LA-UR- $0\mathcal{S}$-5484

Title: | Assessing System Reliability and Allocating Resources: A Bayesian Approach That Integrates Multi-Level Data

Author(s): | Todd L. Graves
Michael S. Hamada

Intended for: | Submission to Journal of Quality Technology

Los Alamos
NATIONAL LABORATORY
———— EST. 1943 ————

Assessing System Reliability and Allocating Resources: A Bayesian Approach That Integrates Multi-Level Data

Todd L. Graves and Michael S. Hamada

Statistical Sciences

Los Alamos National Laboratory

04.22.08 1700

Abstract

Good estimates of the reliability of a system make use of test data and expert knowledge at all available levels. Furthermore, by integrating all these information sources, one can determine how best to allocate scarce testing resources to reduce uncertainty. Both of these goals are facilitated by modern Bayesian computational methods. We apply these tools to examples that were previously solvable only through the use of ingenious approximations, and use genetic algorithms to guide resource allocation.

Key Words: Genetic Algorithm, MCMC, Metropolis-Hastings Algorithm, Reliability Block Diagram, Resource Allocation, Success/Failure Data, System-Level/Subsystem-Level/Component-Level Data.

Introduction

Assessing the reliability of systems represented by reliability block diagram remains important. Take for example, U.S. military weapon systems and nuclear power plants. In making these assessments, often there are information and data available at all levels of these systems, whether they be at the component, subsystem or system level. For example, there may be data from component and subsystem tests as well as expensive full system tests. In this article, we are concerned with assessing the reliability of a system by combining all available information and data at whatever level they are available; here we consider the case where we have success/failure test data.

Much of the reliability literature (Cole (1975), Mastran (1976), Mastran and Singpurwalla (1978), Natvig and Eide (1987), Martz, Waller and Fickas (1988), Martz and Waller (1990)) predates the advances made in Bayesian computation in the 1990's and resorts to various approximations. However, today a fully Bayesian method using the Johnson et al. (2003) framework which combines

all available multi-level data and information can be implemented using Markov chain Monte Carlo (MCMC). In this article, we employ such modern Bayesian methods to make reliability assessments.

In the next section, we introduce the statistical model that combines all available multi-level data and briefly present MCMC for analyzing such data. Then, we illustrate this methodology by making reliability assessments for an air-to-air heat-seeking missile system and a low-pressure coolant injection system in a nuclear power plant first considered by Martz et al. (1988) and Martz and Waller (1990), respectively.

Once multi-level data and information can be analyzed, the question arises of what additional tests should be done when new funding becomes available. That is, what tests will reduce the system reliability uncertainty the most? In this article, we show how a genetic algorithm using a pre-posterior based criterion can address this resource allocation question. We illustrate resource allocation with a simplified system.

A Model for Combining Multi-Level Data

To combine multi-level data for system reliability assessment, we use the Johnson et al. (2003) framework. We introduce the framework's notation and models by considering the reliability block diagram of a simplified system given in Figure 1. First, components, subsystems and the system are referred to as nodes. In this example, the system is node 0 which consists of two subsystems (nodes 1 and 2) in series. The first subsystem consists of two components in parallel (nodes 3 and 4) and the second subsystem consists of three components in series (nodes 5, 6 and 7).

We begin by considering the binomial data model when data are available at a node. At the ith node, there are x_i successes in n_i trials with probability of success (reliability) π_i. If node i is a subsystem or the full system (i.e., not a component), then π_i is expressed in terms of the component reliabilities. For the simplified system, the subsystem reliabilities are expressed as $\pi_1 = 1 - (1 - \pi_3)(1 - \pi_4)$ and $\pi_2 = \pi_5 \pi_6 \pi_7$ and the system reliability is expressed as $\pi_0 = \pi_1 \pi_2 = \{(1 - (1 - \pi_3)(1 - \pi_4))\pi_5 \pi_6 \pi_7\}$. In general, let C be the subset of nodes which are components, and let $\boldsymbol{\pi}_C = \{\pi_i : i \in C\}$; then for $i \notin C$ and for some function h_i, $\pi_i = h_i(\boldsymbol{\pi}_C)$.

Next, we consider prior distributions for node reliabilities. For components, we use beta prior distributions in terms of a best guess for reliability $\tilde{p}_i$ and a precision ν_i which acts like an effective sample size. That is, if the ith node is a component, then $\pi_i \sim Beta(\nu_i \tilde{p}_i, \nu_i(1 - \tilde{p}_i))$. If no information is available, the Jeffreys' prior $Beta(1 \times 0.5, 1 \times (1 - 0.5))$ or a uniform prior $Beta(2 \times 0.5, 2 \times (1 - 0.5))$ can be used.
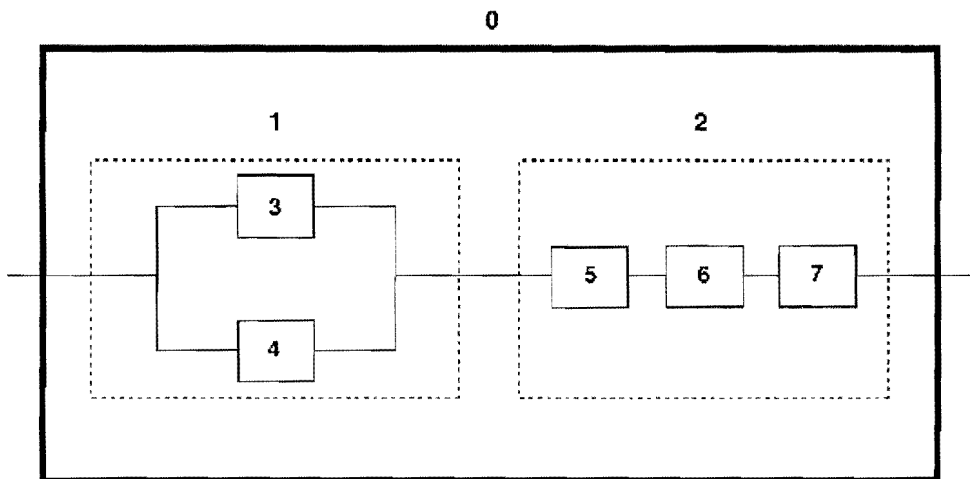
2

Figure 1: Simplified System Reliability Block Diagram.

