

MathSoft

Σ + $\sqrt{\quad}$ - = \times \int \div δ
StatSci Division

Statistical Science Research Report 6

An Extended Example for Testing GRAPHICAL-BELIEF

Russell G. Almond

Statistical Sciences, Inc.
1700 Westlake Ave, N., Suite 500
Seattle, WA 98109
almond@statsci.com

9/2/92

This research is supported by the GRAPHICAL-BELIEF project, NASA SBIR contract NAS 9-18669.
Copyright 1992, Statistical Sciences, Inc. All rights reserved. This copyright covers all text and illustrations.

An Extended Example for Testing GRAPHICAL-BELIEF

Russell Almond, StatSci

ABSTRACT

This example, taken from Martz and Waller [1990] is meant to illustrate several features of the Phase I GRAPHICAL-BELIEF program. This paper presents several variations on the original problem of Martz and Waller. The original problem was to calculate the availability of the low pressure coolant injection system for a nuclear reactor. Problem 1 makes several simplifying assumptions in the original problem; in particular, the Phase I version ignores the uncertainty about the failure rates. This simple example illustrates several features of the system, using it to perform importance analysis and study hypothetical situations and perform diagnosis. Problem 2 adds additional data from Martz and Waller about subsystem reliability. Problem 3 adds a common cause failure to the problem originally posed in Martz and Waller. Problem 4 talks about the difficulties inherent in going from a failure on demand (Bernoulli Process) model to a failure in time (Poisson Process) model.

Problem 1: Simple Reliability Calculation

The most basic sorts of systems encountered in reliability modelling consist of components in a nesting sequence of series and parallel subsystems. Martz and Waller[1990] present such a system as an illustration of their technique. A block diagram for this system is presented in Figure 1. The following description of the problem is taken from Martz and Waller[1990]:

The Bayesian procedure presented [in Martz and Waller] was motivated by the following problem concerning the engineered safety features of a certain 1,150 megawatts electric U.S. commercial nuclear-power boiling-water reactor. In such a reactor, one important safety system is the low-pressure coolant injection (LPCI) system that provides coolant to the reactor vessel during accidents in which vessel pressure is low. It consists of two trains containing pumps, valves, heat exchangers and piping. It normally operates in a standby mode, awaiting a demand for its use. Consequently, certain components must perform a change of state on demand; for example, the motor-driven pumps must start, the motor-operated valves must operate, and the check valves must open. Once started, the system must operate for a designated length of time, and, consequently, various time-related failure modes are also of interest. We restrict our attention here, however, to failure to start on demand (or simply failure on demand). In this case the binomial distribution is the appropriate model for the test data and the probability of the system failure on demand is known as its demand unavailability. The system may be unavailable on demand for two reasons, failure while on standby and unscheduled maintenance. For simplicity only, however, that portion of demand unavailability caused by standby failure is considered here.

Figure 1 [presented here as Figure 1 also] shows the system-demand-availability block diagram corresponding to an accident in which a single pump train would be sufficient to mitigate the accident. The system-demand-availability block diagram thus consists of two trains in parallel, each consisting of two parallel pump trains, both of which operate in series with a motor-operated valve [MOV] and a check valve [CV]. The problem is to estimate the system demand unavailability due to failure while on standby based on tests and prior data on each component, as well as additional prior data on pump trains A-D and LPCI subsystems A-B. Figure 1 also shows the decomposition of the system into a set of 11 series or parallel subsystems required for the model presented in . . . [their paper.]

The situation considered here is one in which a system of conditionally independent components may be decomposed into a set of m series or parallel subsystems of other subsystems or components. Some of the components or subsystems may appear more than once in the system. These multiple appearances do not mean that the same device performs more than one function in the system. Rather, the replicated devices represent those that are either known or believed to have identical reliabilities (or availabilities). Such repeated devices often will have binomial test results only about the common generic device and not about each individual physical device in the system. The method accommodates such replicated devices.

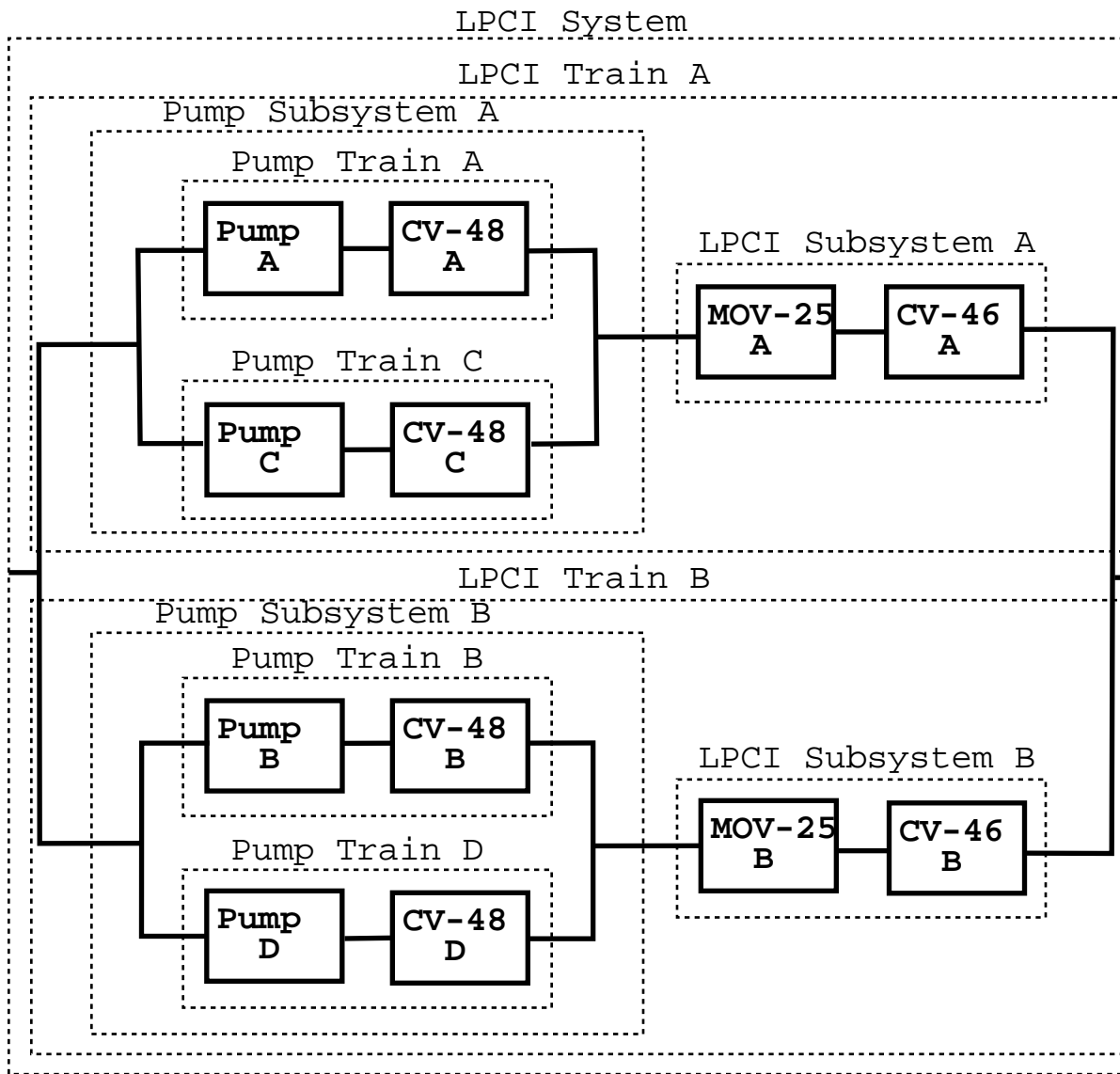


Figure 1. Block Diagram of Martz and Waller [1990] Problem

That ability to smoothly handle such replicated devices is indeed crucial to the ability to model such systems. This is especially true as systems such as the LPCI System described in Martz and Waller are usually designed with deliberate redundancy in order to improve their reliability. Such replicated design features may occur for other reasons as well; in particular, a system designer may use two similar subsystems to perform similar functions in quite unrelated systems. This modular design is a time-saver for the system designer, who does not need to redesign the subsystem. We should be able to use this in modelling its reliability as well.

