Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Multi-Objective Decision Making on Reliability Growth Planning for In-Service Systems

Tongdan Jin
Ingram School of Engineering
Texas State University-San Marcos
San Marcos, TX 78666, USA
tongdan_jin@yahoo.com

Haibo Wang
Division of International Business & Technology Studies
A.R. Sanchez Jr. School of Business
Texas A&M International University, Laredo, TX 78041,
hwang@tamiu.edu

***Abstract*— New designs prioritized for the fast time-to-market usually can not carry out sufficient in-house reliability growth testing due to the stringent delivery deadline. Reliability improvement for those systems can be achieved by implementing corrective actions (CAs) on in-service systems. In this paper, three types of effectiveness functions are proposed to measure the reduction rate of failure modes given different CA resources. Integrated with the effectiveness function, a new failure intensity model is proposed for predicting the mean-time-between-failures (MTBF) of field systems. Finally, a multi-objective optimization model is formulated to maximize the system reliability and to minimize the reliability uncertainty with the constraint of the CA resources. Genetic Algorithms combined with greedy heuristic are applied to search the optimal CA decisions that lead to the maximum reliability growth while minimizing the reliability uncertainty. Results show that the proposed reliability growth program can effectively guide decision-makers to find the most effective corrective actions for achieving the reliability goal for a large fleet of in-service systems. Throughout paper, systems and products will be used interchangeably.**

Keywords—*reliability growth, optimization, corrective actions*

I. INTRODUCTION

In semiconductor, telecommunication, and other high-tech industry, the requirements for highly reliable systems render traditional reliability modeling and optimization algorithms obsolete due to the shift from the centralized manufacturing environment to the distributed manufacturing paradigm. Take the semiconductor Automatic Testing Equipment (ATE) for example. Often the design of this type of multi-million dollars system is undertaken by engineers in the US while hardware assemblies, software development, and system integration are subcontracted to low cost regions outside the US. The advantage of the distributed manufacturing paradigm allows the manufacturer to leverage the global resources to expedite the release time of new products. Another advantage is the cost reduction that makes the product more competitive in the global market.

Figure 1 depicts a generic ATE system configured with eight instrument modules to test two devices at the same time. The system configurations may vary depending on the device to be tested. There are five types of PCBs (printed-circuit-boards) in the system: HSD (high-speed-digital), analog, DC (direction current), RF (radio frequency) and Support modules. The device interface board provides electrical connections between the instrument modules and the devices. The tester computer executes the test program to determine whether the device passes or fails the design specifications.

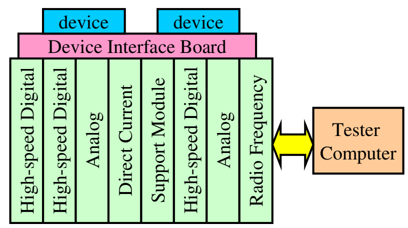

Figure 1 A generic ATE System

Two grand challenges emerged from the distributed manufacturing environment: the increase of non-hardware failures, and the compressed system design cycles. As the system complexities continue to increase, non-hardware failures become a prevalent issue and could dominate field returns, especially in the early system shipment phase. Hardware failures are mainly for component failures such as those electronic devices used in the PCB module. Non-hardware failures, also known as non-component failures, include design weaknesses, software bugs, manufacturing defects, and usage related issues. On the other hand, the rapid advance in technology makes electronic systems obsolete every five to seven years. It forces the manufacturer to compress the design cycle in order to release the new product in the shortest time. The distributed manufacturing paradigm may reduce the product cost and accelerate the delivery time, but it may cost more to the manufacturer or the customers because of the escalated field returns or the unexpected system downtime events.

These challenges are increasingly dominating the design, manufacturing, and deployment of complex electronic systems when systems are designed and developed in a compressed schedule. Hence they deserve more attention from the design and manufacturing communities. Although these issues are brought up by ATE systems, similar challenges exist in other complex hardware/software system domains such as robotic & mobile systems [1, 2], WSN [3, 4], telecommunication equipment [5, 6], and computer networks [7, 8]. The study by Carlson & Murphy [2] found that the current reliability of UGVs (Unmanned Ground Vehicles) in field environments is

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

low, between 6 and 20 hours MTBF (mean-time-between-failures), and the system availability is only 50%.