We also allow the possibility that information ("expert knowledge") is available on the reliabilities of subsystems and/or the full system; we assume that this information is independent of the test data and any information used to build the prior distributions for the component reliabilities. Assume that the information takes the form of an estimated reliability $\tilde{p}_i$ and a precision ν_i. We then express the information contribution, including the x_i successful tests in n_i trials, from the ith subsystem or system as a term proportional to

$$\pi_i^{x_i+\nu_i\tilde{p}_i}\left(1-\pi_i\right)^{n_i-x_i+\nu_i(1-\tilde{p}_i)}.$$

As discussed above, the subsystem or system reliability π_i is expressed in terms of the component reliabilities as $h_i(\boldsymbol{\pi}_C)$. In effect, we have treated this information as if it were derived from binomial data instead of as a beta distribution; the difference involves a change in the exponents of π_i and $(1-\pi_i)$ by one. One effect of this treatment is to ensure that the posterior distribution of $\boldsymbol{\pi}_C$ is well defined. We can define e_i to be the indicator that node i is a component (i.e., $e_i = 1$ if node i is a component, and 0, otherwise), in which case the information contribution from the ith node is

$$\pi_i^{x_i+\nu_i\tilde{p}_i-e_i}\left(1-\pi_i\right)^{n_i-x_i+\nu_i(1-\tilde{p}_i)-e_i},$$

regardless of whether node i is a component. If no information at the ith node is available beyond binomial tests, then $\nu_i = 0$, although $\nu_i > 0$ should be used for components to ensure a proper prior.

3

Table 1: Data for Simplified System

Node	Data	$\tilde{p}$
0	15/20	0.8
1		0.9
2	10/10	0.9
3	34/40	0.9
4	47/50	0.9
5	3/5	0.95
6	8/8	0.95
7	16/17	0.95

In the remainder of this paper, when we refer to the prior distribution, we mean the distribution that arises from combining the component Beta distributions with the upper-level expert knowledge. This is in fact a posterior distribution if there is nonzero expert knowledge, and in this case the components no longer have independent "prior" distributions.

A variety of models might be employed for the ν_i. The ν_i might be treated as constants when they are really thought to be effective sample sizes. On the other hand, they might be described by a distribution, such as $\nu_i \sim Gamma(a_\nu, b_\nu)$. This allows expert knowledge to be downweighted if it is inconsistent with the data. Now consider the data and prior information for the simplified system given in Table 1. Note that no precisions ν_i are provided so that a prior distribution needs to be specified. For illustration, we consider the same precision ν, i.e., $\nu_i = \nu$, and take the prior distribution for ν to be:

$$\nu \sim Gamma(a_\nu = 5, b_\nu = 1).$$

That is, we believe that the expert information on average is worth five Bernoulli observations.

To combine the data with the expert knowledge represented as above, we use Bayes Theorem:

$$f(\boldsymbol{\theta}|\boldsymbol{y}) = f(\boldsymbol{y}|\boldsymbol{\theta})f(\boldsymbol{\theta}) \, / \int f(\boldsymbol{y}|\boldsymbol{\xi})f(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{1}$$

where $\boldsymbol{\theta}$ is the parameter vector (i.e., the component reliabilities π_C and any other unknown parameters), $\boldsymbol{y}$ is the data vector, $f(\boldsymbol{\theta})$ is the prior probability density function and $f(\boldsymbol{y}|\boldsymbol{\theta})$ is the data probability density function (i.e., the binomial probability mass function for binomial data) which viewed as a function of the parameter vector given the data is known as the likelihood. The result of combining the data with expert knowledge is $f(\boldsymbol{\theta}|\boldsymbol{y})$ which is known as the posterior distribution. Since the 1990's, advances in Bayesian computing through Markov chain Monte Carlo or MCMC have made it possible to sample from the posterior distribution (Gelfand and Smith (1990)). Next,

we discuss how the Metropolis-Hastings algorithm (Chib and Greenberg (1995)) can be used to obtain draws or samples from the parameter posterior distribution.

A fully Bayesian analysis of the model described above is nontrivial. The posterior distribution is not analytically tractable: up to a normalizing constant, it is

$$f(\boldsymbol{\pi}_C, \nu | \boldsymbol{x}, \tilde{\boldsymbol{p}}) = f_\nu(\nu) \prod_i \pi_i^{x_i + \nu \tilde{p}_i - e_i} (1 - \pi_i)^{n_i - x_i + \nu(1 - \tilde{p}_i) - e_i}. \tag{2}$$

This looks superficially like a beta distribution, but it is not so simple because of the functional relationships between the π_i; i.e., the subsystem and system $\pi_i = h_i(\boldsymbol{\pi}_C)$. Consequently, a Bayesian analysis requires an implementation of an MCMC algorithm such as Metropolis-Hastings; see, for example, Chib and Greenberg (1995). We use a variable-at-a-time Metropolis-Hastings algorithm as follows. The algorithm loops through all the unknown parameters $\pi_i (i \in C)$ and ν, proposing changes to one parameter at a time and either accepting or rejecting changes according to the Metropolis-Hastings rule. We update the π_i on the logit scale: suppose we are at the stage in one iteration of the algorithm where we are updating π_i (for some $i \in C$). Propose a new value π_i' according to

$$\text{logit } \pi_i' \sim N(\text{logit } \pi_i, s_i^2), \tag{3}$$

where $s_i > 0$ are tunable constants. Accept the value π_i' with probability

$$\min\left\{ 1, \frac{\pi_i'(1 - \pi_i')}{\pi_i(1 - \pi_i)} \frac{f(\boldsymbol{\pi}_C', \nu | \boldsymbol{x}, \tilde{\boldsymbol{p}})}{f(\boldsymbol{\pi}_C, \nu | \boldsymbol{x}, \tilde{\boldsymbol{p}})} \right\}, \tag{4}$$

where $\boldsymbol{\pi}_C'$ is equal to $\boldsymbol{\pi}_C$ except with its ith node reliability replaced by π_i'. If the move is accepted, change the current value of the parameter to be π_i', otherwise its value continues to be π_i. After all the π_i for $i \in C$ have been updated in this way, we update ν on the log scale; this proceeds similarly except that the proposed new values of ν satisfy

$$\log \nu' \sim N(\log \nu, s_\nu^2), \tag{5}$$

so that these proposed new values are accepted with probability

$$\min\left\{ 1, \frac{\nu'}{\nu} \frac{f(\boldsymbol{\pi}_C, \nu' | \boldsymbol{x}, \tilde{\boldsymbol{p}})}{f(\boldsymbol{\pi}_C, \nu | \boldsymbol{x}, \tilde{\boldsymbol{p}})} \right\}. \tag{6}$$

After a complete iteration (after attempts to move each of the π_i for $i \in C$ and also ν), record the current values of all the parameters; this is treated as one sample from the posterior distribution. In practice the first several iterations are discarded as part of a "burn-in" period. Choosing good

values of the s_i is not difficult: in particular, the YADAS software system (Graves (2003, 2006, 2007)) has a method to tune these automatically in the burn-in period.