To turn the model presented in Figure 1 into a graphical model suitable for GRAPHICAL-BELIEF, we must first identify the attributes of the problem. There is one attribute for each subsystem or component failure mode. As the modelled system is simple, there is only one failure mode for each component and subsystem; thus a single binary attribute can adequately describe the state of each component or subsystem. More complicated examples could require either multiple attributes describing the state of a

single component or system, or non-binary attributes corresponding to multiple functional states. The 12 basic components plus the 11 subsystems make a total of 23 attributes in this example.

The graphical model is constructed by connecting components to subsystems. For example, Pump Train A fails if and only if either Pump A or CV-48 A (Check-valve) fail. This defines a relationship between the two components and the subsystem. Because it is most natural to think of this relationship as the state of the Pump Train subsystem is dependent on its two components, we draw this relationship as a pair of arrows from Pump A and CV-48 A to Pump Train A. Similarly, Pump Subsystem A fails if and only if both Pump Train A and Pump Train C fail. This is again drawn with a pair of arrows from the Pump Train subsystems to Pump Subsystem A, indicating the most natural direction of conditioning. Figure 2 shows the graphical structure of the model.

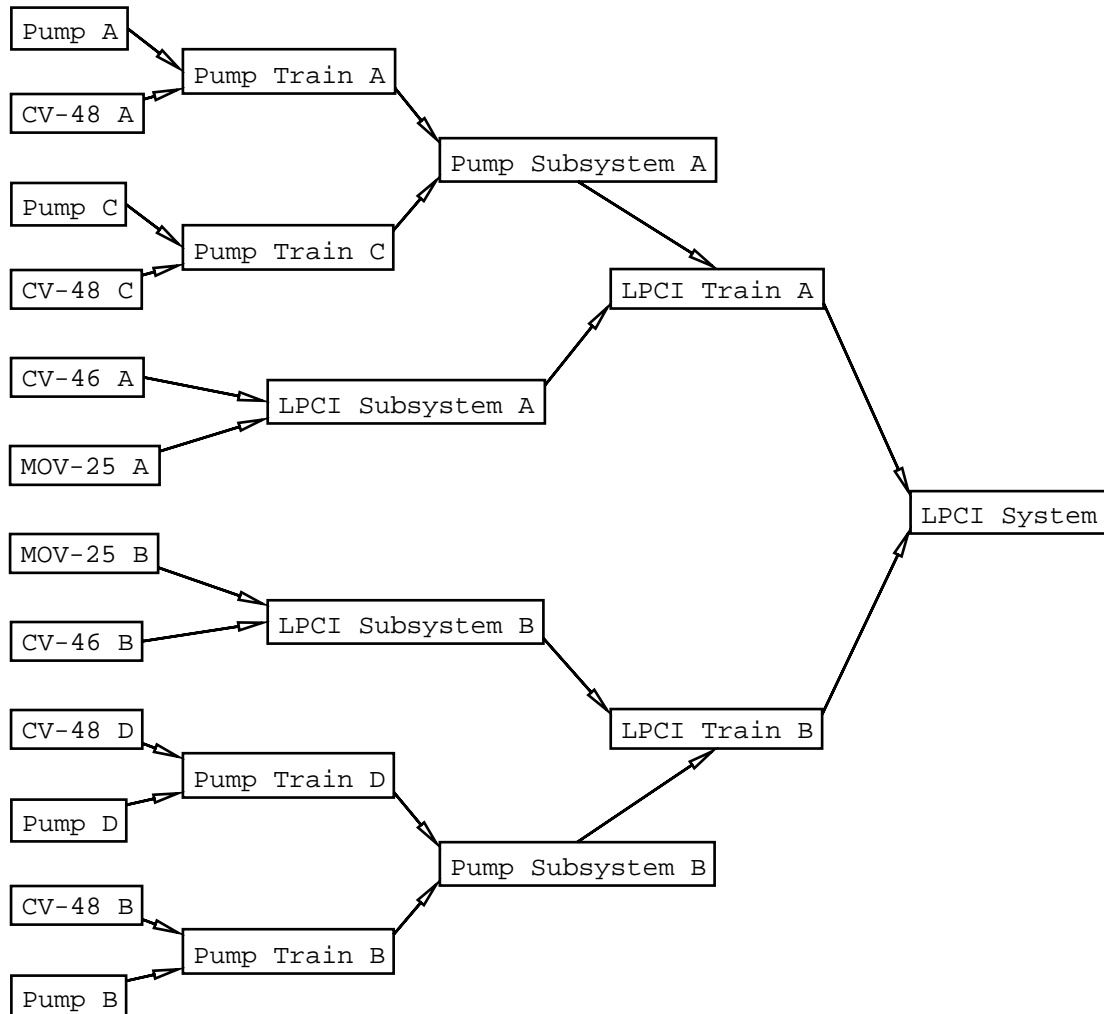


Figure 2. Graphical Model of Martz and Waller[1990] Problem

The graphical structure of this model is very similar to other models which have been used for reliability analysis, such as fault trees or the graphical models used by FEAT (Failure Events Analysis Tool, NASA[1992]). More traditional reliability models make a visual distinction between the “And” and “Or” relationships; limiting the user to the choice of a small collection of relationships. GRAPHICAL-BELIEF makes no such distinction, and the relationship in GRAPHICAL-BELIEF are not restricted to the “And” and “Or” varieties; they can be anything which is expressible using the language of conditional probabilities or belief functions. This includes “And” and “Or” relationships, as well as others, such as “ k out of

n.” Furthermore, nodes in graphical belief models need not be binary (True/False); they can represent a richer collection of states (Working/Stuck Open/Stuck Closed/Rupture). As the modeller has considerable freedom in designing the relationships, the removal of the binary restriction presents no problems. Belief functions and probabilities can also model noisy relationships; this is described in Problem 2.

The relationships among the components as well as the rates of basic events are modelled with *valuations*—a general class of relationship model which includes probability distributions and belief function. Figure 3 shows a session with a valuation (in this case, probability) editor which might be used to elicit the relationship between Pump A and CV-48 A and Pump Train A. The heart of this valuation is the two by two table in the center. This indicates that if either the pump or the check-valve fail, then the Pump Train fails with probability 1.0 (near certainty). Notice that the three possible configurations of Pump A and CV-48 A failures corresponding to either component failing have been grouped together for the purpose of elicitation. This group, labeled “(:OR :XX :YY),” simplifies the table from 8 to 4 entries. The other group “(:NOR :XX :YY)” corresponds to the remaining case where neither component has failed.