Reliability growth testing (RGT), also known as Test-Analyze-And-Fix, can be applied to new system design if the system level test is not expensive or extended in-house testing is permitted. The idea of RGT dates back to Duane [9] when he was in charge of monitoring the lifetime of aircrafts. Later on Crow [10] found that reliability growth curve can be modeled by a Non-Homogenous Poisson Process (NHPP). His findings eventually were summarized and published as the popular Crow/AMSAA model. Since then, significant research activities have been carried out in RGT [11-16]. For instance, Campbell [11] proposed an efficient approach to the allocation of subsystem test time based on a heuristic approach. Benski and Cabau [12] applied design of experiment for RGT. Xie and Zhao [13] proposed a graphical approach to predict the reliability growth based on the Duane model. Coit [14] generalized Campbell's problem to optimize the RGT procedure by incorporating the test time as well as the testing cost. Recently, Krasich et al. [15] and Krasich [16] have described accelerated RGT procedure during product development process. Researchers in [17, 18] have extended NHPP models from the hardware domain to the software reliability growth modeling. Nevertheless, the implementation of RGT becomes increasingly difficult in today's fast-paced manufacturing environment where the product design, manufacturing, integration, and deployment are distributed spatially and temporally.

This paper aims to propose a novel reliability growth planning (RGP) method to improve in-service system reliability. This is particularly imperative for developing complex systems under the compressed design cycle; yet high system reliability is still required by customers. Different from RGT, the idea of RGP is to drive the reliability growth after the product is shipped to the field. Generally speaking, the system manufacturer collects the field failure data, analyze failure mechanisms, and implement corrective actions so that the system failure rate is reduced and the reliability goal is achieved. To achieve that objective, optimizations are formulated to maximize the growth of system reliability given the limited corrective action resources.

This rest of the paper is organized as follows. Section 2 provides a brief review on the reliability challenges emerged from the distributed manufacturing environment. In Section 3, CA effectiveness functions are proposed to link the CA cost with the anticipated reduction rate of failure modes. In Section 4 analytical models are derived to estimate the expected system reliability growth based on the CA effectiveness functions. In section 5, a multi-objective optimization model is formulated to address the optimal CA allocation problem. In Section 6 the new GRP algorithm is applied to ATE systems to demonstrate the performance of the method. Section 7 concludes the paper with some remarks on the future research.

II. Challenges in Reliability Management

Our paper has the following contributions: providing realistic reliability predictions for new system designs necessary for project managers to determine the development schedule; supporting the optimization for reliability growth planning in conjunction with the constrained recourses; providing design feedback to the manufacturing community for enhancing design-for-reliability. We will approach these objectives from two aspects: 1) hardware & non-hardware reliability prediction; and 2) compressed system design cycles.

A.. Hardware & Non-Hardware Reliability Prediction

Figure 2 shows the MTBF of a type of ATE system configured with 20 modules, which is more close to the actual system. If the reliability prediction only considers hardware failures, the system will reach 3,000 hours MTBF at the end of 24 months. The system MTBF would be 1,800 hours if the prediction incorporates hardware, software, and NFF (No-Fault-Found) failures. NFF is such type of failure that the customer experienced the system failure while the failure mode can not be duplicated by the repair center. In reality, the system MTBF is only 1,100 hours at the end of 24 months considering all hardware and non-hardware failures. Obviously, if the reliability prediction is made based on hardware failures only, it will overestimate the reliability performance during the design phase which leads to an optimistic prediction value. The consequence is that project mangers will withdraw the support resources earlier from the current project.

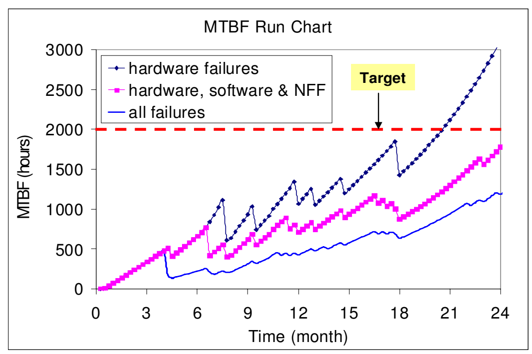

Figure 2 Actual System MTBF vs. Predictions

Accurate reliability prediction is difficult for complex systems, but the prediction is important to the system designer. The prediction assists the design engineers in identifying design weaknesses, planning the reliability timetable, and implementing corrective actions. Optimistic reliability prediction could mislead the system manufacturer to allocate less support recourses after the field deployment. Compared with RGT, few papers have been published to address RGP related to design, manufacturing, software, and process issues for electronic equipment. Johnson & Gullo [19] and Gullo [20] reported an in-service reliability prediction tool HIRAP developed by Honeywell engineers. HIRAP breaks failures into seven categories among which category 1-5 consist of hardware failures and categories 6 and 7 are used to address process, manufacturing and design errors.