The same MCMC algorithm just described for making draws from the joint posterior distribution can be used for making draws from the joint prior distribution

$$f(\boldsymbol{\pi}_C, \nu | \tilde{\boldsymbol{p}}) = f_\nu(\nu) \prod_i \pi_i^{\nu \tilde{p}_i - e_i} (1 - \pi_i)^{\nu(1 - \tilde{p}_i) - e_i},$$

where π_i for a subsystem or system is a function of $\boldsymbol{\pi}_C$. Draws for the subsystem and system reliabilities are obtained by evaluating the appropriate functions with the $\boldsymbol{\pi}_C$ draws. The resulting prior distributions for the node reliabilities and ν are displayed as dashed lines in Figures 2 and 3, respectively.

In assessing the system reliability for the simplified system of Figure 1, we combine the node data with the prior distributions using MCMC as just described that result in the posterior distributions displayed as the solid lines in Figures 2 and 3. From these results, the 90% (central) credible interval for the system (node 0) reliability is calculated as (0.697, 0.861) whose length is 0.164. Note that even though there is no data for the first subsystem (node 1), the system data (node 0) and the component data (nodes 3 and 4), dramatically improve what we know about the first subsystem reliability. As shown in Figure 3, the addition of the data does not change ν much, except that ν is somewhat larger than indicated by the prior distribution.

Reliability Assessments for Two Applications

Next, we consider two substantive applications from the literature (Martz et al. (1988) and Martz and Waller (1990)) to demonstrate making reliability assessments with multi-level data.

Series System Example

Martz et al. (1988) considered the reliability of a certain air-to-air heat-seeking missile system consisting of five subsystems in series each consisting of multiple components themselves combined in series as depicted in Figure 4. The data and prior information that Martz et al. (1988) used are presented in Table 2 as (successes/trials) and best guesses $\tilde{p}$ and precisions ν. Martz et al. (1988) did not provide details on how these data were obtained and how the prior information was arrived at.

To compare with Martz et al. (1988), we treat the precisions as constants and then obtain the posterior node reliabilities using YADAS. The posterior node reliabilities are displayed in Figure 5
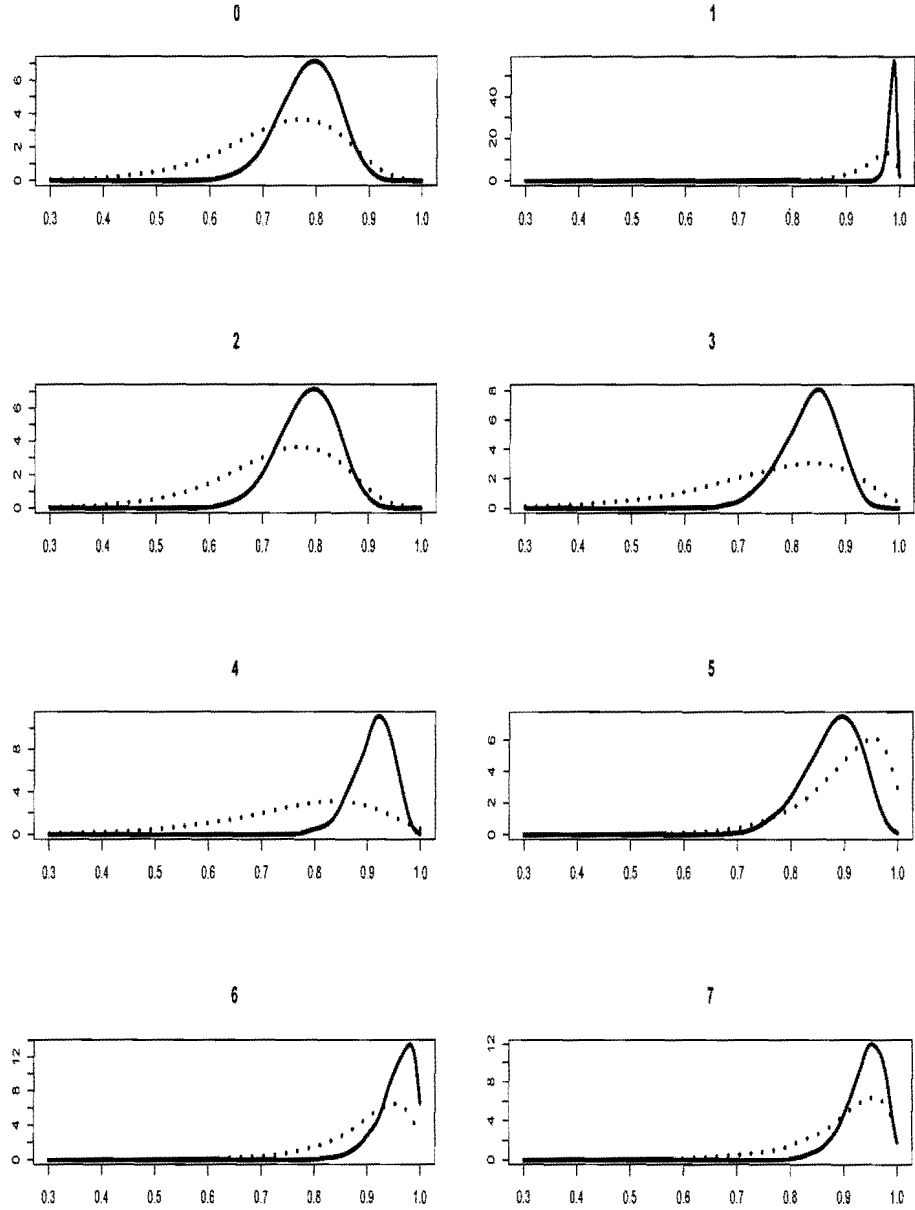
Figure 2: Plot of Simplified System Reliability Priors (dashed lines) and Posteriors (solid lines) for Nodes 0-7.
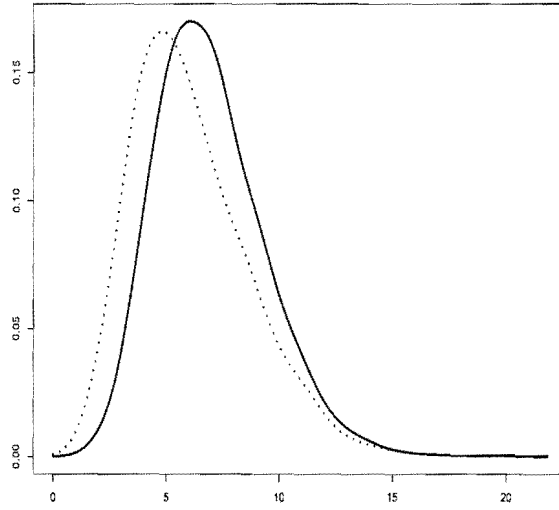
Figure 3: Plot of Simplified System ν Prior (dashed line) and Posterior (solid line).
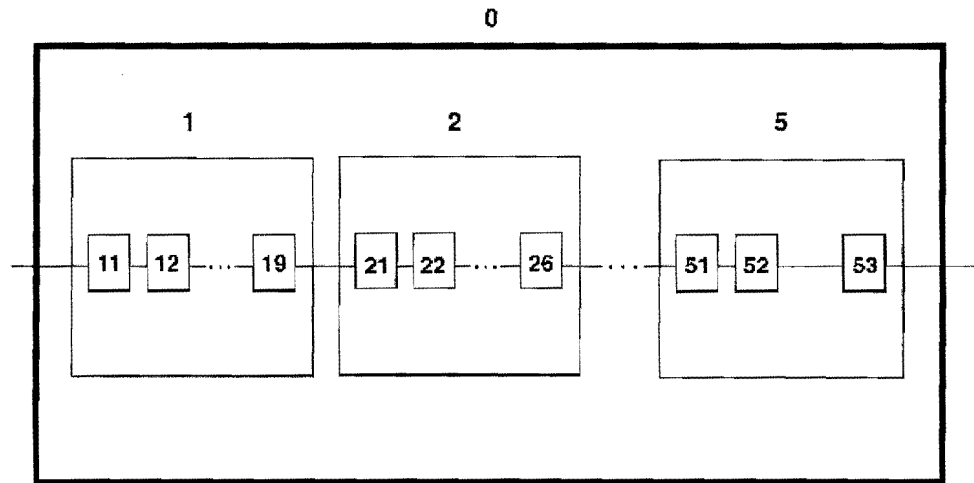