These groupings also simplify the task of modelling imperfect relationships as shown in Problem 2. Note that the distributions describing the relationships between Pump *i* and CV-48 *i* and Pump Train *i* (for *i* = B, C, D) are identical except for the attributes over which they are defined. In order to allow the engineer to easily model these relationships, we allow him to store the modelled relationship between Pump A and CV-48 A and Pump Train A in a library and retrieve it for later re-use. This same relationship can also be used in describing the relationships between Pump Subsystem *i*, LPCI Subsystem *i* and LPCI Train *i* (for *i* = A, B).

Save Consequences Conditions Values

Conditions: (PUMP-A CV-48-A)

Vector: #((:FAIL :NOFAIL) (:FAIL :NOFAIL))

Consequences: (PUMP-TR-A)

Vector: #((:FAIL :NOFAIL))

	<i>Working</i>	<i>Failed</i>
(:OR :XX :YY)	0.0000	1.0000
(:NOR :XX :YY)	1.0000	0.0000

Normalize Read Only?

Probability Belief

Figure 3. Valuation Editor for Relationship between Pump A, CV-48 A and Pump Train A

The relationship between Pump Trains A and C and Pump Subsystem A is an “And” relationship instead of an “Or” but the table at the heart of its valuation editor looks similar (Figure 4). Here the groups have changed to “(:AND :XX :YY)” corresponding to the case where both subsystems have failed and “(:NAND :XX :YY)” corresponding to the three cases where at least one subsystem is working.

Note that most of the graphical structure in Figure 2 can be inferred from the 11 relationships. That is because Pump A and CV-48 A are on the conditional side of their relationship, and Pump Train A is on

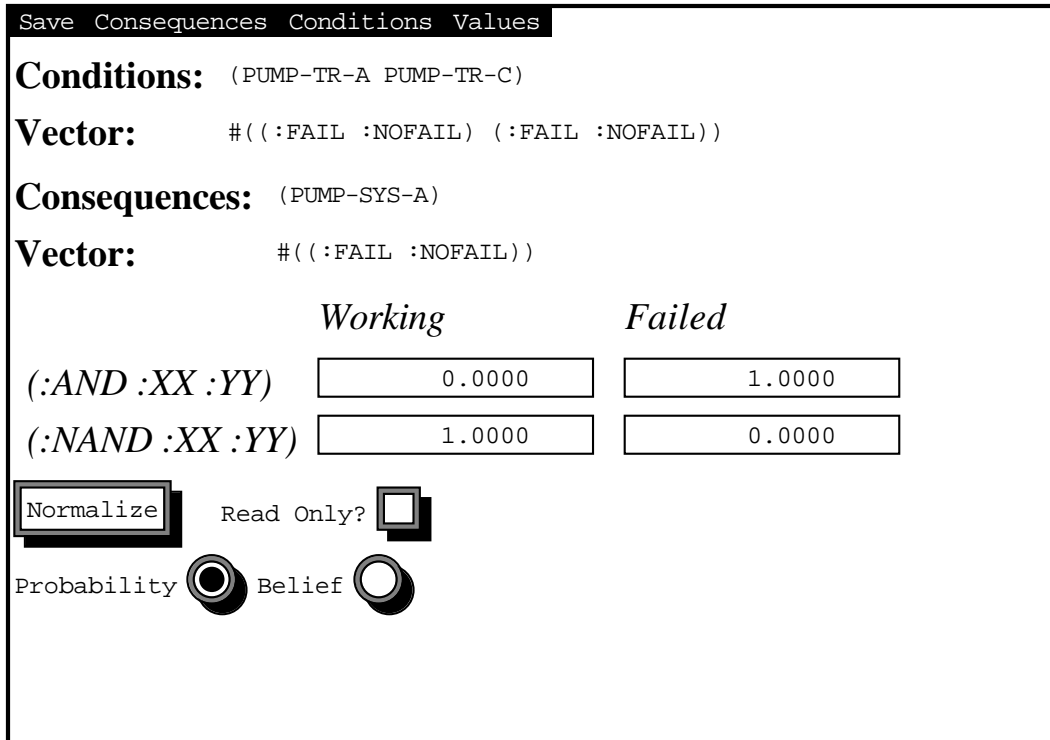


Figure 4. Valuation Editor for Relationship between Pump Trains A and C and Pump Subsystem A

the consequential side, the relationship defined in Figure 3 is displayed as two arrows from Pump A and CV-48 A to Pump Train A. Applying this same principle to the other 10 relationship, we can independently arrive at the graph defined in Figure 2. This is a result of the fundamental principle of graphical modelling: model graphs correspond to factorizations of the joint probability (or belief) distribution of all of the attributes. In fact, the older BELIEF package built the model graph from rule bases which were nearly descriptions of relationships. GRAPHICAL-BELIEF is indented to allow more flexibility about the order of specification of the graphical and numerical components of the relationships while providing tools to support the fundamental links between components of the joint probability function and the graphical model.

We now turn to the 12 basic events in the system, the state of the 12 components. Actually, there is again a fair amount of structure here, because there are only really three types of components in the system, the four pumps, the two motor operated valves and the six check valves. In fact, if we assume that any component is replaced if it fails during use or testing, the failure model for all components of the same type will be identical. One of the simplifications that we will make in the Martz and Waller model is to assume this simple form of replacement maintenance.

Note that Martz and Waller make a different simplifying assumption, namely that there is no informational dependence between the various components, *i.e.*, test data about Pump A tells us nothing about Pump B. This is based on the fact that the pumps are very large and are repaired in place rather than replaced. The best model lies somewhere in between: there is an overall failure rate for pumps of this type and a factor for each specific pump by which it is more or less reliable. Such a hierarchical model is slightly more difficult computationally; although support for such models is a planned extension to the GRAPHICAL-BELIEF prototype (as part of the parameter capability described below).

Data for these systems (pooling the prior information in Table 1 and Table 2 of Martz and Waller) is given in the following table:

<i>Component Type</i>	<i>Failure Mode</i>	<i>Failures</i>	<i>Tests</i>
Motor-driven Pump	Failure to Start	5.62	1149.79
Motor-operated Valve	Failure to Operate	.77	949.9
Check Valve	Failure to Open	.78	15672.12

Table 1. Failure Rate Data for Basic Events

This table contains strange looking fractional test cases. These come from fitting both test data from this system and data from an available data base (Nuclear Regulatory Commission Accident Sequence Evaluation Program data base) combined with specific test data from this system. The failure rates are not known with certainty, therefore probability distributions are fit to the available information. The particular distribution used by Martz and Waller, the Beta distribution, has the advantage that its parameters can be interpreted as number of failures and number of relevant cases; these number were derived by getting first a best estimate of the average failure rate from all available information sources, and then an estimate of the variability expressed in terms of number of relevant cases. Because not all data are directly relevant and the NUREG study fit a slightly different model to the data, the resulting fractional cases appear in the data. These fractional cases present no computational difficulty in the model, and in fact allow engineers to express engineer judgement in terms which are easily integrated with the test data: average failure rates and equivalent number of test cases.

Of more difficulty in the uncertainty in the failure rates. To correctly describe our information about the system failure rate, we must correctly account for this uncertainty in the failure rate. The mechanism to do that in GRAPHICAL-BELIEF is through the use of *parameters*. For example, the model for Pump A might look like Figure 5. Here, the two parameters @Pump and @1-Pump are derived from the information about the pump failure rates: the beta distribution implied by the data from Table 1.