B. Compressed System Design Cycle

New designs must be transformed into competitive yet reliable products in order to gain the competitive advantage in the market. A common strategy is to reduce the design cycle so that new product can be delivered to customers in the shortest time. Complex systems like ATE are always designed in modality that allows for upgrading. The ATE manufacturer often delivers the new system at time right after the completion of several basic system modules. Meanwhile, advanced modules are still in the development or even concept phase with different release time. Customers who purchased the basic system can test current semiconductor devices. If new devices are designed and require more advanced testing features, customers have the flexibility to upgrade the system configuration by plugging advanced ATE modules. In such a compressed design cycle, the ATE manufacturer has no time to implement the RGT during the design and development phase.

III. Corrective Action Effectiveness Function

A.. Corrective Actions (CAs)

Corrective actions are often used to drive the reliability growth of in-service systems. In practice two types of CAs are often used: retrofit and engineering change order (ECO). Retrofit is a process to eliminate known failure modes by proactively replacing field modules that will fail, but not failed yet. Retrofit can also be applied to software upgrade by issuing new versions of the software to customers. ECO is a counter-measure generally implemented in the repair center to upgrade the field return modules by proactively replacing certain components or improving processes prior to the occurrence of failures. Therefore ECO is also called self-PUP (Self-Product-Upgrade) in the sense that the CA is applied only after the module failed and is returned for repair. Retrofit process can immediately eliminate the failure mode from in-service systems, while the ECO is a gradual improvement process.

The cost of retrofit generally is much higher than ECO because the former involves extra personnel, logistics and materials for on-site CA implementation. During the retrofit process, dedicated personnel and a certain amount of the spare modules are required and dispatched to the customer site for the proactive replacement. This increases the overhead of labors, logistics and spare inventory. The replaced modules are sent back to the repair center for root cause removal. These fixed modules are then sent to another customer site for another wave of retrofit activities. The process is repeated until all field systems have completed the retrofit service. On the other hand, the cost of ECO is much lower because it only involves the trouble-shooting time, spare components, module shipping, and storage costs.

B. CA Effectiveness Function

The CA effectiveness can be evaluated by the expected failure reduction rate upon the implementation of the CA. For instance, the current failure rate for a type of failure mode is 2×10^{-8} faults/hour. Upon the implementation of CA, the rate is reduced to 5×10^{-9} faults/hour, then the CA effectiveness can be expressed as 0.75, i.e. $(2\times10^{-8}-5\times10^{-9})/(2\times10^{-8})=0.75$. Typical effectiveness value is between zero and one depending on whether retrofit or ECO is adopted. More specifically, the CA effectiveness depends on the amount of budget allocated for that failure mode. For a specific failure mode, it is reasonable to assume that the more the budget is allocated for the CA activity, the higher the CA effectiveness will be.

In this paper, an analytical model is proposed to characterize the CA effectiveness in terms of percentage of failure mode reduction verse the CA cost. The maximum CA effectiveness is one if a particular failure mode is completely eliminated from in-service systems, and the minimum is zero if no CA is applied. The effectiveness function aims to link the failure mode reduction rate with the amount of the budget used against that failure mode. The following model is suggested to capture the relationship between CA effectiveness and cost

$$h(x)=\left(\frac{x}{c}\right)^b \tag{1}$$

In equation (1), x represents the amount of the budget allocated for a particular CA. Notice that b and c are model parameters and they are all possible values. These parameters can be estimated based on historical CA data or from predecessor products combined with the field system population. For a particular failure mode, the value of c actually is equal to the retrofit cost assuming all field system receive retrofit service. This can be easily justified from the fact that when $x=c$, $h(x)=1$ meaning this type of failure mode is completely eliminated using retrofit. b is the shape parameter that controls the shape of the effectiveness function. Depending on the value of b, three types of effectiveness functions are available: linear, quadratic, and power functions as shown in Figure 3. The value of b often can be estimated based on historical effectiveness data and the nature of the failure mode.

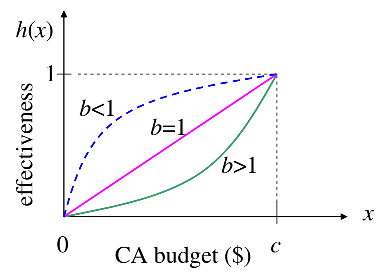

Figure 3 CA Effectiveness Function

Figure 3 depicts the effectiveness function for different values of b given the same c. When b=1, the general model is simplified to a linear function. This model is relatively simple, yet it has wide applications due to its mathematical convenience and simplicity. For many practical problems, the actual CA effectiveness can be approximated by the linear model. If b>1, $h(t)$ becomes a power function. The rational model represents the effectiveness when the parameter b<1. This means the effectiveness decreases once the CA budget reaches certain amount of money. Typical examples include software upgrade.

