NPS-AM-09-023

EXCERPT FROM THE PROCEEDINGS

OF THE

SIXTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

MATHEMATICAL MODELING FOR OPTIMAL SYSTEM TESTING UNDER FIXED-COST CONSTRAINT

Published: 22 April 2009

by

Karl D. Pfeiffer, Valery A. Kanevsky and Thomas J. Housel

6th Annual Acquisition Research Symposium
of the Naval Postgraduate School:

Volume I:
Defense Acquisition in Transition

May 13-14, 2009

Prepared for: Naval Postgraduate School, Monterey, California 93943

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Mathematical Modeling for Optimal System Testing under Fixed-cost Constraint

Presenter: Karl D. Pfeiffer is an Assistant Professor of Information Sciences at the Naval Postgraduate School and an active duty Air Force officer. His current research interests include decision-making under uncertainty, particularly with regard to command and control (C2) systems; stochastic modeling of environmental impacts to weapons and communication systems; and probability modeling and numerical simulation in support of search, identification and pattern recognition applications (e.g., complex system testing, allocation of effort for reconnaissance, etc.).

Author: Valery A. Kanevsky is a Research Professor of Information Sciences at the Naval Postgraduate School. His research interests include probabilistic pattern recognition; inference from randomly distributed inaccurate measurements, with application to mobile communication; patterns and image recognition in biometrics; computational biology algorithms for microarray data analysis; Kolmogorov complexity, with application to value allocation for processes without saleable output; and Monte Carlo methods for branching processes and simulation of random variables with arbitrary distribution functions. Kanevsky's most current work is focused on statistical inference about the state of a system based on distributed binary testing. Another area of interest is in the so-called needle-in-a-haystack problem: searching for multiple dependencies in activities within public communication networks as predictors of external events of significance (e.g., terrorist activities, stock market anomalies).

Author: Thomas J. Housel is a Professor of Information Sciences at the Naval Postgraduate School. Housel specializes in valuing intellectual capital, knowledge management, telecommunications, information technology, value-based business process re-engineering, and knowledge value measurement in profit and non-profit organizations. His current research focuses on the use of knowledge-value added (KVA) and real options models in identifying, valuing, maintaining, and exercising options in military decision-making. His work on measuring the value of intellectual capital has been featured in a *Fortune* cover story (October 3, 1994) and *Investor's Business Daily*, numerous books, professional periodicals, and academic journals (most recently in the *Journal of Intellectual Capital*, 2005). His latest books include: *Measuring and Managing Knowledge* and *Global Telecommunications Revolution: The Business Perspective* with McGraw-Hill (both in 2001).

Abstract

Testing of complex systems is a fundamentally difficult task, whether locating faults (diagnostic testing) or implementing upgrades (regression testing). Branch paths through the system increase as a function of the number of components and interconnections, leading to exponential growth in the number of test cases for exhaustive examination. In practice, the typical cost for testing in schedule or in budget means that only a small fraction of these paths are investigated. Given some fixed cost, then, which tests should we execute to guarantee the greatest information returned for the effort? In this work, we develop an approach to system testing using an abstract model flexible enough to be applied to both diagnostic and regression testing, grounded in a mathematical model suitable for rigorous analysis and Monte Carlo simulation. Early results indicate that in many cases of interest, a good, though not optimal, solution to the fixed-constraint problem (how many tests for budget *x*?) can be approached as a simple best-next strategy (which test returns the highest information per unit cost?). The goal of this modeling work is to construct a decision-support tool for the Navy Program Executive Office Integrated Warfare Systems (PEO IWS) offering quantitative information about cost versus diagnostic certainty in system testing.

Keywords: diagnostic testing, regression testing, automated testing, Monte Carlo simulation, sequential Bayesian inference, knapsack problem

1. Introduction

Many of us commute to work every day in what has become a relatively complex system: an automobile. When this system fails us, we are forced to allocate resources (i.e., time and money) to diagnostic testing and repair of one or more components within the system. The budget for this testing and repair is generally constrained by our prudence and our pocketbooks; we hope that the service technician employs a testing strategy that develops the best answer (e.g., replace the alternator) for the least cost.

There are several possible stopping criteria for this testing. In particular, a logical choice would be to stop testing when the cost of replacement of all suspect parts is less than the cost of conducting one more test. This presupposes, however, that our system under test is in a failed state. Suppose we replace a known defective part or perform upgrade maintenance on a component: how much testing must we accomplish to convince ourselves that the system will operate correctly under all conditions?

In this paper we present a language of description and a mathematical model to describe a system under testing, with the goal of evaluating strategies in terms of the information returned by a set of tests. In this framework, then, testing is the mechanism by which we trade some fixed cost (e.g., time, money) for information about the state of subcomponents in our system. In general, we seek the maximum information available for the minimum cost. In the present study, we consider the following question: Given a fixed budget, what is the maximum information discoverable from a particular test suite?

Mathematical models of component and system reliability have roots in the work of von Neumann (1952) and Moore and Shannon (1956a; 1956b), as well as the seminal text by Barlow and Proschan (1965). The focus of these early works is generally on assessing the overall system reliability, particularly with regard to the economics of preventative vice reactive maintenance (e.g., see, Bovaird, 1961). In the present work, the focus is on efficiently identifying either a defective-by-design or failed component in a complex system.

This fault diagnosis is sometimes referred to as the test-sequencing problem, and has also been well studied (e.g., see, Sobel & Groll, 1966; Garey, 1972; Fishman, 1990; Barford, Kanevsky & Kamas, 2004). In general, these investigators start with a system in a known failed state with the goal of finding the most cost-effective sequence of diagnostics to locate the failed component (or components) under a given set of assumptions.

In contrast to fault diagnosis, the general case of regression testing appears to have received less attention in the open literature, with more specific cases examined in the realm of software engineering (e.g., Leung & White, 1991; White & Leung, 1992; Weyuker, 1998; Tsai, 2001; Rothermel, Untch & Harrold, 2001; Mao & Lu, 2005). These studies typically start with a fully functioning system undergoing component modifications or upgrades, with the task of establishing that component modifications have not introduced new defects into the system.