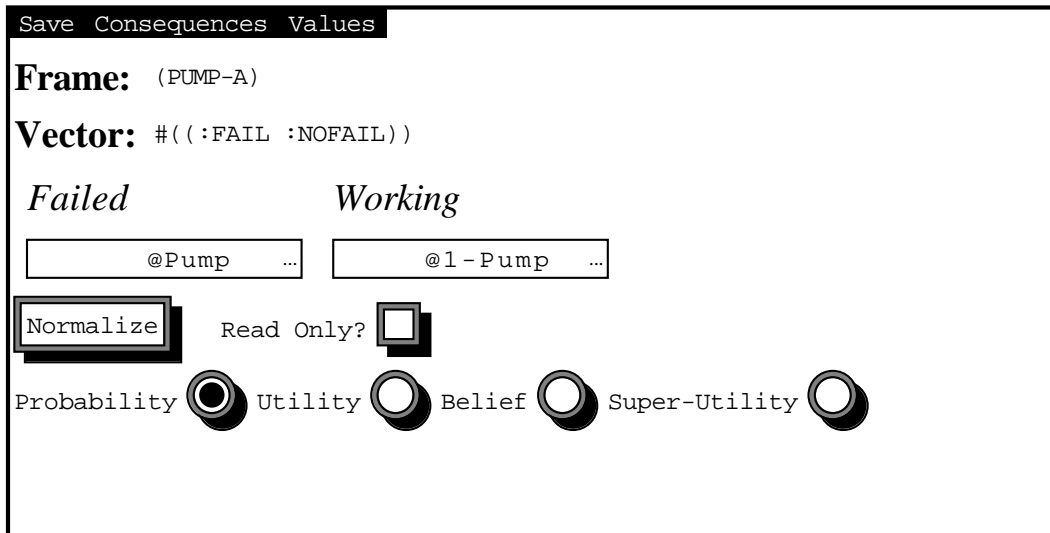


Figure 5. Valuation Editor for Basic Event, With Parameters

There are several different ways in which parameter values can be used. For a large number of uses (for example rough and ready calculations for identifying the most frequent causes of problems and studying reliability trade-off in various proposed design modifications or part substitutions) it is often sufficient to just use the best estimate for the parameters and not worry about the uncertainty in the parameter estimates. For other situations, we may want to use a pessimistic estimate (such as a 95% upper bound) or an optimistic one (such as a 95% lower bound). For the final evaluation of a system, we may decide to use a more time intensive Monte Carlo procedure, sampling from the distribution of the parameters (Almond[1990]). The use of parameters allows us to build a second order model for the system reliability; directly modelling uncertainty about the parameters of the first level graphical model.

The use of parameters for describe basic event rates has another advantage, that of division of labor in the modelling process. In particular, one engineer, a specialist in component reliability and testing, would produce the model for the component failure rates, basing her judgements on test data for the components. A second engineer, specializing in system design, would assemble the component system model, drawing the component models from a library of component models assembled by the other engineer.

As parameters have not yet been implemented in GRAPHICAL-BELIEF, we will ignore the complication produced by uncertainty in the parameter estimates for Phase I. Using mechanisms already established in BELIEF and ElToY (Almond[1992]), we expect that these capabilities can be rapidly incorporated into the Phase II version of GRAPHICAL-BELIEF.

Note that in there is a fair degree of symmetry in Figures 1 and 2. This is not co-incidental, but is rather a deliberate design feature. In this case redundancy is used to improve the reliability of the system. In other cases, it may be used for design efficiency (for example, port and starboard engines). It should be possible to recognize and use such redundancies in the modelling process.

One way this can be done is to allow the engineers to store fragments of models in libraries for later re-use. This is especially useful in systems where there is not such a close correspondence between the failure states and the original components. For example, a particular valve and actuator combination might have a complex interaction which needs to be modelled with a small graph containing say six nodes. This combination may appear a number of different places in the design, playing different roles in each place. By placing the graph fragment (and associated valuations) describing that interaction in a library, it can be re-used as needed.

Another order of re-use becomes available for large graph fragments. If two subsystems are truly exchangeable, then it should only be necessary to calculate the system failure rate once. For the second and further occurrences, the rate calculated in the first subsystem can be used. This could be handled by using the parameter mechanism to create a link between the exchangeable systems. This would need to be a soft link, because there could well be circumstances in which it could be broken (in particular, if we hypothesize the failure of a component in one subsystem but not the other.)

At this point we have sufficient information to calculate the system failure rate. This can be done by propagating the probabilistic information forwards through the graph of Figure 2. Other calculations can be used to explore the system as well. For example, we can hypothesis a system failure (by specifying a distribution with probability 1.0 for system failure on the system failure attribute) and identify the probability of component failures or most likely failure scenarios. We could also hypothesize various component failures and evaluate their impact on the probability of system failure by propagating that information through the graph.

Exploring Problem 1: Basic Calculations

To specify Model 1, we need to specify the 23 attributes of the model and the 23 component or *local* valuations. GRAPHICAL-BELIEF automatically builds the model graph from these relationship definitions: it can determine small pieces of the graph by examining the attributes involved in each valuation. It builds two different representations of the model from this information: one using a directed graph, like the one in Figure 2 and one using an undirected graph. After the model is loaded, it transforms the model into an equivalent “tree” model, smoothing out cycles in the undirected graph by lumping together similar nodes. The tree model is useful internally because there is a one to one correspondence between the “local” component values and the nodes in the tree model. Some of the nodes in the tree models representing component or systems states and others representing “gates” describing the relationship between state variables. (In order to insure this correspondence, GRAPHICAL-BELIEF adds “vacuous” valuations, which do not affect the outcome to otherwise empty nodes).

The tree model is mostly for internal purposes; the user need never be concerned with the tree model as the directed or undirected graphical model serves as a mediating representation for the knowledge in the tree model. The result of this is that there is more than one piece of “local” information associated with many of the nodes in the directed graph. For example, the node *LPCI-SYS* corresponding to the system failure attribute has two pieces of local information: the information about the relationship between system failures and the failure state of the two LPCI-Train subsystems, and local information about just the system failure, a vacuous valuation automatically added by GRAPHICAL-BELIEF. This latter is useful because we can change the this “local” information, for example to condition on failure.

After the model is loaded and the tree model is set up, we are ready to propagate all of the local information to produce the “global” distribution over all attributes. Actually, this global distribution is only a theoretical construct. The fusion and propagation algorithm (Lauritzen and Spiegelhalter[1988], Dempster and Kong[1988], Almond[1990]) calculates various margins of this distribution without ever calculating the full joint distribution. The propagation occurs in the tree model, although through animation, you can see how the calculations in the tree model correspond to those in the directed and undirected model. Finally, you can look at the margin of the global distribution over any attribute (node in the directed or undirected model) or over any set of attributes corresponding to a node in the tree model.

In this case the most interesting node is the probability of system failure, this is 7.82×10^{-7} .

Note that there is no need to calculate cut sets before calculating the probability of failure. This is all handled by the propagation algorithm. Basically it works by calculating the failure probability of successively higher level systems. After the system level failure probability is calculated, information can be propagated back down the model to produce various marginal probabilities of failure. This allows the same model to be used for diagnostic and importance analyses.

Exploring Problem 1: Importance Analysis.