IV. System Reliability Estimate with CA

A. System Failure Rate Function

Due to the limited physical area on a PCB, redundancy usually is not adopted at component level when designing an instrument module of ATE systems. As a result, any component or non-component failure will cause the module to fail in certain electrical characteristics. In other words the module is functional only if no failures of components and non-components are observed. Therefore the instrument failure rate is equal to the summation of the failure rates of all components and the non-components comprising the module. Without loss of generality, we propose the system/module failure rate function as follows

$$\lambda_s(t) = \sum_{i=1}^{k} n_i \lambda_i(t) + \sum_{i=k+1}^{m} \lambda_i(t) \tag{2}$$

$\lambda_s(t)$ = failure rate for the system

$\lambda_i(t)$ = failure rate for component type i for i=1,2, …, k

$\lambda_i(t)$ = failure rate for non-component type i and for i=k+1, k+2, …, m

Equation (2) comprehends both component and non-component failures. Here k is the number of component types and n_i is the quantity for type i used in the system. The second summation term represents the cumulative failure rate for five non-components types as mentioned previously. Notice that i=k+1 for design, k+2 for manufacturing, k+3 for software, k+4 for process, and m=k+5 for NFF. Since the value of m usually is large, by the Central Limit Theorem, $\lambda_s(t)$ tends to be a Gaussian process with corresponding mean and variance as follows,

$$\mu_s(t) = E[\lambda_s(t)] = \sum_{i=1}^{k} n_i E[\lambda_i(t)] + \sum_{i=k+1}^{m} E[\lambda_i(t)] \tag{3}$$

$$\sigma_s^2(t) = \mathrm{var}(\lambda_s(t)) = \sum_{i=1}^{k} n_i^2 \,\mathrm{var}(\lambda_i(t)) + \sum_{i=k+1}^{m} \mathrm{var}(\lambda_i(t)) \tag{4}$$

Equation (4) is derived assuming all $\lambda_i(t)$ are mutually independent; meaning correlations between different component types (and/or non-component types) can be ignored. This assumption is reasonable in most applications where failures for component and non-component are independent.

Quite often the lower bound of the product reliability, which is the reciprocal of the upper bound of the failure rate, is used to characterize the reliability of the product. Considering the normal approximation, the upper bound of the PCB failure rate can be expressed as:

$$\lambda_{s,upper}(t) = Z_{1-\alpha}(\mu_s(t), \sigma_s^2(t)) \tag{5}$$

Where 1-α is the confidence interval for the system reliability. Usually, this upper bound can be improved by approaches such as Design for Reliability (DFR) through selecting reliable components, reducing temperature variations, and mitigating non-component issues in early development phase. In this paper, we are planning to improve the in-service system reliability assuming the system has already gone through the DFR process.

B. Integrating CA with the Failure Rate Function

Intuitively, recourses of corrective actions are often prioritized to attack failure modes with high failure rate profiles. If the failure rate remains at a low level or exhibits a downturn trend, it is generally not recommended for implementing CAs. To obtain a generalized system failure rate estimate under CA activities, we assume that all existing failure modes could potentially receive CA regardless of the failure rate profile. Based on equation (2), the system failure rate upon the implementation of CA becomes

$$\begin{aligned}\lambda_{s,CA}(t) &= \sum_{i=1}^{k} n_i (1-h_i(x_i))\lambda_i(t) + \sum_{i=k+1}^{m} (1-h_i(x_i))\lambda_i(t) \\ &= \sum_{i=1}^{k} n_i g_i(x_i)\lambda_i(t) + \sum_{i=k+1}^{m} g_i(x_i)\lambda_i(t)\end{aligned} \tag{6}$$

Notice $g_i(x_i)$=1-$h(x_i)$, representing CA ineffectiveness for failure mode i. In the best scenario (e.g. $h(x_i)$=1 for retrofit), failure mode i is completely eliminated by the CA because $g_i(x_i)$=1-$h_i(x_i)$=0. Substituting equation (6) into equations (3) and (4), the expected system failure rate and the associated variance upon CAs are given as

$$E[\lambda_{s,CA}(t)] = \sum_{i=1}^{k} n_i g_i(x_i) E[\lambda_i(t)] + \sum_{i=k+1}^{m} g_i(x_i) E[\lambda_i(t)] \tag{7}$$

$$\mathrm{var}(\lambda_{s,CA}(t)) = \sum_{i=1}^{k} n_i^2 g_i^2(x_i)\,\mathrm{var}(\lambda_i(t)) + \sum_{i=k+1}^{m} g_i^2(x_i)\,\mathrm{var}(\lambda_i(t)) \tag{8}$$

