Delft University of Technology
Software Engineering Research Group
Technical Report Series

Techniques for Diagnosing Software Faults

Rui Abreu, Peter Zoeteweij, and Arjan J.C. van Gemund

Report TUD-SERG-2008-014

TU Delft

SERG

TUD-SERG-2008-014

Published, produced and distributed by:

Software Engineering Research Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

ISSN 1872-5392

Software Engineering Research Group Technical Reports:
http://www.se.ewi.tudelft.nl/techreports/

For more information about the Software Engineering Research Group:
http://www.se.ewi.tudelft.nl/

— This document is reporting work in progress, and thus likely to change —

Techniques for Diagnosing Software Faults

Rui Abreu **Peter Zoeteweij** **Arjan J.C. van Gemund**

Software Technology Department
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology
P.O. Box 5031, NL-2600 GA Delft, The Netherlands
{r.f.abreu, p.zoeteweij, a.j.c.vangemund}@tudelft.nl

1 Introduction

This technical report is meant to report our findings and ideas with respect to spectrum-based fault localization and model-based diagnosis. In the following we want to introduce and compare model-based diagnosis (MBD), spectrum-based fault localization (SFL) and our contributions using 3-inverters as a running example (which is simple, yet sufficiently interesting).

The remainder of this paper is organized as follows. The concepts and definitions used in this paper are given in the next section. The combination of model-based diagnosis and Bayesian reasoning, as it is normally applied to, e.g., digital circuits, is discussed in Section 3. Spectrum-based fault localization, including the system transformation for instrumentation to collect data to reason about failures is discussed in Section 4.1. In Section 4.2 we investigate several novel approaches for applying model-based diagnosis, and notably Bayesian reasoning to systems that have been prepared for spectrum-based fault localization.

2 Preliminaries

Definition 1 *By a system under observation, or* system *we mean a tuple $\langle C, V, S\rangle$, where*

- *C is a finite, non-empty set of* components *$\{c_1, \ldots, c_n\}$.*
- *V is a finite, non-empty sequence $x_1, \ldots, x_k$ of* observable variables*, with respective domains $\mathcal{D}_1, \ldots, \mathcal{D}_k$,*
- *$S \in \mathcal{D}_1 \times \ldots \times \mathcal{D}_k$ represents the specified behavior.*

Further, by an observation *we mean a tuple $\langle v_1, \ldots, v_k\rangle \in \mathcal{D}_1 \times \ldots \times \mathcal{D}_k$. An observation obs $\notin S$ is called a* failure*.*

With each component $c_m \in COMPS$ we associate a *health variable* h_m which denotes component health. The health states of a component are healthy (*true*) and faulty (*false*), but this concept can easily be generalized to any finite domain [**?**].

Definition 2 *An h-literal is h_m or $\neg h_m$ for $c_m \in COMPS$.*

Definition 3 *An h-clause is a disjunction of h-literals containing no complementary pair of h-literals.*

Definition 4 *Let S_N and S_P be two disjoint sets of healthy and faulty components, respectively, such that $C = \{m \mid m \in S_N \cup S_P\}$ and $S_N \cap S_P = \emptyset$. A diagnosis candidate is $d_k(S_N, S_P)$*

$$(\bigwedge_{c_m \in S_N} \neg h_m) \wedge (\bigwedge_{c_m \in S_P} h_m)$$

Definition 5 *A diagnosis $D = \{d_1, \ldots, d_k, \ldots, d_K\}$ is an ordered set of all K diagnosis candidates.*