In the present study, we treat testing as a unified activity, with risk and cost as the common tension regulating the degree of testing required. From a fault-diagnosis perspective, we want to arrive at a replacement or maintenance decision quickly while ensuring the system is restored to perfect functionality. From a regression testing perspective, particularly with the open architectures employed within the Integrated Warfare System, following an engineering change or upgrade to a component, we want to conduct enough testing to verify that the system remains in perfect function. The element of risk is that costs incurred for perfect knowledge may approach infinity, *or may not be achievable with a given test suite*. From a practical perspective, then, we accept with some level of confidence (e.g., 99% certainty, 95% certainty) that our diagnosis or prognosis is correct.

The rest of this paper is organized as follows. Section 2 presents the model formulation and fundamental definitions. Section 3 details the mathematical model derived from this framework. Section 4 outlines numerical experiments examining testing strategies in terms of this model and presents simulation results. Section 5 discusses conclusions and avenues for future work.

2. Definitions and Model Formulation

The growing use of commercial off-the-shelf technologies in current weapons systems (Caruso, 1995; Dalcher, 2000), coupled with the complexity of end-to-end systems (Athans, 1987; Brazet, 1993), suggests that we may never have enough information to fully specify our system as a white box, with all software, hardware and communication interfaces perfectly characterized. Thus, we construct our model with broad parameters that can be constrained as narrowly as available information permits.

We characterize the model system **S** as a collection of modules and a suite of tests used to interrogate these modules. We examine the system through this test suite to identify defective modules or to determine that no defective modules exist. We assume that tests return ambiguous information about the state of modules within the system; that is, no single test is likely to return perfect knowledge about a particular module.

Thus, in general, we expect that some sequence of tests must be applied to arrive at a correct diagnosis, where the term *correct* may require careful definition in terms of acceptable risk or required level of confidence. Stochastic simulation of the model system provides a framework in which different testing strategies may be applied and measured for further insight. Using this Monte Carlo approach, we may also test the bounds of our initial assumptions with additional simulation.

2.1 System and module objects

Within the system **S**, each module M_i represents the smallest diagnostic or replaceable unit, which does not necessarily correspond to a single physical component in the modeled system. We consider, for example, a motherboard comprised of a central processing unit (CPU), physical random access memory (RAM), a graphics adapter, and keyboard interface, each of which may cause the motherboard to fail. This might be modeled as a single module labeled *Motherboard* if the standard corrective maintenance action is to replace the motherboard. With more granular diagnostics and maintenance practices, however, we might model these components as *MB_CPU, MB_RAM, MB_Graphics, and MB_Keyboard* because each was testable and replaceable.

Fundamental to this aspect of the model is a source of failure rate data for the system components. These failure rates become the a priori data in the larger probability model and do not necessarily need to be precise to add value to the iterative simulation results. The relative rates among the modeled components (e.g., the *Server* module fails about five times as often as the *Router* module) should be close to the observed data in the physical system to provide the most realistic convergence in testing to a correct diagnosis.

2.2 Test objects

Tests are modeled as system of objects which, when executed, provide an ambiguous assessment of one or more modules within **S**. This ambiguity stems from two essential elements that map the tractable model to physical reality.

The first ambiguous aspect is that any given test likely exercises only a portion of the functionality within a module. Each M_i is modeled as a unit circle A_i (Figure 1). Defects, when present, are assumed uniformly distributed on this circle. We assume that while multiple modules may be defective, only one defect exists per module. A defect in M_i is modeled as a random point on A_i or equivalently a random point on the interval [0, 1]. Although the module is the unit of replacement, we parameterize the sub-module details by treating them as a continuous space covered, in part, by a given test.

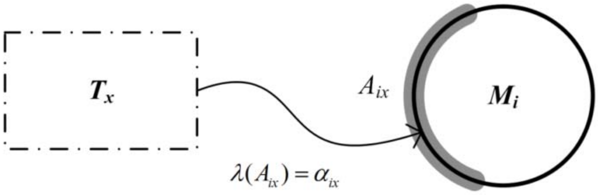

Figure 1. The Simple Coverage of Test T_x on Module M_i, Indicated by the Gray Arc A_{ix}. (The scalar measure of this coverage $\lambda(A_{ix}) = \alpha_{ix}$ represents the fraction of M_i exercised by T_x.)

We model the coverage of test T_x on module M_i as the arc A_{ix} (Figure 1). When T_x is executed, or applied to the model system, the arc A_{ix} on M_i is inspected for a defect. Given the assumption that defects appear uniformly on this unit circle, the probability that a defect in M_i will be detected by T_x is the measure of this arc $\lambda(A_{ix}) = \alpha_{ix}$. The scalar probability of detection by a test is precisely this user-specified functionality exercised by the test. This element of our language of description permits some ambiguity in characterizing the physical system without loss of rigor in modeling these tests and modules. In practice, given a sufficient number of real-world cases from the physical system, this estimate for A_{ix} could be refined through analysis of simulation results.

The second ambiguous aspect is that any given test likely covers multiple modules, such that any test result must be interpreted as applying to *all* modules covered by that test (Figure 2). For example, a positive result (FAIL) from a diagnostic test that covers the modules *Carburetor, Distributor Cap,* and *Spark Plug Wiring* indicates that at least one of these modules contains a defect (has failed), though additional testing would be required to identify which module is the culprit. Because we expect that a given test exercises multiple modules in the system, we speak more generally of the coverage of T_x on **S** (Figure 2).

Within the model, an executed test assumes one of two values: PASS or FAIL. A PASS result for a given test T_x indicates that no region covered by this test contains a defect. A FAIL result indicates that at least one of the modules covered by T_x contains a defect or is BAD in the model definition. While a FAIL result should reduce the set of modules that may need to be replaced, a perfect result—replacing only failed modules—will typically require some sequence of tests. Indeed, for a particular configuration of tests and modules, this perfect result may not be achievable. Analysis of simulation results should help identify those cases in which further testing will yield no new information.