Although the bottom line failure probability is important in accepting the final product (and in this case for regulatory purposes), that is not the only interesting question. A branch of reliability work known as *importance analysis* asks what components and systems are the biggest contributors to system level failure. This form of “explanation,” especially if done early enough in the design process, can allow the engineers to improve the performance of critical systems. There are a number of different approaches to this problem. In GRAPHICAL-BELIEF, tracing the messages back through the model shows which systems and components are the most important.

We start by looking at the messages associated with the messages coming in from one of the LPCI-Trains (recall that messages are really passed along edges in the tree model, so the information comes from both trains at once in this case. We get the following display:

```
Messages received by #: |(LPCI-SYS LPCI-TR-A LPCI-TR-B)|
Local Information:
Conditional Probability over frame:(#<Frame: (LPCI-TR-A LPCI-TR-B)> ;
#<Frame: (LPCI-SYS)>)
Condition      mass          focal element
[(:AND :XX :YY) 1.00000000    {Failed}]
[(:NAND :XX :YY) 1.00000000    {Working}]
Received from #: |(LPCI-SYS)|:          (No information from System level)
Probability over frame:#<Frame: (LPCI-SYS)>
(:FAIL 1.00000000 :NOFAIL 1.00000000 )
Received from #: |(LPCI-TR-B)|:          (LPCI Train B)
Probability over frame:#<Frame: (LPCI-TR-B)>
(:FAIL 8.846978867d-4 :NOFAIL 0.999115302 )
Received from #: |(LPCI-TR-A)|:          (LPCI Train A)
Probability over frame:#<Frame: (LPCI-TR-A)>
(:FAIL 8.846978867d-4 :NOFAIL 0.999115302 )
```

From this information we see: (1) The structure of the local relationship between the LPCI-Trains and the LPCI-System (a parallel or “and”) relationship, (2) there is no information about system failure, and (3) the failure rates for both of the subsystems are equivalent. We arbitrarily choose to follow LPCI Train B.

```
Messages received by #: |(LPCI-S-B PUMP-SYS-B LPCI-TR-B)|
Local Information:
Conditional Probability over frame:(#<Frame: (PUMP-SYS-B LPCI-S-B)> ;
#<Frame: (LPCI-TR-B)>)
Condition      mass          focal element
[(:OR :XX :YY) 1.00000000    {Failed}]
[(:NOR :XX :YY) 1.00000000    {Working}]
Received from #: |(LPCI-TR-B)|:          (No information from Train level)
Probability over frame:#<Frame: (LPCI-TR-B)>
(:FAIL 1.00000000 :NOFAIL 1.00000000 )
Received from #: |(PUMP-SYS-B)|:          (Pump System, Pumps B and D)
Probability over frame:#<Frame: (PUMP-SYS-B)>
(:FAIL 2.437767624d-5 :NOFAIL 0.999975622 )
Received from #: |(LPCI-S-B)|:
Probability over frame:#<Frame: (LPCI-S-B)> (LPCI subsystem, MOV)
(:FAIL 8.603411836d-4 :NOFAIL 0.999139659 )
```

Here we see that the LPCI system is 30–40 times more likely to fail than the Pump System. This makes good sense because of the extra redundancy in the two pump trains within each system. We follow the LPCI system:

```
Messages received by #: |(MOV-25-B CV-46-B LPCI-S-B)|
Local Information:
Conditional Probability over frame:(#<Frame: (MOV-25-B CV-46-B)> ;
#<Frame: (LPCI-S-B)>)
```

```

Condition      mass          focal element
[(:OR :XX :YY) 1.00000000    {Failed}]
[(:NOR :XX :YY) 1.00000000    {Working}]
Received from #:|(LPCI-S-B)|:      (No information from Train level)
Probability over frame:#<Frame: (LPCI-S-B)>
(:FAIL 1.00000000 :NOFAIL 1.00000000 )
Received from #:|(MOV-25-B)|:      (Motor Operated Valve)
Probability over frame:#<Frame: (MOV-25-B)>
(:FAIL 8.106116191d-4 :NOFAIL 0.999189388 )
Received from #:|(CV-46-B)|:      (Check Valve)
Probability over frame:#<Frame: (CV-46-B)>
(:FAIL 4.976990855d-5 :NOFAIL 0.999950230 )

```

Here we see that the Motor Operated Valve (MOV-25-B) is much less reliable than the check-valve. Thus the Motor Operated Valve is the most critical component of the system.

Although useful even in this primitive form, this “explanation” facility is admittedly rather crude. David Madigan (Madigan[1992a] has surveyed the literature about explanation techniques, including some graphical techniques for displaying importance information. The Phase II version of GRAPHICAL-BELIEF will incorporate some of those techniques.

Exploring Problem 1: Sensitivity Analyses

Just as the GRAPHICAL-BELIEF (especially the importance analysis problem described above) can assist in upstream processes like design, it can also assist in downstream processes like maintenance and support. For example, suppose that a routine test reveals that Pump C is defective and needs to be repaired or replaced. Fixing Pump C is likely to take a week, should we require that the reactor be shut down during that week?

The answer to this question will be dependent (in part) of the probability of failure of the LPCI system given that Pump C has failed. This question is simple to answer with GRAPHICAL-BELIEF. We simply modify the factor of the model corresponding to Pump C to indicate that Pump C has failed. Propagating this information throughout the model yields a new estimate of failure probability: 5.125×10^{-6} . More complex scenarios, even ones involving multiple failures, are just as simple to calculate.

A graphical reliability model could be used as part of a mission planning expert system. For example, imagine that the modelled system was the air circulation system on a manned spacecraft. If one of the redundant pumps fails, mission control needs to decide whether to pull an astronaut from other time critical tasks to repair the system. As GRAPHICAL-BELIEF is designed as a general purposes graphical modelling tool, eventually mission planning models and reliability models could be combined to help address mission scheduling problems arising from on board failures.

The manufacturing department might be interested in a different kind of question. Returning to the example of the LPCI system, they might have obtained a competitive bid on the four pumps and want to know what the impact on overall reliability would be if they substituted the new less expensive pumps with a higher failure rate (say 4 times as high). Again we merely need to substitute the failure rate for each pump with that of the new pumps. In the Phase II version of GRAPHICAL-BELIEF, we can use the parameter feature to substitute for all pumps at the same time. Using the alternative pumps, the new estimate of failure probability is: 1.5470498×10^{-6} .

Exploring Problem 1: Diagnosis

Another use for the system at this level would be to look at the “diagnostic” problem of identifying the most likely faults given a system failure. This is very similar to the importance analysis, only now we assume that we can obtain test data about the system if we need it. As we obtain new information, we make temporary changes to the model expressing that new information. Updating the model tells us what to do next.