Figure 4: Series System Example Reliability Block Diagram.

Table 2: Data for Series System Example

| Node	Data	$\tilde{p}$	ν
0		115/265	265
1	8/8		
2	7/8		
3	191/205	257/269	269
4		55/66	66
5			
11	30/30	0.5	1
12	80/80	0.5	1
13	39/40	0.5	1
14	30/30	0.5	1
15	90/90	846/848	848
16	10/10	0.5	1
17	29/30	0.5	1
18	20/20	0.5	1
19	5/5	0.5	1
21	50/50	399/402	402
22	50/50	278/302	302
23	99/100	1098/1102	1102
24	23/25	654/690	690
25	50/50	299/301	302
26	55/55	348/352	352
31	129/130	246/250	250
32	130/130	245/250	250
33	129/130	247/250	250
34	129/130	272/276	276
35	130/130	357/360	360
36	247/250	254/257	257
37	129/130	250/252	252
38	249/250	250/252	252
39	330/330	341/352	352
41		797/802	802
42		796/802	802
43		794/802	802
44		791/802	802
45		386/402	402
51		1026/1122	1122
52		1087/1092	1092
53		1084/1092	1092
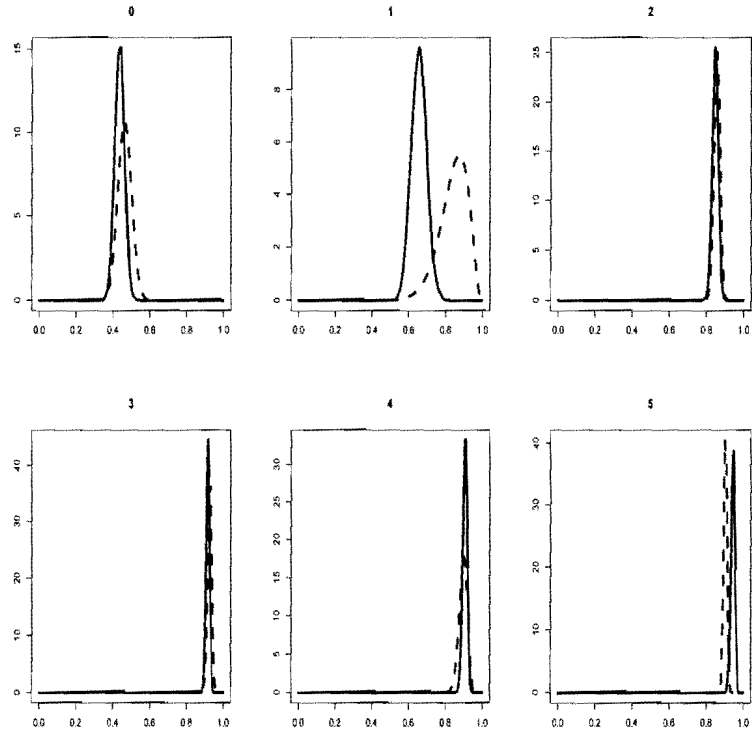
Figure 5: Plot of Series System Example Reliability Posteriors. (Dashed lines are from Martz et al. (1988). Solid lines are from fully Bayesian method.)

Table 3: Comparison of Posteriors for Series System Example (0.05, 0.5, 0.95 quantiles)

Node	Fully Bayesian	Martz et al. (1988)
0	(0.393, 0.436, 0.479)	(0.403, 0.463, 0.525)
1	(0.588, 0.655, 0.723)	(0.701, 0.851, 0.947)
2	(0.820, 0.848, 0.873)	(0.830, 0.858, 0.883)
3	(0.901, 0.917, 0.931)	(0.908, 0.927, 0.944)
4	(0.886, 0.908, 0.925)	(0.858, 0.898, 0.931)
5	(0.926, 0.945, 0.961)	(0.889, 0.904, 0.918)

as solid lines; the Martz et al. (1988) results are displayed as dashed lines. The median (0.50 quantile) and 90% credible intervals (0.05, 0.95 quantiles) for the system and subsystem posterior reliabilities from the fully Bayesian and Martz et al. (1988) methods are given in Table 3.