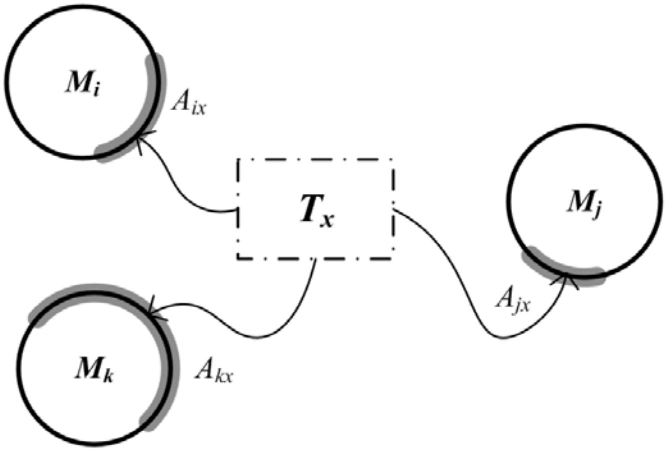

Figure 2. Notional Depiction of the Coverage of T_x on S, with Multiple Modules Exercised by This Test (A FAIL result from T_x indicates that at least one of the subset {M_i, M_j, M_k} has failed.)

The use of vector arcs to model the coverage relationship between tests and modules enables precision when specifying the coverage by multiple tests on a single replaceable unit (Figure 3). Although several tests in the system suite may exercise a given module, it is likely in the physical system that these tests overlap significantly. This language of description, then, permits a user specification of the physical system in broad terms (e.g., the *Remote Control* test and *Obstacle Detection* test both exercise about 70% of the *Garage Door Motor* module, with about 20% overlap between the two tests). Even if these data are estimated from the physical system, existing case data and simulation results could be used to provide better specification of these joint coverages.

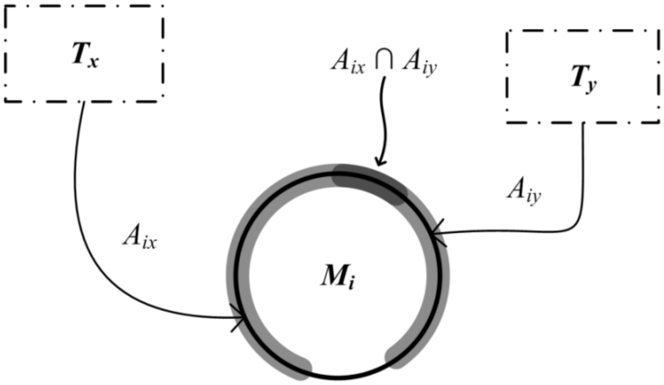

Figure 3. Overlapping Coverage between Tests T_x and T_y Are Characterized with the Arcs A_{ix} and A_{iy} (The joint coverage is computable as the intersection of these arcs.)

2.3 Summary

This conceptual model captures the essential elements of a system with respect to testing, suitable for both diagnostic and regression work. The physical system is specified in terms of modules, tests, and coverages, with model elements constructed in such a way that imperfect information can still be used as an initial state.

Although the model requires that the physical system be decomposable into discrete units of replacement, this does not limit the usefulness of this approach. Within the system, we expect different levels of maintenance (e.g., depot maintenance, field-level maintenance) and different levels of diagnostic techniques, all of which could be treated as different layers within this framework. We next formalize these model elements in mathematical language to construct a suitable computer simulation to investigate these testing strategies.

3. Mathematical Analysis

Our goal in this study of system testing is to maximize certainty for a given cost. In developing a probability framework to model this process, we first form simple objective measures to characterize knowledge of the system state. We next examine a simple, step-wise strategy to predict a test sequence that will maximize or minimize these measures. We then compare this strategy to a two-test approach.

The motivation for examining a two-test (or *k*-test) strategy under fixed cost is that this problem very much resembles the classic knapsack problem (Corman, Leiserson & Rivest, 2002). Choosing at each step the single test that offers the largest increase in information (that is, inserting the largest item into the knapsack first) does not guarantee that we will, for a fixed cost, achieve the greatest information gain (maximize the content of our knapsack). It would be computationally advantageous, though, if we could demonstrate that for many cases of interest, a simple best-next strategy can approach a *k*-step strategy in information return (Cover & Thomas, 1991).

3.1 Module definitions

Within our system **S**, we define B_i and G_i as the events that module M_i is bad or is good, with corresponding probabilities:

$$\begin{aligned} \mathrm{P}(B_i) &= b_i \\ \mathrm{P}(G_i) &= 1 - b_i \end{aligned} \tag{3.1}$$

Each b_i represents information we have about the state of module M_i, and the collection $\{b_i\}$ gives us some insight into the health of **S**. Prior to any testing, we expect each b_i is initialized based on an a priori failure rate for M_i.

The probability b_i is an intuitive measure of information, and we see that as b_i tends to 0 (good) or 1 (bad), our knowledge about M_i becomes more certain. A classic, quantitative measure of this knowledge is the information entropy (Shannon, 1948):

$$h_i = -b_i \log_2 b_i - (1 - b_i)\log_2(1 - b_i) \tag{3.2}$$

We see that as b_i tends to 0 or 1, h_i is minimized (Figure 4). By applying tests from our diagnostic suite, we should become more certain about the state of a module (good or bad) and so act to nudge b_i to the edges of the interval [0, 1]. We can measure this improvement in certainty as a reduction in the individual module entropy h_i, aggregated over the system:

$$H = \sum_{i=1}^{n} -b_i \log_2 b_i - (1 - b_i)\log_2(1 - b_i) \tag{3.3}$$

Entropy is computationally attractive as a continuous and differentiable function over the interval of interest (Figure 4), though h_i may be less intuitive when deciding which modules to replace. A measure similar to entropy, though not differentiable at maximum entropy, is:

$$q_i = \max(b_i, 1 - b_i) \tag{3.4}$$

We can think of q_i as a quality gauge of this replacement (or maintenance) decision with respect to a particular module. If, for example, a particular module has a b_i = 0.70, we may replace it knowing that this informed guess should be correct 70% of the time. This also means that in 30% of these cases, we will unnecessarily replace or perform more granular debugging on this module. Our number of correct diagnoses across the system will increase as each b_i is adjusted, by testing, away from b_i=0.5 towards either 0 or 1 (Figure 4). Although this is not a rigorous result, it can be shown that minimizing system entropy is approximately equivalent to maximizing the number of correct diagnoses.