We start by setting the local value associated with LPCI-SYS attribute to failure with probability 1.0 and propagating the changes. We now use part of the GRAPHICAL-BELIEF command language which prints the failure rates for all variables (components and subsystem) in the model:

```
Node:  LPCI-SYS
Probability over frame:#<Frame:  (LPCI-SYS)>
(:FAIL  1.00000000      :NOFAIL  0.00000000      )
Node:  LPCI-TR-A
Probability over frame:#<Frame:  (LPCI-TR-A)>
(:FAIL  1.00000000      :NOFAIL  0.00000000      )
Node:  LPCI-S-A
Probability over frame:#<Frame:  (LPCI-S-A)>
(:FAIL  0.972468903     :NOFAIL  2.753109676d-2)
Node:  MOV-25-A
Probability over frame:#<Frame:  (MOV-25-A)>
(:FAIL  0.916258116     :NOFAIL  8.374188377d-2)
Node:  CV-46-A
Probability over frame:#<Frame:  (CV-46-A)>
(:FAIL  5.625638910d-2  :NOFAIL  0.943743611     )
Node:  PUMP-SYS-A
Probability over frame:#<Frame:  (PUMP-SYS-A)>
(:FAIL  2.755480329d-2  :NOFAIL  0.972445197     )
Node:  PUMP-TR-A
Probability over frame:#<Frame:  (PUMP-TR-A)>
(:FAIL  3.233254084d-2  :NOFAIL  0.967667459     )
Node:  CV-48-A
Probability over frame:#<Frame:  (CV-48-A)>
(:FAIL  3.259196348d-4  :NOFAIL  0.999674080     )
Node:  PUMP-A
Probability over frame:#<Frame:  (PUMP-A)>
(:FAIL  3.200821425d-2  :NOFAIL  0.967991786     )
```

[Redundant components and systems, whose values are symmetric, have been removed to reduce the size of the included output.]

Here we can see that the most likely fault is in the motor operated valve, whose conditional failure probability is about 91%. This would be the first case to look in the case of a system failure. We therefore send the maintenance team to check the MOV's. They report that MOV-25-A has in fact failed, but MOV-25-B is working properly; we update our model with this new information and repropagate.

At this point the model reveals that the probability that CV-46-B has failed is .67 and the probability that each of Pump D and Pump B is failed is .32. This means that the next place to check would be the check valve and then the pumps. We could continue this procedure as needed, conditioning on new data as it arrived from the field teams.

To really use the system in diagnosis, we would want to augment the design information with information about test points and procedures. This would mean that the graphical diagnostic model would be the graphical reliability model plus some additional test information. Recall that one of the important sources of information is conditional probability models which can be used to describe the probability of a symptom or test result given an error condition.

An important part of the engineering life cycle is evaluation of field returns and failures. These provide new information about the system, especially about the system failure models and the failure rates. New Bayesian updating techniques can update the information about the component and system level failure

rates from the data from field tests and returns. These field data can also be used to critique the model, insuring that all critical failure modes (especially common cause failures) have been identified.

A final use for the GRAPHICAL-BELIEF model would be in generating scenarios for a training program. The graphical belief model describes the joint failure probability of components and subsystems in the system. If we modify the propagation algorithm so that we propagate going up and simulate (draw from the joint probability distribution) going in the reverse direction, we can generate a random “scenario.” Thus we can generate random failure states for use in training and evaluation of maintenance procedures and personnel.

Problem 2: Uncertain (Noisy) Relationships

Martz and Waller have some additional information from an IEEE study about two of the subsystems, the Pump Trains and the LPCI Subsystem. These data suggest that the subsystems are perhaps not quite as reliable as the individual component data would suggest. This can be modelled in a number of ways. Unfortunately, the subsystem level data is not known to be independent. Martz and Waller develop a method for finding the distribution of failure rates based on an estimate of of the dependence; we will use a simple procedure which assumes independence.

The model we build is based on the fact that the distribution we elicited for the Pump Train system failure in Figure 3 is in fact a conditional probability distribution. The distribution says that if either component has failed, then the system fails with probability 1.0 (certainty) and if both components work, then the system works with probability 1.0 (certainty). If the data from the IEEE study is relevant, this latter conditional distribution may not be realistic. In particular, we may want to modify the distribution shown in Figure 3 by reducing the probability that the system will work even if both components are working. Using the data from Martz and Waller’s Table 3 and applying their discounting factor of .25 suggests that 0.995 might be a reasonable value for the probability that the system works when all components are operating correctly. The resulting distribution, shown in Figure 6, is a “Noisy Or” model, similar to those described in Pearl[1988]. A similar calculation suggests a probability of 0.998 for the “noise” (unanticipated failure) probability of the LPCI subsystem.

Save
Consequences
Conditions
Values

Conditions: (PUMP-A CV-48-A)

Vector: #((:FAIL :NOFAIL) (:FAIL :NOFAIL))

Consequences: (PUMP-TR-A)

Vector: #((:FAIL :NOFAIL))

	<i>Working</i>	<i>Failed</i>
(:OR :XX :YY)	<input type="text" value="0.00000"/>	<input type="text" value="1.00000"/>
(:NOR :XX :YY)	<input type="text" value="0.99500"/>	<input type="text" value="0.00500"/>

Read Only?

Probability
Belief

Figure 6. Noisy Valuation for Relationship between Pump A, CV-48 A and Pump Train A

This particular model is rather artificial from the point of view of reliability modelling. It essentially assigns a small probability to cases where the system fails for some mysterious reason even though none of the components is at fault. Although it is realistic for a system to fail for an unanticipated reason, the model used in Figure 6 has a small probability of producing an undiagnosable failure. More useful would be to assume another failure mode, a “subsystem fault” and to include that as another attribute in the model. Thus the graph fragment associated with the Pump Train subsystem would be as shown in Figure 7. The new Pump Train Fault event would have probability of .005 of occurrence (or some other value based on system level data).

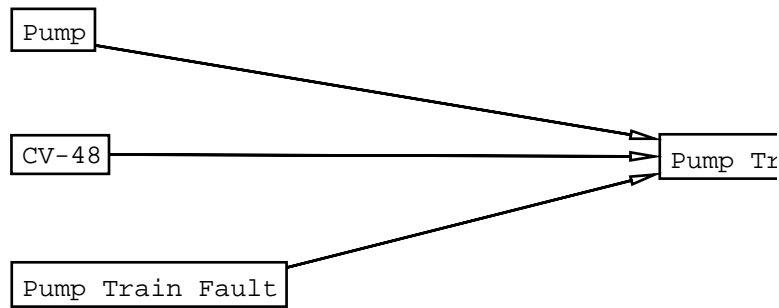


Figure 7. Pump Train Model with Subsystem Fault

Although the model which introduces a new failure mode is more useful for diagnosis, the ability to use general probability and conditional probability (and even belief function) relationships has other uses. The technique used in Figure 6 can be used to perform a sensitivity analysis for the subsystem level data (*i.e.*, if we lower the reliability of the subsystem will this greatly effect the overall system reliability). The ability to perform such calculations could enable project managers to channel design and reliability modelling effort into the most critical safety systems. It also allows us to model much more complex dependencies than simple logical ones. For example, in a medical problem we could model the probability of various symptom states given various disease states, or in a control problem the probability of a sensor report given a system state. We will continue to use the slightly unrealistic model of Figure 6 in order to illustrate these capabilities of the system.

Exploring Problem 2

To effect this change in the model, need only replace 6 of the local valuations. We can do this either by editing the valuations with the valuation editor or reading them in from a file. As we have added no new nodes or edges to the model, we do not need to recompute the tree model, merely reload the local values. We can then propagate to get the final result.

Adding the additional uncertainty increase the failure probability from 7.82×10^{-7} to 8.74×10^{-6} .

Note that there is an important difference between our use of system level data and that of Martz and Waller. A number of authors (*e.g.*, Speed[1985]) have criticized the probabilistic risk assessment (PRA) models for failure to use the system level data. However, the heart of Speed’s critique is that such data can be used to evaluate the adequacy of the model. One of the great fears of the modelling process is that some important failure mode (especially a common cause failure) will be overlooked. Our model introduces a parameter which describes the probability that such an overlooked situation will occur in a given system, and uses the system level test data to estimate the value of that parameter. Martz and Waller[1990] does not directly address this problem of model adequacy.