Equations (7) and (8) are important expressions to predict the system failure rate upon the implementation of CA. It also links the CA cost, x_i, to the improvement of the system reliability through the elimination of failure mode i. Hence decision-makers can rely on this model to allocate appropriate resources to obtain the maximum reliability growth. In the following, a multi-objective optimization problem (MOOP) will be formulated to determine which failure modes will receive CA activities such that the system reliability is maximized.

V. OPTIMIZATION FORMULATION

Reliability growth planning aims to allocate appropriate CA resources to remove or mitigate critical failure modes so that the system can achieve the anticipated reliability goal. This is equivalent to minimizing the system failure rate by appropriately allocating resources to maximally reduce known failure modes.

On the other hand, engineers planning the CA process almost always assume that the estimated model parameters are exact values. The actual failure rate of one particular failure mode may vary over time, therefore the uncertainties in λ_i should be incorporated into the decision process. This can be achieved by formulating a multi-objective optimization model in which both the mean and the variance of $\lambda_{s,CA}(t)$ will be minimized. Now the MOOP, denoted as P1, is given as

Problem P1:

Min $f(\mathbf{x})$={ $E[\lambda_{s,CA}(\mathbf{x};t)]$, $\mathrm{var}(\lambda_{s,CA}(\mathbf{x};t))$ } (9)

Subject to:

$$\sum_{i=1}^{m} x_i \le C \tag{10}$$

$$x_i \ge 0 \quad \text{for } i=1, 2, \ldots, m \tag{11}$$

Notice that x_i is the CA budget allocated for failure mode i and it is decision variable. The constraint is a linear function with maximum available budget C for all CA processes. Objective functions are given by equations (7) and (8) respectively.

It would be particularly beneficial to reduce the variability of the system failure rate if sources or quality of the estimated model parameters differ appreciably within the system. Coit [14] made an argument which can be used to justify the formulation of P1. Considering a scenario where one failure mode shows great promise for improvement as indicated by high λ_i but there is significant estimation variability. The other failure mode indicates less fluctuation for the improvement, but the model parameters are more deterministic. A corrective action strategy which ignores the uncertainty may put unwarranted resources on one failure mode with potential improvement. This resource allocation could be promising, but it could also be risky. The risk could be mitigated if a more conservative plan can be implemented which assures the resource allocation to other types of failure modes. Obviously P1 is formulated to accommodate the parameter variability as discussed above.

VI. APLICATION TO ATE SYSTEMS

The proposed MOOP reliability growth algorithm will be demonstrated on a fleet of field ATE systems. The current system MTBF is only 838 hours based on the field failure data, while the target system is 2,400 hours. Data collected from the repair center indicates that all defective returns can be classified into 16 major failure modes. Among those, eleven failure modes belong to hardware issues. The other five failure modes belong to non-hardware issues which can be further classified into design weakness, software bugs, manufacturing defects, process issue, and NFF. For each failure mode, the corresponding failure rate and the associate variance are estimated and listed in Table 1. Meanwhile, CA effectiveness coefficients *b* and *c* are also estimated and listed in the table. Without loss of generality, the linear effectiveness model is assumed for all types of failure modes.

Table 1: Parameters for Effective Functions and Failure Modes

FM i	Failure Category	n_i	c_i ($)	b_i	$E[\lambda_i(t)]$	var($\lambda_i(t)$)
1	Comp 1	30	500,000	1	6.35E-7	5.02E-14
2	Comp 2	25	30,000	1	3.04E-6	9.18E-14
3	Comp 3	60	40,000	1	1.89E-7	4.13E-16
4	Comp 4	45	75,000	1	2.80E-6	1.64E-12
5	Comp 5	30	370,000	1	3.36E-6	7.42E-13
6	Comp 6	26	45,000	1	4.06E-6	4.06E-12
7	Comp 7	50	230,000	1	2.65E-6	1.03E-12
8	Comp 8	30	150,000	1	1.15E-6	2.06E-13
9	Comp 9	10	10,000	1	1.04E-5	1.18E-11
10	Comp 10	6	20,000	1	1.23E-5	3.21E-12
11	Comp 11	25	10,000	1	1.08E-6	2.56E-13
12	Design	1	300,000	1	7.19E-5	3.22E-10
13	Software	1	200,000	1	1.42E-4	1.68E-10
14	Mfg	1	350,000	1	1.98E-5	7.74E-11
15	Process	1	150,000	1	1.39E-4	8.03E-10
16	NFF	1	180,000	1	1.09E-5	1.80E-12