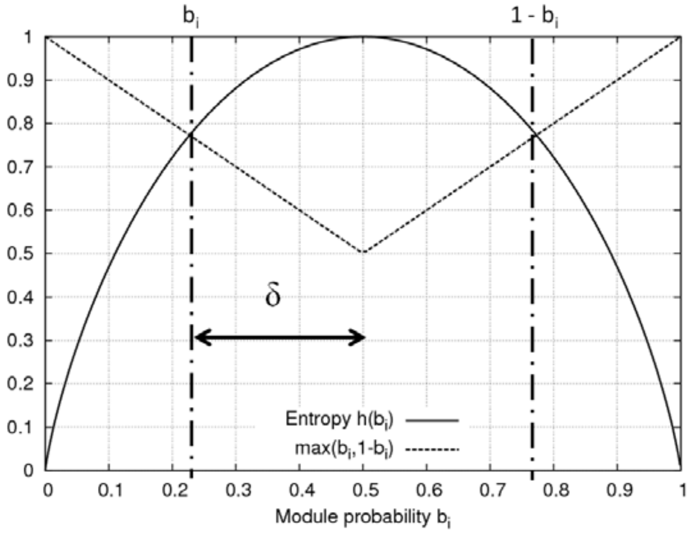

Figure 4. Module Entropy *$h(b_i)$* and *q_i* = max (*b_i*, 1-*b_i*), with Notional Module Probability *b_i* Indicated (The scalar *δ* represents the displacement of *b_i* from maximum entropy (*b_i* = 0.5). Note that by symmetry, *$h(b_i) = h(1 – b_i)$*, with distance *2δ* between these states.)

Consistent with previous studies (e.g., Birnbaum, Esary & Saunders, 1961; Butterworth, 1972; Ben-Dov, 1981), we characterize our knowledge of **S** as a vector of these module probabilities, $\{b_i\}$, where each component probability b_i is on the interval [0, 1]. The true state of **S** is a bit vector with each component exactly 0 (good) or 1 (bad); in practice, we are unlikely to achieve this "perfect" knowledge, but computation of h_i or q_i permits some insight into this true state.

Earlier studies typically examined scenarios in which **S** will function only if all modules are working correctly (serial system), some modules are working correctly (k-of-n system), or if at least one module is working correctly (parallel system). In the present study, we make no assumptions about whether **S** is in a known down state and instead focus on characterizing the health of the system—similar to literature in optimal maintenance strategies (e.g., Boivard, 1961; or Barlow & Proschan, 1965). The focus in the present work is on the nature of the test suite available to the diagnostician or maintainer, and the most effective use of that suite to better characterize **S.** We next present the mathematical model for tests and testing.

3.2 Test definitions

Similar to our model of modules, we define P_x and F_x as the events that test T_x passes or fails, respectively. We expect either result (pass or fail) to return ambiguous information because a test likely exercises or covers only some fraction of the functionality of a module (Figure 1), and because the test likely exercises several modules simultaneously (Figure 2). Thus, a passing result for T_x indicates only that no defect was detected, while a failing result narrows the pool of suspect modules to those exercised by T_x.

After execution of a test, we update the prior probability b_i to the new probability b_i' based on the test outcome:

$$b_i' = \begin{cases} \mathrm{P}(B_i \mid P_x) & \text{if } T_x \text{ passes} \\ \mathrm{P}(B_i \mid F_x) & \text{if } T_x \text{ fails} \end{cases} \tag{3.5}$$

Using Bayes' rule, we can compute these probabilities as:

$$\mathrm{P}(B_i \mid P_x) = \frac{\mathrm{P}(P_x \mid B_i)\mathrm{P}(B_i)}{\mathrm{P}(P_x)} = \left(\frac{\mathrm{P}(P_x \mid B_i)}{\mathrm{P}(P_x)}\right) b_i \tag{3.6}$$

$$\mathrm{P}(B_i \mid F_x) = \frac{\mathrm{P}(F_x \mid B_i)\mathrm{P}(B_i)}{\mathrm{P}(F_x)} = \left(\frac{\mathrm{P}(F_x \mid B_i)}{\mathrm{P}(F_x)}\right) b_i \tag{3.7}$$

These results suggest that tests can be seen as operators that transform b_i into b_i'. We note that the module probability is unchanged if test T_x has no coverage on M_i. In this case, the conditional probabilities $\mathrm{P}(P_x|B_i)$ and $\mathrm{P}(F_x|B_i)$ degenerate to the unconditional probabilities $\mathrm{P}(P_x)$ and $\mathrm{P}(F_x)$, and so we have $b_i' = b_i$.

Because the execution of test T_x necessarily incurs some cost in budget or schedule, we are motivated to compute a forecast value, q_{ix}, from a weighted sum of the unconditional probabilities on P_x and F_x (Equation 3.8). Given the prior state of the system as the set of module probabilities $\{b_i\}$, this method can then be used to assess the expected change *across the system* for a particular test T_x.

$$\begin{aligned} q_{ix} &= \max\begin{pmatrix} \mathrm{P}(B_i \mid P_x) \\ \mathrm{P}(G_i \mid P_x) \end{pmatrix} \mathrm{P}(P_x) + \max\begin{pmatrix} \mathrm{P}(B_i \mid F_x) \\ \mathrm{P}(\mathrm{G}_i \mid F_x) \end{pmatrix} \mathrm{P}(F_x) \\ &= \max\begin{pmatrix} \mathrm{P}(P_x \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(P_x \mid G_i)\mathrm{P}(G_i) \end{pmatrix} + \max\begin{pmatrix} \mathrm{P}(F_x \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(F_x \mid G_i)\mathrm{P}(G_i) \end{pmatrix} \end{aligned} \tag{3.8}$$

Using the expected value of b_i after test T_x (Equation 3.8), we can form a composite measure over our system of n modules with the sum $\mathrm{Q}(T_x)$:

$$Q(T_x) = \sum_{i=1}^{n} q_{ix} \tag{3.9}$$

We expect that if $\mathrm{Q}(T_x) > \mathrm{Q}(T_y)$, then test T_x will return more information than test T_y (Figure 4). This value, however, is a forecast; the actual information returned for a particular test execution may vary widely by scenario.

We note also that because Q depends upon our current knowledge of the system as the set of probabilities $\{b_i\}$, and because this set is constantly updated by testing, the choice of a particular test T_x may yield widely varying $\mathrm{Q}(T_x)$ depending upon **when** T_x is executed in the test sequence.