Note that there is quite a difference in the subsystem 1 results. The difference in location is due to the fact that the approximations used in Martz et al. (1988) do not use higher–level information (system data) to inform estimates of lower–level parameters (such as subsystem 1 reliability). The expert judgment estimate of system reliability, 116/267 or 0.43, is lower than the data and expert judgment at the lower levels would imply, and the fully Bayesian analysis needs to attribute this unreliability to one of the subsystems. Subsystem 1 and in particular component 19 have the sparsest information and are the natural targets. For this reason, the fully Bayesian analysis is more useful than the approach of Martz el al. (1988) in evaluating the usefulness of gathering more data at low levels. In practice one would review the information that led to the low system reliability estimate. The fully Bayesian analysis could be rerun with random ν's, and this would presumably allocate positive probability to the event that $\tilde{p}_0$ is an underestimate.

Complex Series/Parallel System Example

Martz and Waller (1990) considered the reliability of a low-pressure coolant injection system, an important safety system in a nuclear-power boiling-water reactor. It consists of twin trains consisting of pumps, valves, heat exchanges and piping whose reliability block diagram is displayed in Figure 6. The data and prior information that Martz and Waller (1990) used are presented in Table 4 as (successes/trials) and best guess $\tilde{p}$ and precision ν.

Martz and Waller (1990) based component prior distributions (i.e., for nodes 121, 122, 1111, 1112, 1121, 1122, 221, 222, 2111, 2112, 2121, 2122) on data from the Nuclear Regulatory Commission Accident Sequence Evaluation Program database (U.S. Nuclear Regulatory Commission (1987)) and
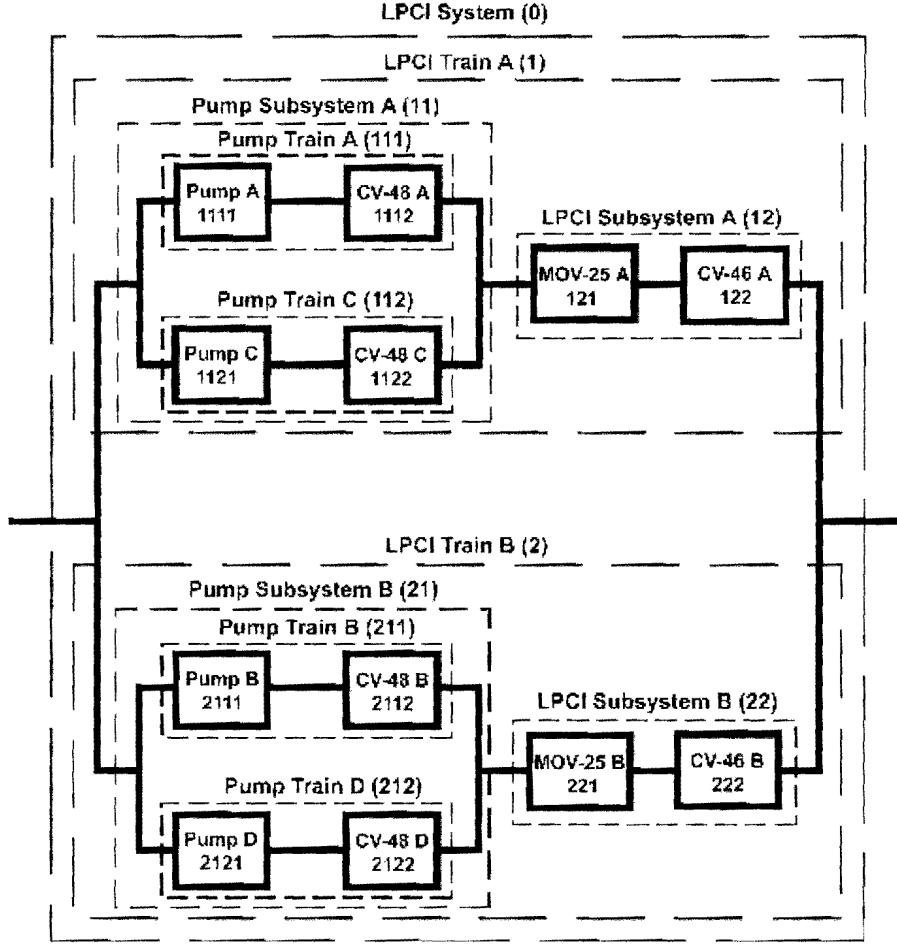
11

Figure 6: Complex Series/Parallel System Example Reliability Block Diagram.

some subsystem prior distributions (i.e., for nodes 12, 111, 112, 222, 211, 212) on composite IEEE Std. 500 reliability data (IEEE (1983)). See Martz and Waller (1990) for more details.

Like Martz and Waller (1990), we treat the precisions as constants and obtain the posterior node reliabilities using YADAS. The resulting posterior reliabilities for the subsystems and system are displayed in Figure 7. Also, the summaries of the posterior reliabilities for all nodes are given in Table 5.

Table 4: Data for Complex Series/Parallel System Example

| Node	Data	$\tilde{p}$	ν
0			
1			
2			
11			
12		242.87/244.66	244.66
111		1.55/1.58	1.58
112		1.55/1.58	1.58
121	240/240	470.13/471.90	471.90
122	240/240	14232.34/14234.12	14234.12
1111	236/240	191.17/191.79	191.79
1112	240/240	14232.34/14234.12	14234.12
1121	238/240	191.17/191.79	191.79
1122	240/240	14232.34/14234.12	14234.12
21			
22		242.87/244.66	244.66
211		1.55/1.58	1.58
212		1.55/1.58	1.58
221	240/240	470.13/471.90	471.90
222	240/240	14232.34/14234.12	14234.12
2111	240/240	191.17/191.79	191.79
2112	240/240	14232.34/14234.12	14234.12
2121	238/240	191.17/191.79	191.79
2122	240/240	14232.34/14234.12	14234.12