Note: FM=Failure Mode, Mfg=manufacturing

The optimal RGP program given in Problem P1 is searched using the Genetic Algorithm (GA) embedded with the greedy heuristic. Figure 4 describes the computational GA process. Originally proposed by Holland [21], GA is a probabilistic method that uses a set of designs rather than a single design during the search process. The GA in this example used crossover and mutation operators. The crossover is given 50% probability while and the mutation is given 10% probability. Nevertheless, the main deficiency of GA is the randomness involved in the search process. To compensate this shortcoming, the gradient information of two objective functions is calculated for the members of the new population. The integration of GA and the gradient method generates an intelligent or guided, instead of random, search mechanism. Excessive CA budgets are distributed to the genes of the member that shows the greatest improvement of the objective functions based on the gradient information.

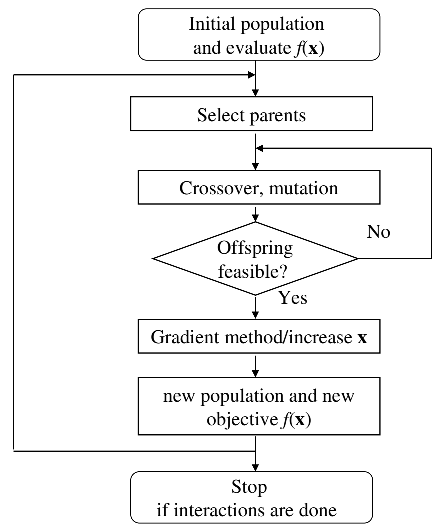

Figure 4 the GA Search Procedure

Goal programming, goal attainment and Pareto optimality are three common methods to search the solution for MOOP problems. Goal programming and goal attainment are effective if the decision-maker has some prior knowledge of the scope of objective functions and the relative weights of each objective function. In most cases, problems are formulated when there is little knowledge of the system level implications. Hence goal programming and goal attainment become less effective compared to the Pareto optimally. The Pareto optimal solution turns out to be an alternative approach for solving MOOP because it can provide a set of non-dominant solutions. The decision-maker can select a best-compromised one from a set of non-dominant solutions. In this paper the Pareto optimality will be used to solve problem P1.

Pareto optimal solutions can be obtained via the weighted method that transforms the original MOOP into a single objective problem by assigning an appropriate non-negative weight, w_i for i=1 and 2, to each objective function. The sum of these weights does not need to be unity. However, if there are two objectives, it is a common practice to choose weights such that $w_1+w_2=1$. Now Problem P1 can be transformed into the following single objective optimization

Problem P2:

Min $z(\mathbf{x}) = w_1 E[\lambda_{s,CA}(\mathbf{x};t)] + w_2\left(\mathrm{var}(\lambda_{s,CA}(\mathbf{x};t))\right)^{\frac{1}{2}}$ (12)

Subject to:

$$\sum_{i=1}^{m} x_i \le C \qquad (13)$$

$$w_1 + w_2 = 1 \qquad (14)$$

Solving the single objective problem in P2 could obtain one non-dominant solution. By appropriately changing the weights w_1 and w_2, a set of non-dominant solutions will be obtained. A set of solutions are obtained by varying w_1 from 0.1 to 0.9 as listed in Table 2. Based on the solution set, values of two objective functions are plotted in Figure 5 with horizontal-axis representing $E[\lambda_{s,CA}(\mathbf{x};t)]$ and the vertical-axis for $\mathrm{var}(\lambda_{s,CA}(\mathbf{x};t))$. Among nine solutions in Table 2, three non-dominant solutions are identified and labeled as *A*, *B*, and *C* respectively in Figure 5.

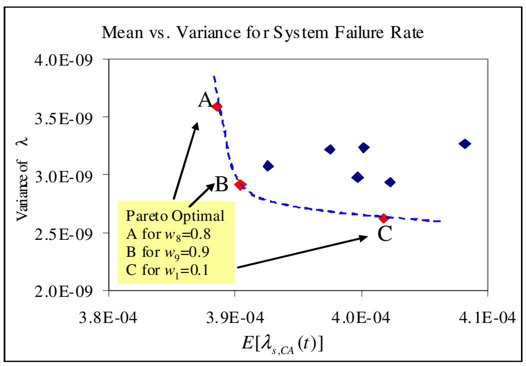

Figure 5 Pareto Optimal Solutions

Now the decision-maker can choose a best comprised solution from the non-dominant solution set. If he/she prefers risk-averse option, then *C* is superior to *A* and *B* because it yields the smallest variance of system failure rate. If a risk-neutral design is preferred, then either *A* or *B* can be chosen because they have a lower failure rate than *C*, but the uncertainty in the system failure rate will be higher. If one needs a risk-averse design with high point system reliability, then *B* is the more ideal candidate compared to *A* or *C*.