To calculate q_{ix} we need the conditional probability $\mathrm{P}(P_x|B_i)$, which we compute by first considering the unconditional probability $\mathrm{P}(P_x)$. We note that a test T_x will pass if every module covered is either good (Equation 3.1) or bad but undetected. A test T_x will fail to find a defect with probability $(1 - \alpha_{ix})$, or the complement to the fractional coverage of T_x on M_i (Figure 1). Considering all n modules in the system, then, we have:

$$\begin{aligned} \mathrm{P}(P_x) &= \prod_{i=1}^{n} \left(\underbrace{(1-b_i)}_{GOOD} + \underbrace{(1-\alpha_{ix})b_i}_{BAD, NOT\ DETECTED} \right) \\ &= \prod_{i=1}^{n} \left[1 - \alpha_{ix} b_i \right] \end{aligned} \tag{3.10}$$

Given that module M_i is bad ($b_i = 1$), we then have:

$$\mathrm{P}(P_x \mid B_i) = (1-\alpha_{ix}) \prod_{j \neq i}^{n} \left[1 - \alpha_{jx} b_j \right] \tag{3.11}$$

Similarly, we note that $\mathrm{P}(F_x) = 1 - \mathrm{P}(P_x)$, and we can then see that:

$$\mathrm{P}(F_x \mid B_i) = 1 - (1-\alpha_{ix}) \prod_{j \neq i}^{n} \left[1 - \alpha_{jx} b_j \right] \tag{3.12}$$

We see that if T_x has no coverage on M_i ($\alpha_{ix} = 0$), Equations 3.11 and 3.12 reduce to the unconditional probabilities on T_x. Similarly, if T_x has perfect coverage ($\alpha_{ix} = 1$), Equations 3.11 and 3.12 reduce to 0 (T_x cannot pass if M_i is bad) and 1 (T_x must fail if M_i is bad).

Using Equation 3.10 and its complement, the conditional probabilities given that M_i is good are:

$$\mathrm{P}(P_x \mid G_i) = \prod_{j \neq i}^{n} \left[1 - \alpha_{jx} b_j \right] \tag{3.13}$$

$$\mathrm{P}(F_x \mid G_i) = 1 - \prod_{j \neq i}^{n} \left[1 - \alpha_{jx} b_j \right] \tag{3.14}$$

A quick check of the boundaries shows that if M_i is good and there are no other coverages on M_i (all $\alpha_{jx} = 0$), Equation 3.13 reduces to 1 (T_x must pass), and Equation 3.14 reduces to 0 (T_x cannot fail). Indeed, this set of equations (3.11-3.14) addresses computationally the ambiguity associated with test results, coverages and modules (Figure 1 and Figure 2).

3.3 Test strategies

The objective function $Q(T_x)$ (Equation 3.9) is necessarily a one-step method if we choose that T_x which maximizes Q. If we have only the budget or schedule to execute one more test, maximizing Equation 3.9 will yield the optimal result. In practice, though, we expect that we may have the resources to execute some number of tests; and, similar to the classic knapsack problem (Corman, Leiserson & Rivest, 2002), we are not guaranteed that this simple, one-step strategy will generally yield the largest information gain for a given cost.

As a simple example of this knapsack problem, consider three tests {T_1, T_2, T_3} with a forecast information return of {$Q(T_1) = 3$, $Q(T_2) = 4$, $Q(T_3) = 6$} for associated cost {3, 4, 5}; the units of information and cost are not important to our point and can be thought of as unit cost per bit. Given a fixed cost constraint of 7, the single best-next test choice is T_3, with a cost-per-bit 5/6 and a net return of 5. The choice of T_3, though, means that we cannot execute another test within the cost constraint of 7. This strategy, then, is clearly not optimal for the fixed constraint because the choice of T_1 and T_2, each individually more expensive than T_3, yields a net information return of 7.

To guarantee an optimal solution, then, we are obligated to compute 2^k possible outcomes for a suite of k tests for all n modules in **S**. To mitigate this computational burden, we examine the real differences between an optimal solution and a good solution for our scenarios of interest.

For additional insight, we consider a two-step strategy and its associated objective function with four components (Equation 3.15), computed over all modules (Equation 3.16):

$$\begin{aligned} q_{ixy} \quad &= \quad \max\begin{pmatrix} \mathrm{P}(P_x P_y \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(P_x P_y \mid G_i)\mathrm{P}(G_i) \end{pmatrix} + \max\begin{pmatrix} \mathrm{P}(F_x P_y \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(F_x P_y \mid G_i)\mathrm{P}(G_i) \end{pmatrix} \\ &+ \quad \max\begin{pmatrix} \mathrm{P}(P_x F_y \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(P_x F_y \mid G_i)\mathrm{P}(G_i) \end{pmatrix} + \max\begin{pmatrix} \mathrm{P}(F_x F_y \mid B_i)\mathrm{P}(B_i) \\ \mathrm{P}(F_x F_y \mid G_i)\mathrm{P}(G_i) \end{pmatrix} \end{aligned} \tag{3.15}$$

$$Q(T_x T_y) = \sum_{i=1}^{n} q_{ixy} \tag{3.16}$$

We note that in these pair-wise calculations (or in k-wise calculations), we must consider the possible intersection of coverages between T_x and T_y (Figure 3). That is, we expect two tests with significant overlap in module coverage should not yield a higher q_{ixy} than two paired tests with similar fractional coverages but no overlap or intersection between the two tests. Although the analytic work for these conditional probabilities follows Equation 3.11—3.14, we next turn to simulation to exercise this strategy for comparison to the single-step, best-next strategy.

3.4 Summary

We have presented the mathematical details supporting the abstract model presented in Section 2. We characterize our knowledge of the system health as a collection of probabilities {b_i}, where b_i indicates the probability that component M_i is bad (b_i=1) or good (b_i=0). Using a sequential Bayes' approach, prior probabilities in {b_i} are updated following test execution. As more tests are applied, each T_x should act to minimize entropy H (Equation 3.3) or increase our certainty about the state of each module as either good or bad.

The paper focuses on strategies to choose a test, or sequence of tests, to guarantee the best (or at least a very good) return on the budget or schedule allocated to testing. In the present work, our constraint is a given, fixed cost and our goal is to find the greatest information gain possible in the test suite within this cost. Although an optimal solution may often be computationally untenable, we next examine simulation results to better estimate the distance between “optimal” and “good.”