Problem 3: Common Cause Failure Modes

Common cause failures have been a critical concern in reliability studies since the famous WASH-1400 study. The problem with common cause failures is that they violate the critical independence assumptions made when calculating the failure rate for redundant systems. For example, suppose that the motor operated valves were sensitive to the presence of live steam in the reactor room. Then, if a pipe burst were to fill the room with steam, the failure probability of both MOV-25 A and MOV-25 B would increase. This would make a system failure much more likely.

Fortunately, most of the difficulty with common cause failures lies with anticipating them, not with modelling them. Recall that independence in a graphical model is represented by separation in the graph. Thus to represent dependence on a common cause failure, we need only create a new attribute (node) for the common cause failure mode and link it to the components whose failure it causes. Figure 8 shows the modified model. In order to perform calculations with this model, we will assume that the chance of live steam is 1 in a thousand and that the presence of live steam makes each MOV 10 times more likely to fail.

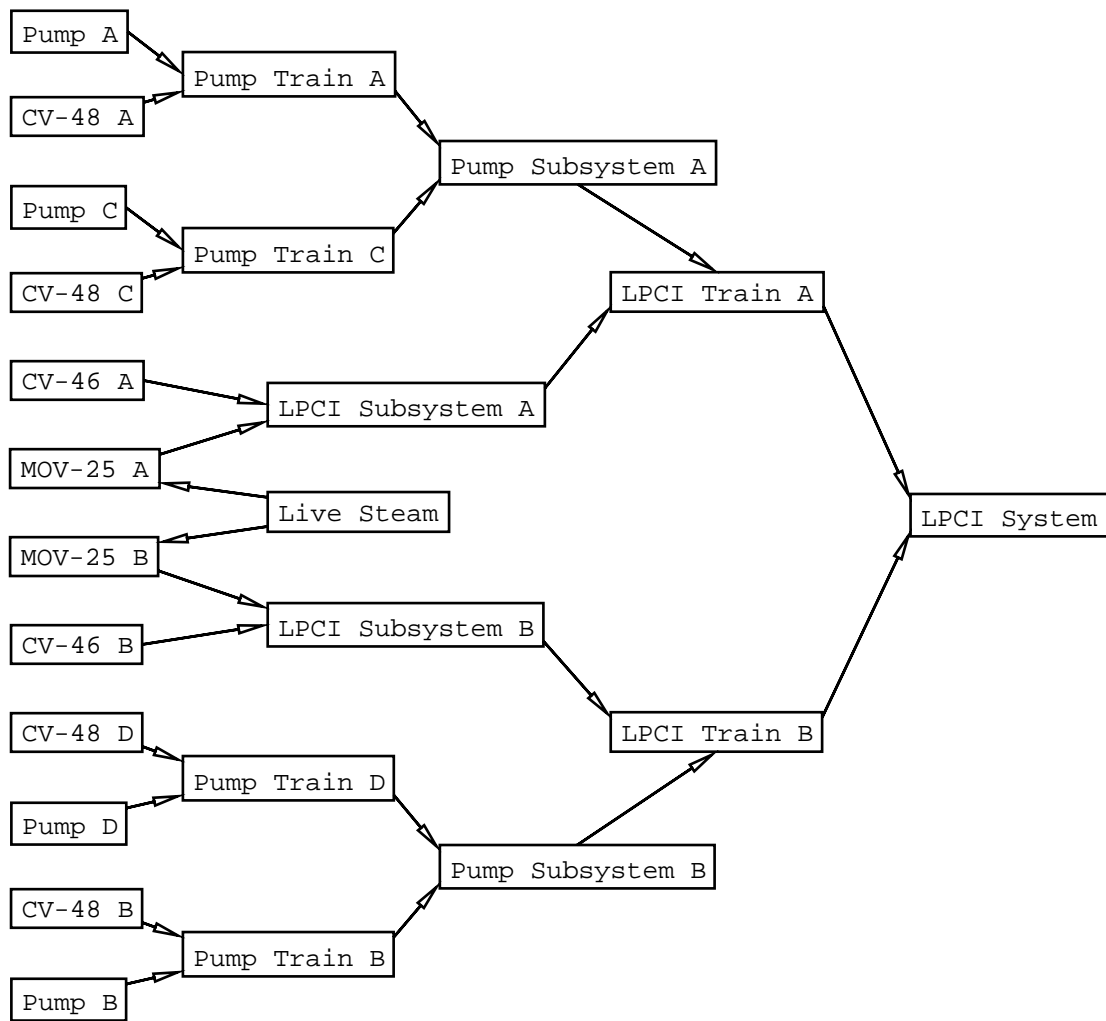


Figure 8. Adding “Live Steam” Common Cause Failure

Note that the inclusion of the Live Steam node creates a loop in the graph. As this is not a directed cycle (you can’t go around the loop by following the arrows), this does not present any difficulty. There is some additional computational complexity introduced by these calculations, but not much. A great number of common cause failures, or ones that were very far upstream (and hence created large undirected

loops) could add more computational complexity to the system than can be handled by the current version of GRAPHICAL-BELIEF. These problems can be addressed by a different computation mechanism; and should be addressed in the Phase II version.

Exploring Problem 3

Because we have introduced new nodes and edges, we need to recompute the tree model. In the course of this computation we introduce some new edges to “fill-in” cycles in the undirected model. We add vacuous values to the nodes in the tree model corresponding to these cycles. Their purpose is merely to insure that the information about dependent components is propagated together. Figure 9 shows the undirected model graph filled in cycles.