13
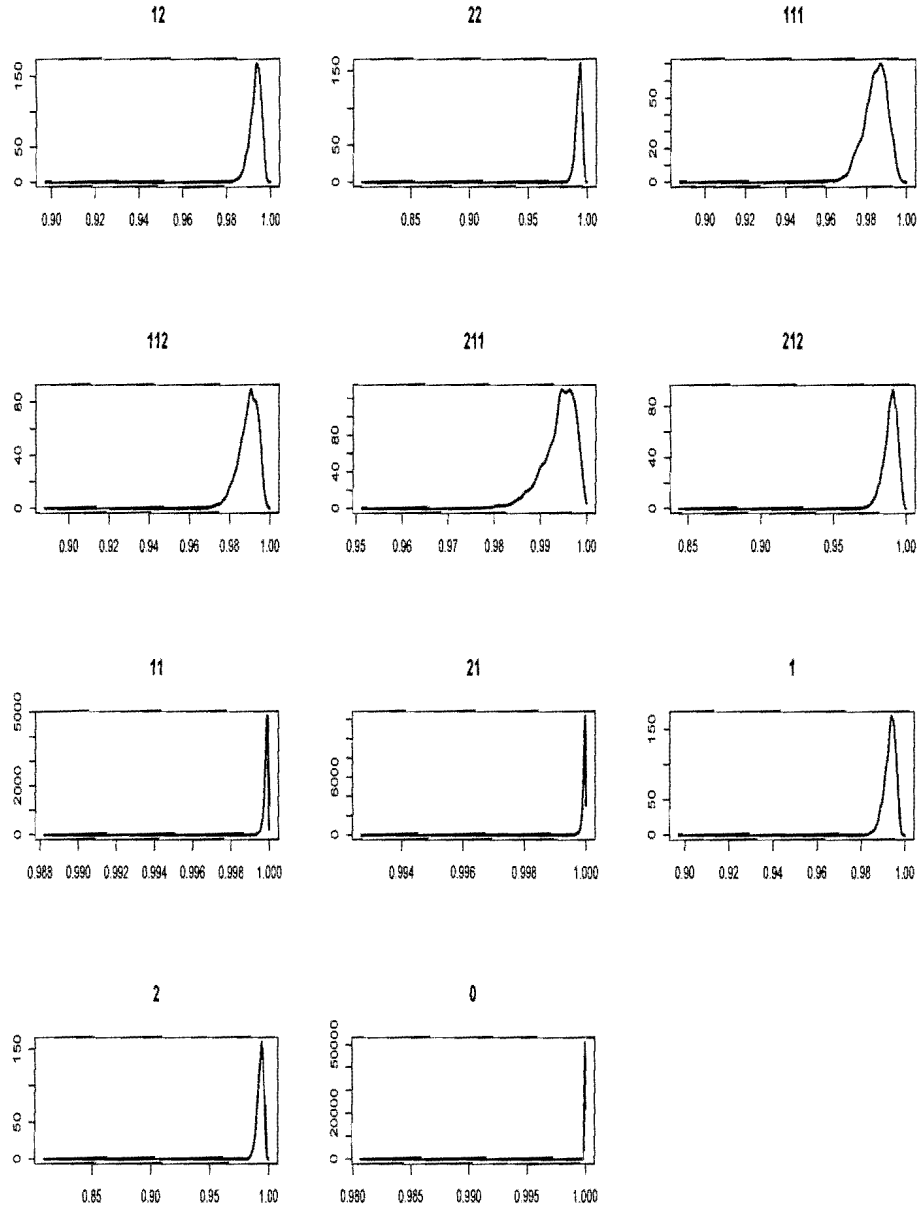
Figure 7: Plot of Complex Series/Parallel System Example Reliability Posteriors for Subsystems and System.

Table 5: Posterior Summaries for Complex Series/Parallel System Example (0.05, 0.5, 0.95 quantiles)

Node	
0	(0.99990, 0.99996, 0.99998)
1	(0.98835, 0.99354, 0.99678)
2	(0.98812, 0.99342, 0.99680)
11	(0.99962, 0.99985, 0.99995)
12	(0.98835, 0.99354, 0.99678)
111	(0.97371, 0.98505, 0.99311)
112	(0.97990, 0.98990, 0.99568)
121	(0.98853, 0.99373, 0.99696)
122	(0.99962, 0.99985, 0.99996)
1111	(0.97389, 0.98519, 0.99329)
1112	(0.99962, 0.99986, 0.99996)
1121	(0.98010, 0.99008, 0.99584)
1122	(0.99961, 0.99985, 0.99996)
21	(0.99983, 0.99995, 0.99999)
22	(0.98812, 0.99342, 0.99680)
211	(0.98681, 0.99468, 0.99847)
212	(0.98040, 0.98994, 0.99599)
221	(0.98826, 0.99359, 0.99697)
222	(0.99962, 0.99985, 0.99996)
2111	(0.98697, 0.99485, 0.99865)
2112	(0.99961, 0.99986, 0.99996)
2121	(0.98054, 0.99011, 0.99614)
2122	(0.99962, 0.99985, 0.99996)

Resource Allocation

Once there is a way to analyze data, then test design can be addressed. When additional funding becomes available, the question of where should the tests be done and how many should be taken arises. In this section, we consider the optimal allocation of additional testing within a fixed budget that results in the least uncertainty of system reliability. We explore this by using the simplified system in Figure 1. We must determine how many tests should be performed at the system, subsystem and component level (i.e., nodes 0-7) under a fixed budget for specified costs at each level (system, subsystem, component). In this article, we use a genetic algorithm (GA) (Goldberg (1989), Michalewicz (1992)) to do the optimization. But other optimization methods like particle swarms (Eberhart and Kennedy (1995)) could easily be used instead.

Thus, we assume that there is a cost for collecting additional data with higher-level data being more costly than lower-level data. Consider the following costs as an example of the costs for testing at each node. Recall that node 0 is the system, nodes 1 and 2 are subsystems and nodes 3–7 are components:

$$(0: \$5), (1: \$2), (2: \$3), (3: \$1), (4: \$1), (5: \$1), (6: \$1), (7: \$1).$$

We evaluate a candidate allocation (i.e., a specified number of tests for each of the eight nodes) using a pre-posterior based criterion as follows. We take a draw from the current joint posterior distribution (based on the current data) of the node reliabilities and draw binomial data according to the candidate allocation. Then we combine these new data with the current data using the same prior distributions to obtain an updated posterior distribution of the node reliabilities; again we use MCMC to obtain N_p draws from this updated posterior distribution. The length of the 90% central credible interval of the system reliability posterior distribution is taken as a measure of uncertainty. This is repeated N_d times, each with a different draw from the current joint posterior distribution of the node reliabilities. The uncertainty criterion is then calculated as the 0.90 quantile of the resulting 90% credible interval lengths.