4. Modeling Approach and Simulation Results

In support of this research, a desktop simulation was developed to implement the analysis presented in Section 3 and to further examine the choices among test strategies. In addition to the best-next and two-step strategies, a random-strategy case was coded to select a test sequence randomly, and a pathological worst-case strategy was created to minimize rather than maximize the information return per test. The random and worst-case configurations were developed to provide some contrast to the “best” strategies.

4.1 Model description

The simulation code implements object models of Tests and Modules, collected under a System object. Configuration parameters are set in an XML text file (Figure 5). Each XML file represents a *simulation,* which is comprised of one or more *configurations* or simulation cases. Each configuration or case is then executed for some number of trials.

Within each configuration, the number of modules and tests are explicitly set, while the module a priori failure rates and test-module coverages are established randomly within minimum and maximum parameters (Figure 5). These random coverages are reconfigurable between trials to permit a Monte Carlo investigation of the initial data. While these randomized scenarios provide some insight into systems testing, sufficient flexibility exists in the computer code and the XML configuration parameters to encompass more realistic systems.

Because of the iterative nature of the model, the algorithm should be relatively insensitive to the initial conditions in the model with respect to module a priori failure rates. That is, the state vector {b_i} is constantly adjusted through the application of tests, and we expect this convergence to dominate the final or quasi-steady state of knowledge regarding our system.

```
<?xml version="1.0"?>
<simulation>
<!-- ======================== -->
<!-- SAMPLE CONFIGURATION FILE -->
<!-- ======================== -->
<configuration>
<!-- =================== -->
<!-- EXECUTION PARAMETERS -->
<!-- =================== -->
<CaseName>best</CaseName>
<Strategy>best</Strategy>
<RandomSeed>-1</RandomSeed>
<NumberOfTrials>10</NumberOfTrials>
<DecisionThreshold>0.90</DecisionThreshold>
<DefectsPerTrial>1</DefectsPerTrial>
<LogFileName>simulation.log</LogFileName>
<ReconfigureTestsPerTrial>yes</ReconfigureTests
PerTrial>
<!-- ================ -->
<!-- MODULE PARAMETERS -->
<!-- ================ -->
<NumberOfModules>10</NumberOfModules>
<FailureRate>
  <Minimum>0.5</Minimum>
  <Maximum>0.5</Maximum>
</FailureRate>
<CostPerModule>
  <Minimum>1.0</Minimum>
  <Maximum>1.0</Maximum>
</CostPerModule>
<SumCostOfAllModules>100.0</SumCostOfAllModules
>
<TestsPerModule>
  <Minimum>1</Minimum>
  <Maximum>5</Maximum>
</TestsPerModule>
<!-- ============== -->
<!-- TEST PARAMETERS -->
<!-- ============== -->
<NumberOfTests>35</NumberOfTests>
<CostPerTest>
  <Minimum>1.00</Minimum>
  <Maximum>1.00</Maximum>
</CostPerTest>
<SumCostOfAllTests>100.0</SumCostOfAllTests>
<ModulesPerTest>
  <Minimum>1</Minimum>
  <Maximum>3</Maximum>
</ModulesPerTest>
<CoveragePerModule>
  <Minimum>0.20</Minimum>
  <Maximum>1.00</Maximum>
</CoveragePerModule>
</configuration>
<!-- =================== -->
<!-- END OF CONFIGURATION -->
<!-- =================== -->
</simulation>
```

Figure 5. Sample Configuration XML File

4.2 Model processing

Prior to the start of a configuration run (set of trials), a failure deck is created based on the relative failure rates of modules within the system. Similar to a deck of playing cards, modules appear in the failure deck based on their standing relative to the minimum failure rate in the system; thus, if the minimum failure rate across the system is 0.2, a module with a failure rate of 0.6 will appear three times within the failure deck. The same deck is employed across all trials in a configuration run to simulate the relative appearance of failures in a physical system. No a prior assumption was made in the mathematical analysis (Section 3) about the number of, and the simulation code reflects this versatility.

Prior to the start of a trial, a test deck with one entry for each test is created (copied) from the system configuration. Strategies (best-next, best-next-two, random, and worst) consume this list as the “next” choice is executed; thus, as a test is executed it is removed from the deck, insuring that no test will be executed more than once per trial. This also reduces the search space for the next test. A new test deck must be generated with each trial.

A single trial is processed in the following manner:

1. All module b_i are initialized from failure rate data.

2. All module coverages are established; these are either duplicated from the previous trial or randomized subject to the same configuration parameters.

3. Some p number of modules (where $0 \le p \le n$) are selected from the failure deck and a defect is planted in each module. It is possible (and interesting) to run the simulation with no defects planted.

4. A test is chosen based on a simple strategy (e.g., best, random, or worst)

5. This test is applied to the system object

6. All affected b_i are updated based on the outcome of (5)

7. If there is still a test in the test deck, return to (4)

Although we are interested in improving test strategies under fixed-cost constraint, in these trials we simply execute until the set of tests is exhausted. The motivation for this approach is that by allowing the simulation to process all tests available, we also gain insight into the effectiveness of the given test suite; this line of investigation will be pursued in future studies.

4.3 Simulation results

Using a 2 GHz Intel processor, a simulation of 300 trials using the best-next, best-next-two, random and worst strategies required about 17 minutes (for all four) using a randomized configuration of 40 modules and 100 tests, with one defect planted. The zero-defect simulations required about the same execution. In all of these cases, the initial probability distribution was to set each module to b_i=0.5 (maximum entropy).