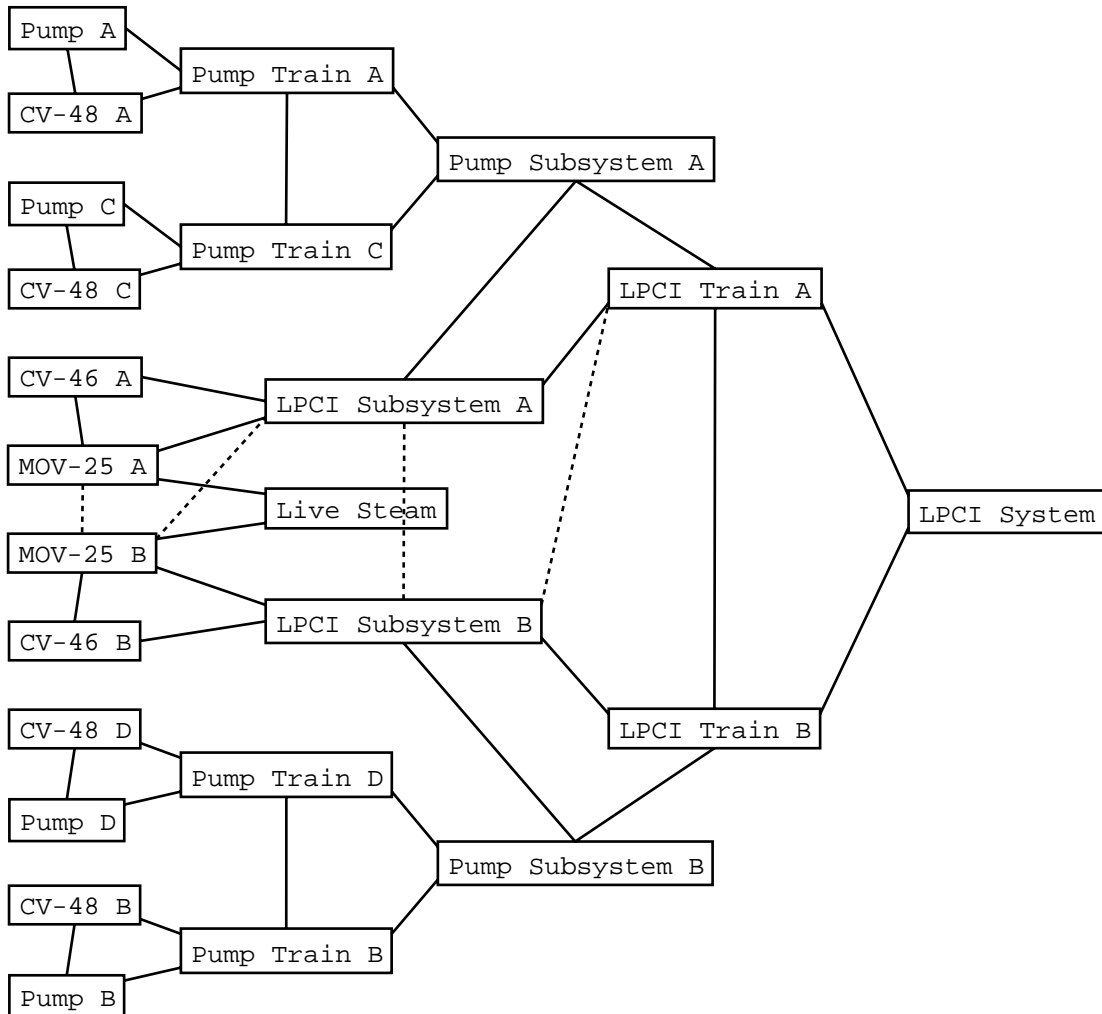


Figure 9. Adding “Live Steam” Common Cause Failure

After adding the common cause failure, the system failure probability increases to 5.62×10^{-5} .

Although this method is extremely useful for common cause failures which introduce small local dependencies, major common cause failures are still a problem. Although it is not explicitly stated in Martz and Waller[1990], their model almost certainly assumes that both the power supply and the control system are operating properly. If the failure of either the power supply or the control system cause a failure of the LPCI-system with probability 1.0, this would likely be handled at the level of a global model for all systems. In some circumstances it would be appropriate to build two different models for different circumstance, for

example, the probability of a plan landing correctly in both a powered and an unpowered landing. Geiger and Heckerman[1991] describe these conditional models in more detail.

Problem 4: Failures in Time

The problem described in Martz and Waller[1990] is relatively simple, in part, because it is only concerned with calculating the demand unavailability of the system. More complex systems require both the calculation of failure on demand rates and failure during operation rates. Although, we will not demonstrate this with the current version of GRAPHICAL-BELIEF the extension from failure on demand to failure during operation adds additional complexity, but does not prevent the same methods from being used to calculate failure rates.

Failure on demand components and systems can be studied with a relatively simple model, the Bernoulli process. Each test—opportunity for the system or component to work or not work—is considered a discrete point in time. At each time point, the system either works or does not work. As all the tests are considered to be “exchangeable” (performed under equivalent conditions), the probability of a failure at any demand point is the same. This naturally leads to the fact that the number of failures in n tests will follow a binomial distribution with parameters n and p , where p is the failure rate for the system or component.

As the number of tests (demands) and number of failures are the sufficient statistics for this binomial distribution, the data about the component or system can be summarized in terms of these components. As the beta distribution is the natural conjugate for the binomial, it is very attractive to model uncertainty about the failure rate p in terms of a beta distribution or class of beta distributions, especially as the parameters of the beta distribution (with a little re-expression) can be interpreted in terms of the sufficient statistics: an approximate number of failures and tests.

To make the transition to failures in time requires introducing two additional time constants: the amount of time for which the process has been observed and the amount of time for which the system will be required to function. The amount of time for which the system will be required is a true nuisance parameter, especially if it varies between subsystems, however, it is a mainly a matter of bookkeeping. The time for which the process has been observed, plays the role that the number of tests played for failure on demand data.

The simplest model for failure in time data is the Poisson process. This process makes the simplifying assumption that the failure rate is constant throughout the lifetime of the component. As this is likely to be approximately true for short periods of time, it is relatively simple to build up an approximate model for the full system lifetime from Poisson models for short time stretches. It is called the Poisson process because for an amount of time t and failure rate α , the probability of seeing k events follows a Poisson distribution with parameter αt . For short time periods t and low failure rates α , the probability of multiple failures is usually negligible. In many situations, repair during the course of the system operation is infeasible, so often it is desirable to reduce the Poisson process to a failure/no failure calculation. As the probability of no event (failure) during the time interval t is $e^{-\alpha t}$, the probability of one or more event (failure) during the time interval is $1 - e^{-\alpha t}$, which is approximately αt when αt is small.

The sufficient statistics for the Poisson process are the number of failures and the amount of observation time for the system. The natural conjugate family for the failure rate is the gamma distribution, whose parameters can be interpreted in terms of the sufficient statistics. These models are described in greater detail in Almond[1991a,b]. Thus simply by allowing for gamma and beta distributions over parameters, and extra parameters to handle time constants, the Phase II version of GRAPHICAL-BELIEF should be able to handle both failure on demand and failure in time systems.

A related and far more difficult problem is to try and perform reasoning about the times of failure events. For example, given a fault in the system, what is the effect on the mean time to failure of a critical system. Again, David Madigan has compiled a survey of the literature on this problem (Madigan[1992]), some of the simpler methods could be incorporated into the Phase II or other future development.

References

- Almond, Russell G. [1990].** *Fusion and Propagation in Graphical Belief Models: An Implementation and an Example*. Ph.D. dissertation and Harvard University, Department of Statistics Technical Report S-130. To be published as a monograph from Van Nostrand Reinhold.
- Almond, Russell G. [1991a].** "Building Blocks for Graphical Belief Models." *Journal of Applied Statistics*, **18**, 63–76.
- Almond, Russell G. [1991b].** "Belief Function Models for Simple Series and Parallel Systems." Personal Technical Report A-6. Technical Report 207, University of Washington, Department of Statistics.
- Almond, Russell G. [1992].** "ELTOY: A tool for Education and Elicitation," Documentation for ELTOY system, distributed through `statlib@stat.cmu.edu`.
- Geiger, Dan and Heckerman, David[1992].** "Advances in Probabilistic Reasoning." In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, Interface Foundation of North America, pp 22-29.
- Madigan, David[1992a].** "Approaches to Explanation in Bayesian Networks," Statistical Science Research Report #8.
- Madigan, David[1992b].** "Temporal Reasoning with Probabilities: A Review," Statistical Science Research Report #7.
- Martz, H. F. and Waller, R. A. [1990].** "Bayesian Reliability Analysis of Complex Series/Parallel Systems of Binomial Subsystems and Components." *Technometrics* (**32**), pp 407–106.
- Speed, T. P. [1985].** "Probabilistic Risk Assessment in the Nuclear Industry: WASH-1400 and Beyond." *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer*, Le Cam, Lucien M. and Olshen, Richard A. (eds.), **1**, 173-200.
- Pearl, Judea [1988].** *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.