VII. CONCLUSION

This paper proposes a multi-objective optimization model to minimize the system failure rate and the associated variance. The CA effectiveness function is able to link the failure reduction rate with the expenditure of CA resource. The Genetic Algorithm combined with greedy heuristic is applied to search the optimal solution in the constraint of the CA budget. The proposed algorithm is demonstrated on the reliability growth management of ATE systems. Results show that the proposed RGP algorithm can effectively guide decision makers to identify the critical failure modes and implement the CA for obtaining the reliability goal using limited resources.

RGP is becoming increasingly important as design schedules continue to shrink and the reliability testing budget is cut off. Under these constraints, reliability growth can not be effectively achieved through traditional reliability growth testing as it often requires extended in-house testing time with dedicated personnel. RGP, however, allows for the system reliability growth from in-house to the field operation. As such the new product will quickly gain the market share as well as win the customer satisfactions when the system reliability continues to improve. This in turn generates more revenues and field data for further improvement of the product reliability.

Table 2: Solutions Based on Weighted Method

Design variable	**w_1=0.1 (C)**	0.2	0.3	0.4	0.5	0.6	0.7	**0.8 (A)**	**0.9 (B)**
x_1	**0**	1,125	1,477	6,900	1,986	0	1,619	**1,072**	**16,281**
x_2	**28,902**	28,645	29,996	28,770	29,364	28,346	27,686	**29,327**	**29,500**
x_3	**39,344**	40,000	39,994	39,718	39,152	39,596	39,870	**39,103**	**39,333**
x_4	**73,741**	62,685	74,990	71,926	73,409	70,864	74,757	**73,318**	**74,477**
x_5	**9,999**	0	2,919	0	1,720	0	13,066	**7,264**	**9,115**
x_6	**39,565**	45,000	39,694	44,006	44,045	36,336	41,529	**43,991**	**44,250**
x_7	**99,828**	68,915	61,370	87,874	73,902	130,720	175,960	**40,592**	**68,853**
x_8	**58,145**	96,798	77,838	75,269	82,104	112,550	63,432	**125,300**	**67,316**
x_9	**9,757**	10,000	9,738	9,997	10,000	9,984	10,000	**9,874**	**9,792**
x_{10}	**19,513**	19,554	19,476	19,722	19,707	19,967	20,000	**19,911**	**19,583**
x_{11}	**9,759**	10,000	10,000	9,997	9,853	9,984	10,000	**10,000**	**9,906**
x_{12}	**0**	7,330	7,363	0	22,221	0	19,736	**13,008**	**0**
x_{13}	**112,880**	83,750	120,520	90,502	95,108	40,056	39,364	**127,860**	**145,400**
x_{14}	**11,846**	82,773	2,250	29,217	3,122	53,360	0	**0**	**0**
x_{15}	**63,433**	52,270	77,539	68,796	72,229	56,845	70,426	**58,767**	**60,538**
x_{16}	**72,300**	40,282	74,834	65,842	72,058	39,578	42,432	**50,101**	**54,016**
Total Cost	**649,012**	649,127	649,998	648,536	649,980	648,186	649,877	**649,488**	**648,360**
$E[\lambda_{s,CA}(t)]$	**4.02E-4**	3.98E-4	4.00E-4	4.00E-4	4.08E-4	3.93E-4	4.02E-4	**3.89E-4**	**3.90E-4**
$\mathrm{var}(\lambda_{s,CA}(t))$	**2.62E-9**	3.22E-9	3.23E-9	2.98E-9	3.26E-9	3.08E-9	2.94E-9	**3.59E-9**	**2.91E-9**

REFERENCES

[1] Belkhouche, F., Belkhouche, B., and Rastgoufard, P., "Parallel navigation for reaching a moving goal by a mobile robot," Robotica, vol. 25, 2007, pp. 63-74.

[2] Carlson, J., Murphy, R.R., "How UGVs physically fail in the field," IEEE Transactions on Robotics and Automation, vol. 21, no. 3, 2005, pp. 423-437.