Briefly, we describe how a GA can be used to find a nearly optimal allocation. A GA operates on a "population" of candidate allocations, where a candidate allocation is a vector of node sample sizes. The GA begins by constructing an initial population or generation of M allocations by randomly generating allocations that do not exceed the given fixed budget. The uncertainty criterion for each of these allocations in the initial population is evaluated and the allocations are ranked from smallest

to largest, i.e., the best allocation has the smallest criterion in the initial population. The second (and subsequent) GA generations are then populated using two genetic operations: crossover and mutation (Goldberg (1989), Michalewicz (1992)). A crossover is achieved by randomly selecting two parent allocations from the initial (or current) generation without replacement with probabilities inversely proportional to their rank among the M allocations in the initial (or current) generation. A new allocation is generated node by node from these two selected parent allocations by randomly picking one of the two parents each time and taking its node sample size. The two parent allocations are then returned to the initial (or current) population before the next crossover is performed. In this way, an additional M allocations are generated using the crossover operator. The generated allocations are checked to make sure they do not exceed the budget, so that new allocations are generated until there are M such allocations. The uncertainty criterion is then evaluated for each of these new allocations. A mutation of each of the initial (or current) M allocations is obtained node by node by first randomly deciding to change the node sample size and if so then randomly perturbing the current node sample size. Using mutation, M additional allocations which remain within the budget are generated and the uncertainty criterion for each is evaluated. At this point there are $3M$ allocations. In the next generation, the current population consists of the M best allocations from these $3M$ allocations, i.e., with the smallest uncertainty criterion. The GA is executed for G generations. We implemented the GA for resource allocation in R (R Development Core Team (2004)) which generates the candidate allocations. An allocation is evaluated in R by repeatedly building YADAS input data files, running the YADAS code using the reliability package (through the R "system" call) to analyze the new and current data, and reading the resulting YADAS output files back into R to calculate the uncertainty criterion.

In the implementation, there are a number of issues regarding the choice of M, G, N_p and N_d. As the population size M and number of generations G increases, more candidate allocations (i.e., $M(1 + 2 \times G)$) are entertained, but then more calculation is required. As the number of posterior draws for each generated data set N_p and the number of generated data sets to analyze N_d increases, the uncertainty criterion is better evaluated, but the calculation needed to evaluate a single candidate allocation can dramatically increase let alone that for $M(1 + 2 \times G)$ candidate allocations. One has to realize that the nearly optimal allocation found by the GA may not be the optimal allocation if the difference between them is less that the variability of the evaluated uncertainty criterion, i.e., within the simulation error of the uncertainty criterion.

One might ask if there are any general insights regarding resource allocation with assessment of

system reliability in mind. If we consider testing at the same level, for components (or subsystem), the component (or subsystem) with the most uncertainty will require more testing than the others. If the subsystems are connected in series, but some subsystems have components connected in series where as other subsystems have components connected in parallel, in terms of component testing, the parallel configured subsystems will require less testing; this can be explained by examining the subsystem reliability expression, which shows that the reliability of series configured subsystems is of second order in their component reliabilities, where as that for parallel configured subsystems is of first order. The allocation will also depend on the testing costs relative to the amount of information they provide. If we consider a series configured subsystem, if the subsystem cost exceeds the sum of the components costs, then performing components tests will be recommended; if the subsystem cost is less than the sum of the components costs, then performing some subsystem tests may be recommended if they provide relatively more information. But for complicated systems with many subsystems and components whose costs are all different, it will be difficult to choose an optimal allocation with these rules of thumb. However, the proposed methodology balances all these costs and information across the entire system in finding a nearly optimal allocation.

Next, we illustrate the GA for the resource allocation problem described above for the simplified system depicted in Figure 1 for a fixed budget of $1000. The length of the 90% credible interval of system reliability based on the existing data is 0.164. We use populations of size $M = 20$ and $G = 50$ generations, so that 2020 ($= 20(1 + 2 \times 50)$) candidate allocations were generated and evaluated. To evaluate the uncertainty criterion, we generated $N_p = 2000$ posterior draws per data analysis and generated $N_d = 500$ data sets corresponding to posterior draws based on the existing data. For this situation, what allocation yields the most reduction in the uncertainty criterion for system reliability?

Based on the proposed methodology described above, the GA produced the traces presented in Figures 8 and 9 which display the best uncertainty criterion and allocation found during each generation. The uncertainty criterion drops to 0.0804 for the initial population and decreases to 0.0725 by generation 50 with an allocation of (0, 0, 175, 0, 0, 208, 137, 128) for nodes 0–7. We evaluated this allocation with $N_p = 50000$ and $N_d = 100000$ and obtained uncertainty criterion values of 0.073358 and 0.073363, so we take the uncertainty criterion for this allocation as 0.0734. These results suggests that there is enough data for node 1, the two component parallel subsystem and the cost structure prohibits additional system tests (i.e., the system cost equals the sum of the subsystem costs, which equals the sum of the components costs.). Because the node 2 subsystem
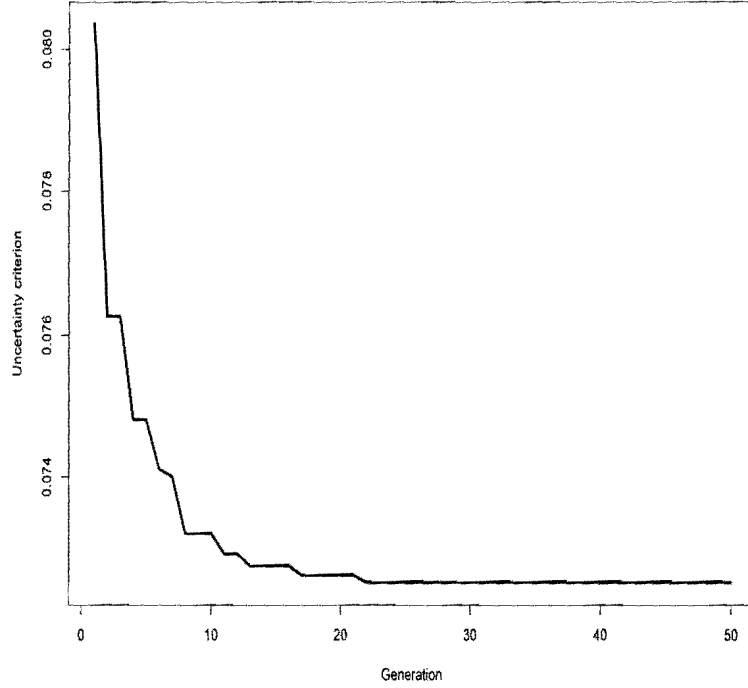
18

Figure 8: GA Evolution of Uncertainty Criterion.

cost equals the sum of its component costs, we tried an allocation which proportionally allocated the subsystem tests to its components (i.e., splitting up $175 \times 3 = 525$ by the proportion (208/473, 137/473, 128/473) found by the GA) giving the allocation (0, 0, 0, 0, 0, 439, 289, 270). Evaluating this allocation again with $N_p = 50000$ and $N_d = 100000$ gave uncertainty criterion values of 0.071439 and 0.071426, which we round to 0.0714. Consequently, there is some improvement by doing all component tests for the node 2 subsystem.

Discussion

We have illustrated how to respond to the challenge of integrating all information available at the various levels of a system in order to estimate its reliability. Bayesian models have always been natural for doing this integration, and the computational tools have now caught up to make this practical. Moreover, because we are able to analyze such data, we can now consider the problem of allocating additional resources that best reduce the uncertainty in the system reliability assessment.