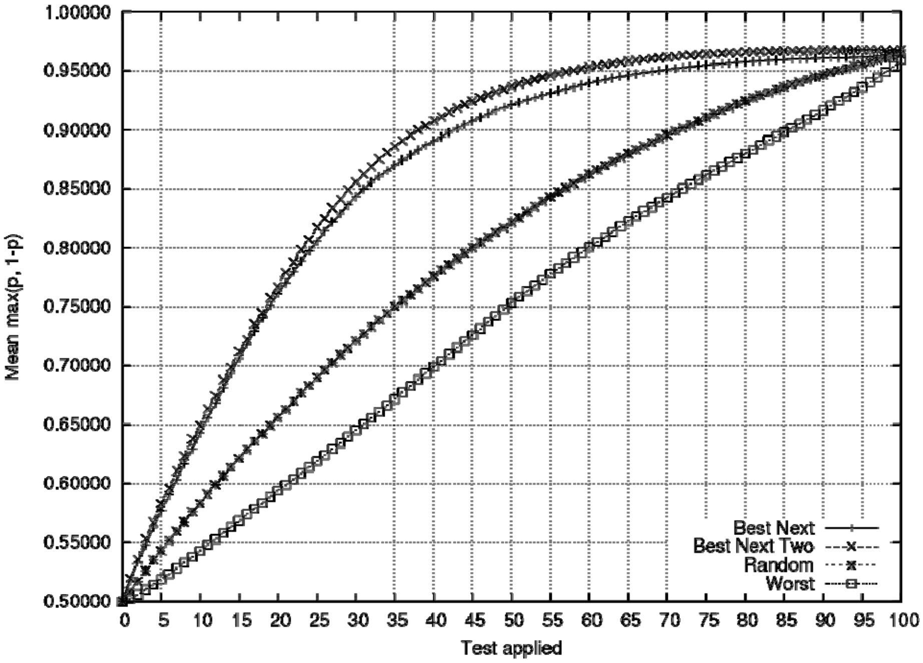

Figure 6. Comparison of Mean q_i among Simulations of a 40-module, 100-test System with One Defect Planted (using the one-step (best-next), two-step (best-next-two), random and worst cases)

Comparisons between the one-step and two-step approach (Figure 6) show little difference in the first 25 or so tests in terms of the mean maximum probability q_i—suggesting that the one-step approach would yield an acceptable return on a fixed budget or schedule for testing *for the randomized system configurations tested.* More realistic scenarios may show more significant divergence between one-step and two-step (and by extension, *k*-step) methods.

Similar comparisons in terms of information entropy (Figure 7) show more displacement between the one-step and two-step methods, though both methods are clearly superior to the random and "worst" case methods. In a similar simulation configuration but with no defect (Figure 8), the information entropy shows similar descent. At the tail end of the testing process, though, the steady-state *H* is lower in the no-defect case (Figure 8) than in the one-defect (Figure 7) case.

Although the best-next-two strategy appears somewhat better than the best-next strategy for the fixed-cost constraint, the CPU time required for this two-step simulation expanded roughly by a factor of four on a per-trial basis, which is consistent with Equation 3.8 and Equation 3.15.

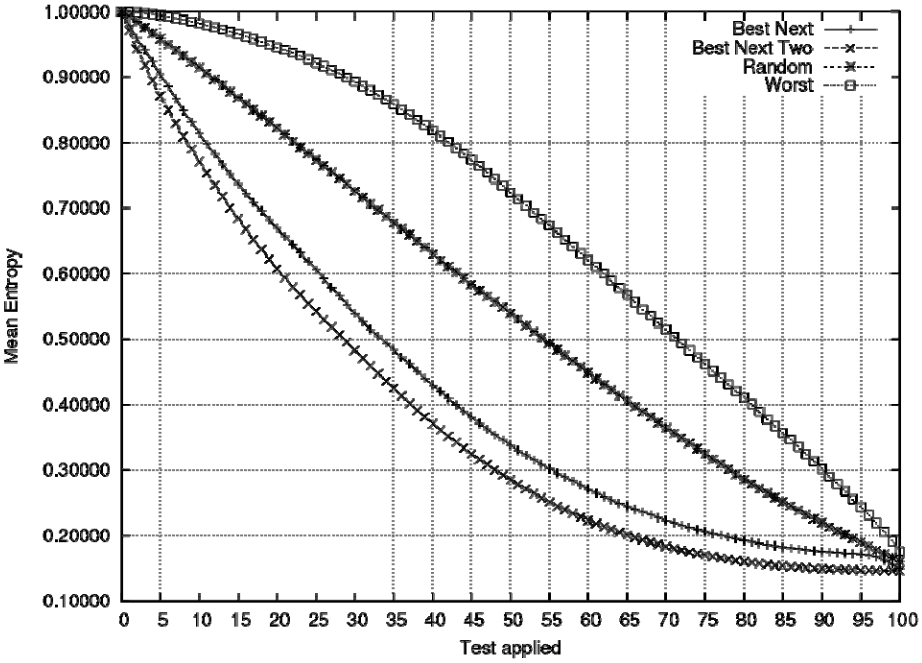

Figure 7. Comparison of Mean Entropy *H* among Simulations of a 40-module, 100-test System with One Defect Planted (using the one-step (best-next), two-step (best-next -two), random and worst cases)

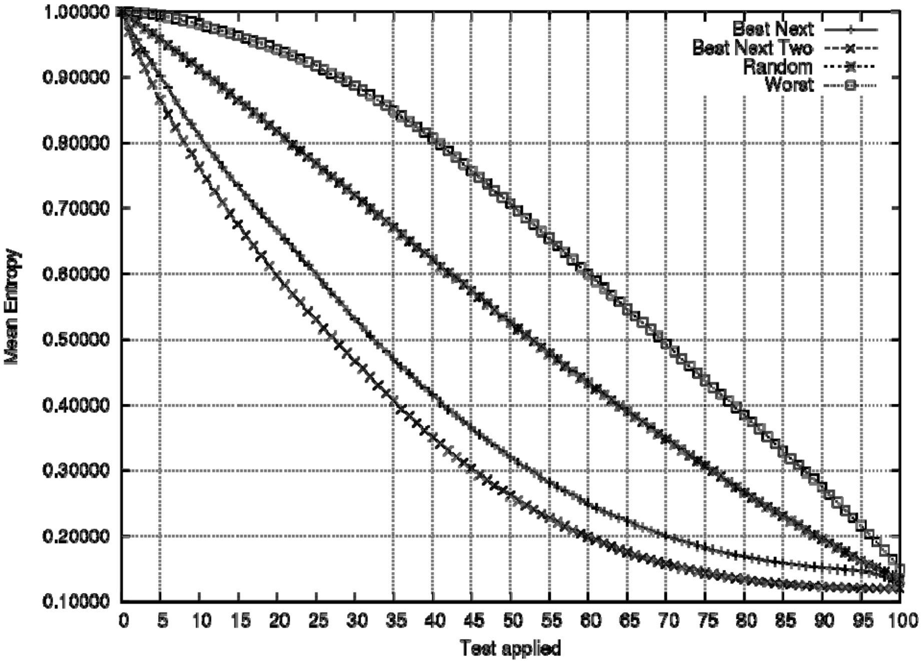

Figure 8. Comparison of Mean Entropy *H* among the NO DEFECT Simulations of a 40-module, 100-test System (showing only slightly lower entropy values for all cases, though slope of descent is similar in all cases)

5. Conclusions and Future Work

In this study, we have developed a simple, effective framework to examine the testing of complex systems. The idealized numerical experiments demonstrate that a simple best-next approach is computationally efficient if not optimal, though the convergence to a correct diagnosis appears to be very close to a two-step approach. Further investigation with more exhaustive *k*-step testing strategies should confirm that for many cases of interest the simple best-next approach yields acceptable results for a fixed-cost constraint.

