliability analysis for complex, repairable systems. *SIAM,* Prochan, F. and Serfling, R.J. (eds), *Reliability and Biometry*, Philadelphia, PA. pp. 379–410.

[8] Lloyd, D.K. (1986) Forecasting reliability growth. *Quality and Reliability Engineering Journal*, **2**, 19–23.

[9] Robinson, D. and Dietrich, D. (1987) A new nonparametric growth model. *IEEE Transactions on Reliability*, **36**, 411–418.

[10] Robinson, D. and Dietrich, D. (1988) A system-level reliability growth model, in *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, Piscataway, NJ. pp. 243–247.

[11] Crow, L.H. (1995) Reliability growth projections with applications to integrated testing, in *Proceedings of the 41st Technical Meeting, Institute of Environmental Sciences*, IES, Prospect, IL. pp. 93–103.

[12] Crow, L.H. (1993) Confidence intervals on the reliability of repairable systems, in *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, Piscataway, NJ. pp. 126–133.

[13] Crow, L.H. (1990) Evaluating the reliability of repairable systems, in *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, Piscataway, NJ. pp. 275–279.

[14] Ferdous, J. and Pandey, M. (1995) Reliability estimation with Weibull interarrival times. *Reliability Engineering and System Safety*, **50**, 285–296.

[15] Campbell, C.L. (1990), Subsystem reliability growth allocation, in *Proceedings of the 36th Technical Meeting, Institute of Environmental Sciences*, IES, Prospect, IL. pp. 748–751.

[16] Mazzuchi, T. and Soyer, R. (1993) Bayes method for assessing product reliability during development testing. *IEEE Transactions on Reliability*, **42**, 503–510.

[17] Calabria, R., Guida, M. and Pulcini, G (1992) Bayes procedure for estimation of current system reliability. *IEEE Transactions on Reliability*, **41**, 616–620.

[18] Robinson, D. and Dietrich, D. (1989) Nonparametric Bayes reliability growth model. *IEEE Transactions on Reliability*, **38**, 591–598.

[19] Park, W. and Seoh, M. (1992) Goodness-of-fit tests for the power-law process. *IEEE Transactions on Reliability*, **41**, 107–111.

[20] Park, W. and Seoh, M. (1994) More goodness-of-fit tests for the power-law process. *IEEE Transactions on Reliability*, **43**, 275–278.

[21] Crow, L.H. and Basu, A.P. (1988) Reliability growth estimation with missing data, in *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, Piscataway, NJ. pp. 248–253.

[22] Rosen, J.B. (1960) The gradient projection method for nonlinear programming, part I. *SIAM Journal of Applied Mathematics*, **8**, 181–217.

[23] Rosen, J.B. (1961) The gradient projection method for nonlinear programming, part II. *SIAM Journal of Applied Mathematics*, **9**, 514–553.

[24] Bazaraa, M.S. and Shetty, C.M. (1979) *Nonlinear Programming*, John Wiley &Sons, New York.

[25] Bishop, P.G. and Bloomfield, R.E. (1996) Conservative theory for long-term reliability growth prediction, in *Proceedings of the International Symposium on Software Reliability*, IEEE, Los Alamitos, CA. pp. 308–317.

[26] Angus, J.E. (1992) Lower confidence bound on the percentage improvement in comparing two failure rates. *IEEE Transactions on Reliability*, **41**, 239–240.

[27] Benski, C. and Cabau, E. (1995) Unreplicated experimental designs in reliability growth programs. *IEEE Transactions on Reliability*, **44**, 199–205.

[28] Ebrahimi, N. (1996) How to model reliability growth when times of design modifications are known. *IEEE Transactions on Reliability*, **45**, 54–58.

[29] Xie, M. and Zhao, M. (1996) Reliability growth plot – an underutilized tool in reliability analysis. *Microelectronics and Reliability*, **36**, 797–805.

Biography

David W. Coit is an Assistant Professor at Rutgers University. He received a BS degree in Mechanical Engineering from Cornell University (1980), an MBA degree from Rensselaer Polytechnic Institute (1988) and MS and PhD degrees (1993, 1996) in Industrial Engineering from the University of Pittsburgh. From 1980 to 1992, he was employed at the IIT Research Institute (IITRI), Rome NY, where he worked as a Reliability Engineer and Program Manager. From 1988 to 1992, he was the Engineering Manager at IITRI's Assurance Technology Center. His current research involves reliability prediction and optimization, accelerated life testing models and industrial applications for artificial neural networks. He is a member of IEEE, IIE and INFORMS.