[3] AboElFotoh, H.M.F., Iyengar, S.S. and Chakrabarty, K., "Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures," IEEE Transactions on Reliability, vol. 54, no. 1, 2005, pp. 145-155.

[4] Shrestha A., Xing, L., "Quantifying application communication reliability of wireless sensor networks," International Journal of Performability Engineering, vol. 4, no. 1, 2008, pp. 43-56.

[5] Jackson, D.S., Pant, H., and Tortorella, M., "Improved reliability-prediction and field-reliability-data analysis for field-replaceable units," IEEE Transactions on Reliability, vol. 51, no. 1, 2002, pp. 8-16.

[6] Khan, F., "Equipment reliability: a life-cycle approach," Engineering Management Journal, vol. 11, no. 3, 2001, pp. 127-135.

[7] Grassi, V., Patella, S., "Reliability prediction for service-oriented computing environments," IEEE Internet Computing, vol. 10, no. 3, 2006, pp. 43-49.

[8] Hamlili, A., "Reliability evaluation and prediction of improvable information and communication networks," Information and Communication Technologies (ICTTA '06), 2006, vol. 2, pp. 3587-3592.

[9] Duane, J.T., "Learning curve approach to reliability monitoring," IEEE Transactions on Aerospace, vol. 2, 1964, pp. 563-566.

[10] Crow, L.H., "Reliability analysis for complex, repairable systems," SIAM Reliability and Biometry, 1974, pp. 379-410.

[11] Campbell, C.L., "Subsystem reliability growth allocation," Proceeding of the 36th Technical meeting, Institute of Environmental Science, IES, Prospect,, IL, 1990, pp. 748-751.

[12] Benski, C., Cabau E., "Unreplicated experimental design in reliability growth programs," IEEE Transactions on Reliability, vol. 44, 1995, pp. 199-205.

[13] Xie, M., Zhao, M., "Reliability growth plot-an underutilized tool in reliability analysis," Microelectronics and Reliability, vol. 36, 1996, pp. 797-805.

[14] Coit, W.D., "Economic allocation of test times for subsystem-level reliability growth testing," IIE Transactions, vol. 30, 1998, pp. 1143-1151.

[15] Krasich, M., Quigley, J., and Walls, L., "Modeling reliability growth in the system design process," Proceedings of Reliability and Maintainability Symposium, 2004, pp. 424-430.

[16] Krasich, M., "Accelerated reliability growth testing and data analysis method," Proceedings of Reliability and Maintainability Symposium, 2006, pp. 385-391.

[17] Hwang S., Pham, H., "Quasi-renewal time-Delay fault-removal consideration in software reliability modeling, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 39 , no. 1, 2009, pp. 200 – 209.

[18] Inoue, S., Yamada, S., "Generalized discrete software reliability modeling with effect of program size," IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 37, no. 2, 2007, pp. 170-179.

[19] Johnson, B., Gullo, L., "Improvements in reliability assessment and prediction methods", Annual Reliability and Maintainability Symposium, 2000, pp181-187.

[20] Gullo, L., "In-service reliability assessment and top-down approach provides alternative reliability prediction method", Annual Reliability and Maintainability Symposium, 1999, pp. 365-377.

[21] Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975.

Dr. Tongdan Jin is an assistant professor in the Ingram School of Engineering at Texas State University-San Marcos. He was an assistant professor in the Department of Engineering, Math and Physics at Texas A&M International University between 2006 and spring 2009. Prior to that, he held a reliability engineer position for five years in Teradyne Inc., Boston, Massachusetts. He obtained the Ph.D. in Industrial & Systems Engineering and MS in Electrical & Computer Engineering from Rutgers University. He is the member of IEEE and INFORMS. He has more than 20 papers published in peer-reviewed journals and conference proceedings. He also serves as a peer reviewer for IEEE Transaction on Reliability, IEEE Transactions on Automation Science and Engineering, and International Journal of Product Development. His research interests include stochastic modeling, reliability estimation and optimization, and financial engineering.

Dr. Haibo Wang is an assistant professor of decision science in the Division of International Business and Technology Studies, A.R. Sanchez Jr. School of Business at Texas A&M International University. He earned a Ph. D. in Production Operations Management from the University of Mississippi. He has provided consulting services to many organizations in China and US. His researches have been published in such outlets as the European Journal of Operational Research, IEEE transactions on Control System Technology, International Journal of Flexible Manufacturing Systems, Journal of the Operations Research Society, Communications in Statistics, Journal of Heuristics, Journal of Combinatorial Optimization, Journal of Optimization Letters, and International Journal of Information Technology and Decision Making.