We have discussed the case of binomial test data only for systems represented by reliability block diagrams. Hamada et al. (2004) showed how binomial data can be analyzed for problems
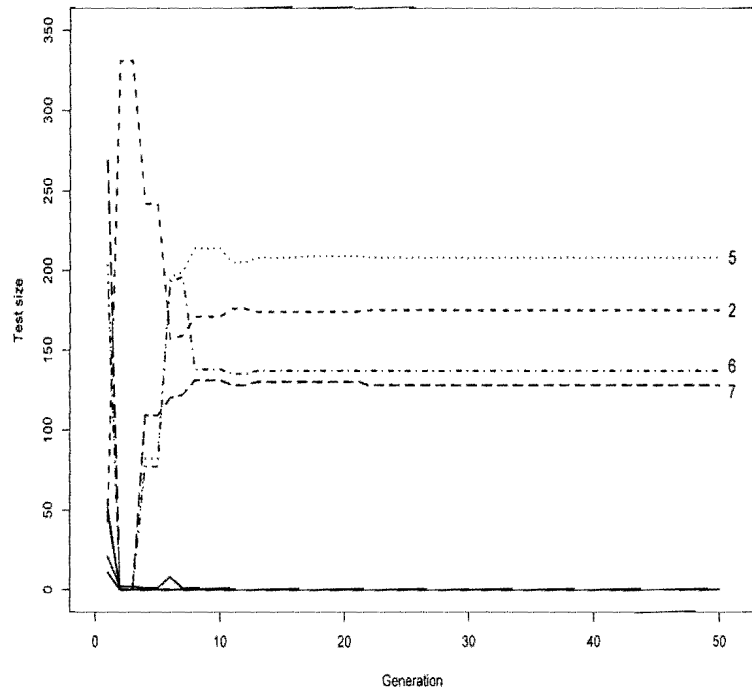
Figure 9: GA Evolution of Resource Allocation. Nodes 2 and 5–7 test sizes are identified.

using fault tree representations. Component and subsystem tests may generate continuous data such as lifetimes, and their distributions may depend on covariates such as different suppliers. Graves and Hamada (2004) presented an example of such an analysis. However, the problem of resource allocation for non-binomial test data is a topic for future research.

Acknowledgments

We thank C. C. Essix for her encouragement of this work. We also thank Vivian Romero for her assistance in producing the reliability block diagram figures used in this paper.

References

Chib, S. and Greenberg, E. (1995). "Understanding the Metropolis-Hastings Algorithm". *The American Statistician*, 49, pp. 327–335.

Cole, P.V.Z. (1975). "A Bayesian Reliability Assessment of Complex Systems for Binomial Sampling". *IEEE Transactions on Reliability*, R-24, pp. 114–117.

Eberhart, R.C. and Kennedy, J. (1995). "A New Optimizer Using Particle Swarm Theory". *Proceedings Sixth Symposium on Micro Machine and Human Science*, IEEE Service Center, Piscataway, NJ, pp. 39–43.

Gelfand, A.E. and Smith, A.F.M. (1990). "Sampling-Based Approaches to Calculating Marginal Densities". *Journal of the American Statistical Association*, 85, pp. 398–409.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, NY.

Graves, T.L. (2003). "An Introduction to YADAS", yadas.lanl.gov.

Graves, T.L. (2006). "The YADAS Reliability Package", Los Alamos National Laboratory Technical Report LA-UR-06-7739. Available for download at the YADAS website at http://www.stat.lanl.gov/yadas/node1.html#download.

Graves, T.L. (2007). "Design Ideas for Markov Chain Monte Carlo Software". *Journal of Computational and Graphical Statistics*, 16, pp. 24–43.

Graves, T.L. and Hamada, M.S. (2004). "Bayesian Methods for Assessing System Reliability: Models and Computation". In *Modern Statistical and Mathematical Methods in Reliability*, Series on Quality, Reliability, and Engineering Statistics 10, A. Wilson, N. Limnios, S. Keller-McNulty, and Y. Armijo (Eds.), World Scientific Publishing Co., Singapore.

Hamada, M., Martz, H.F., Reese, C.S., Graves, T., Johnson, V. and Wilson, A.G. (2004). "A Fully Bayesian Approach for Combining Multilevel Failure Information in Fault Tree Quantification and Optimal Follow-On Resource Allocation". *Reliability Engineering and System Safety*, 86, pp. 297-305.

Institute of Electrical and Electronic Engineers (IEEE) (1983). *IEEE Guide to the Collection and Presentation of Electrical, Electronic, Sensing Component, and Mechanical Equipment Reliability Data for Nuclear-Power Generating Stations*. Distributed by Wiley-Interscience, New York, NY.

Johnson, V., Graves, T., Hamada, M. and Reese, C.S. (2003). "A Hierarchical Model for Estimating the Reliability of Complex Systems (with discussion)". *Bayesian Statistics 7*, Bernardo, J.M., Bayarri, M.J., Berger, J., Dawid, A.P., Heckerman, D., Smith, A.F.M. and West, M. (Eds.). Oxford University Press, London, pp. 199–213.

Martz, H.F., Waller, R.A. (1990). "Bayesian Reliability Anaysis of Complex Series/Parallel Systems of Binomial Subsystems and Components". *Technometrics*, 32, pp. 407-416.

Martz, H.F., Waller, R.A. and Fickas, E.T. (1988). "Bayesian Reliability Analysis of Series Systems of Binomial Subsystems and Componenents". *Technometrics*, 30, pp. 143-154.

Mastran, D.V. (1976). "Incorporating Component and System Test Data into the Same Assessment: a Bayesian Approach". *Operations Research*, 24, pp. 491-499.

Mastran, D.V. and Singpurwalla, N.D. (1978). "A Bayesian Estimation of the Reliability of Coherent Structures". *Operations Research*, 26, pp. 663-672.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, New York, NY.

Natvig, B. and Eide, H. (1987). "Bayesian Estimation of System Reliability". *Scandinavian Journal of Statistics*, 14, pp. 319-327.

R Development Core Team (2004). "R: A Language and Environment for Statistical Computing". R Foundation for Statistical Computing, Vienna. (http://www.R-project.org).

Spiegelhalter, D., Thomas, A., Best, N. and Lunn, D. (2004). *WinBUGS Version 1.4 User Manual.*

U.S. Nuclear Regulatory Commission (1987). "Reactor Risk Reference Document (Vols. 1–3, draft)". Report NUREG-1150.