A novel aspect of this approach is the focus on tests as distinct objects providing information about modules. Much of the previous work in this area has focused on knowledge of the initial distribution of failure rates, requiring almost perfect knowledge of these a priori data in order to be effective (Butterworth, 1972; Ben-Dov, 1981). In terms of software testing, previous studies have relied on some knowledge of the internal structure of components or reusable objects, or near-perfect knowledge of the module interconnections (Rothermel, Untch & Harrold, 2001; Mao & Lu, 2005). Here, we assume only that our system has some test suite; these tests may be derived from requirements documents, from formal system acceptance plans, or from daily systems operations. Application of this model requires only that we be able to characterize these tests in terms of approximate coverages on units of replacement.

Although a variable cost per test is accounted for in the simulation code, the runs presented in this paper assumed a constant cost per test. In effect, we assume a unit cost per test, such that the number of tests becomes the associated cost. Further simulation work with more realistic configurations of modules, tests and coverages should yield more insight into operational diagnostic and regression problems.

The treatment of tests as distinct objects readily enables use of this approach for simulation even when no bug or defect is known to be present. This testing scenario, which commonly follows a major system upgrade or engineering change to a system, is often referred to as regression testing. This zero-defect case may also be useful to evaluate the quality of a test suite with respect to all states of a system. In the zero-defect cases run in this study, the mean entropy (Figure 8) and maximum probability (not shown) do not go to 0 and 1, respectively, as we would expect if all states were reachable from the test suite. This line of research would be particularly well-suited for the classic regression or test-retest problems.

List of References

Athans, M. (1987). Command and control (C2) theory: A challenge to control science. *IEEE Transactions on Automatic Control*, *32*(4), 286-293.

Bardford, L., Kanevsky, V., & Kamas, L. (2004). Bayesian fault diagnosis in large-scale measurement systems. In *Proceedings of the IMTC 2004: Instrumentation and Measurement Technology Conference* (pp. 1234-1239). Como, Italy: IEEE.

Barlow, R.E., & Proschan, F. (1965). *Mathematical theory of reliability.* Philadelphia, PA: John Wiley and Sons.

Ben-Dov, Y. (1981). Optimal testing procedures for special structures of coherent systems. *Managment Science*, *27*(12), 1410-1420.

Birnbaum, Z., Esary, J., & Saunders, S. (1961). Multi-component systems and structures and their reliability. *Technometrics*, *3*(1), 55-77.

Boivard, R.L. (1961). Characteristics of optimal maintenance policies. *Management Science*, *7*(3), 238-253.

Brazet, M.D. (1993). AEGIS ORTS-the first and future ultimate integrated diagnostics system. *Aerospace and Electronic Systems Magazine*, *9*(2), 40-45.

Butterworth, R. (1972). Some reliability fault-testing models. *Operations Research, 20*(2), 335-343.

Caruso, J. (1995). The challenge of the increased use of COTS: A developer's perspective. In *Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems* (pp. 155-159). Santa Barbara, CA: IEEE.

Corman, T.H., Leiserson, C. ., Rivest, R.L., & Stein, C.R. (2002). *Introduction to algorithms* (2nd ed.). Boston: MIT Press.

Cover, T.M., & Thomas, J.A. (1991). *Elements of information theory.* New York: John Wiley and Sons.

Dalcher, D. (2000). Smooth seas - rough sailing: The case of the lame ship. In *Proceedings of the Seventh International Conference on Engineering of Computer Based Systems (EBCS 2000)* (pp. 393-395). Edinburgh, Scotland: IEEE.

Fishman, G.S. (1990). How errors in component reliability affect system reliability. *Operations Research*, *38*(4), 728-732.

Garey, M. (1972). Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*, *23*(2), 173-186.

Leung, H., & White, L. (1991). A cost model to compare regression test strategies. In *Proceedings of the Conference on Software Maintenance* (pp. 201-208). Sorrento, Italy: IEEE.

Mao, C., & Lu, Y. (2005). Regression testing for component-based software systems by enhancing change information. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC’05)* (pp. 1-8). IEEE.

Moore, E., & Shannon, C. (1956a). Reliable circuits using less reliable relays, Part I. *Journal of the Franklin Institute*, 191-208.

Moore, E., & Shannon, C. (1956b). Reliable circuits using less reliable relays, Part II. *Journal of the Franklin Institute*, 281-298.

Rothermel, G., Untch, R.H., & Harrold, M.J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering, 27*(10), 929-948.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*, 379-423; 623-656.

Sobel, M., & Groll, P. (1966). Binomial group-testing with an unknown proportion of defectives. *Technometrics, 8*(4), 631--656.

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*(4857), 1285-1293.

Tsai, W. (2001). End-to-end integration testing design. In *Proceeedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC)* (pp. 166-171). Chicago, IL: IEEE.

von Neumann, J. (1952). Probabilistic logics and synthesis of reliable organisms from unreliable components. In *Automata Studies (AM-34)* (pp. 45-98). Princeton, NJ: Princeton University Press.

Weyuker, E. (1998). Testing component-based software: a cautionary tale. *IEEE Software, 15*(5), 54-59.

White, L., & Leung, H. (1992). A firewall concept for both control-flow and data-flow in regression integration testing. In *Proceedings of the Conference on Software Maintenance* (pp. 262-270). IEEE.

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for DoD

- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy’s Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes

- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org