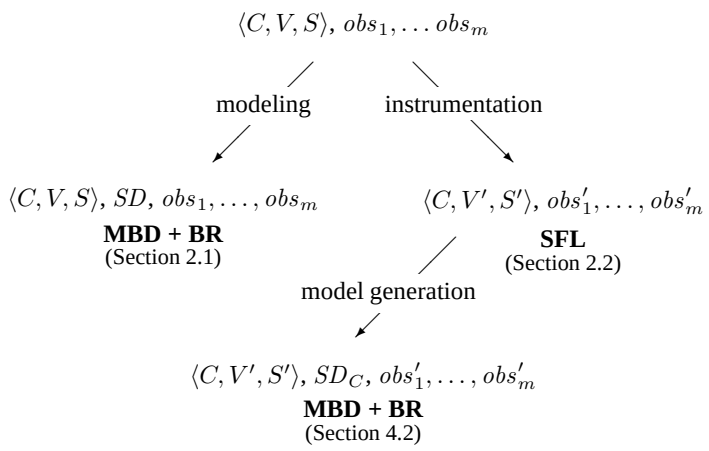


Figure 1. Approaches to diagnosis considered in this paper

For simplicity, we refer to d in terms of a set of the negations literals only.

The purpose of diagnosis is to identify the component, or combination of components that causes observed system failures, and the starting point of such an analysis is a system $\langle C, V, S\rangle$, and a sequence of observations $obs_1, \ldots, obs_m$ that contains at least one failure. Note that in our definition of a system under observation the components in C are not related to the observations, and without further information we cannot exclude any diagnosis candidate from the powerset of C except the empty diagnosis candidate, which indicates that all components function correctly.

In this report we will describe and compare several techniques that help make meaningful selections of diagnosis candidates in one or both of the following ways:

- by reducing the number of diagnosis candidates, and
- by ranking diagnosis candidates with respect to the likelihood that they explain the observations.

Figure 1 illustrates the combinations of techniques that we will consider.

The primary technique for reducing the number of diagnosis candidates is *model-based diagnosis* (MBD). It entails that the system description $\langle C, V, S\rangle$ is complemented with a model, on the basis of which we can exclude diagnosis candidates that do not logically explain all observations. The number of remaining diagnosis candidates is typically large, and Bayesian reasoning (BR) is normally applied as a companion to model-based diagnosis, to rank the remaining diagnosis candidate with respect to the probability that they reflect reality in presence of the observed behavior. In addition to providing the ranking, the calculated probabilities can also play a role in determining the quality of a diagnosis, and to guide the search for valid diagnosis candidates, but these applications are outside the scope of this paper.

As a second technique for ranking diagnosis candidates, we will consider spectrum-based fault localization (SFL). In this case, the observations relate to the activity of the components, and the diagnosis candidates are ranked according to the extent to which this activity coincides with the occurrence of failures. The measurements required for SFL are achieved through instrumentation, and we model this by a transformation of a system into a variant of that system that provides the necessary observations.

Traditionally, MBD+BR and SFL are applied to hardware and software, respectively. Some approaches to model-based software diagnosis exist, but a major problem with these approaches is that in general, neither the software models used in the development cycle, nor the models that can be derived from existing code, allow for a significant reduction of the number of diagnosis candidates. In this paper we investigate a different approach, where Bayesian reasoning is applied to calculate the probability that diagnosis candidates are supported by observations in the context of the instrumented system, which we also use for SFL. In addition, model-based diagnosis based on a simple, automatically generated causal model (SD_C in Figure 1) is used to counter the computational complexity.

2.1 Model-based Diagnosis

Without loss of generality, in this section we consider *digital systems*, which we define to be systems whose variable domains $\mathcal{D}_1, \ldots, \mathcal{D}_k$ equal the set $\{\mathbf{true}, \mathbf{false}\}$.

Definition 6 *A* system description *for a digital system $\langle C, V, S\rangle$ is a propositional formula M that involves at least the following propositional variables:*

- *the observable variables of the system, $x_1, \ldots, x_k$,*
- *$h_1, \ldots, h_n$, where n is the number of components in C. These are the so-called* health variables*, where h_i represents the proposition that component c_i is healthy (functioning correctly).*

Furthermore we require that

$$h_1 \wedge \ldots \wedge h_n \ \wedge \ x_{obs^+} \ \wedge \ x_{obs^-} \ \wedge \ M \ \not\models \ \bot \quad \textit{iff} \quad obs \in S,$$

where for $obs := \langle v_1, \ldots, v_k\rangle$, x_{obs^+} denotes the conjunction of literals x_i for which $v_i = \mathbf{true}$, and x_{obs^-} denotes the conjunction of literals $\neg x_i$ for which $v_i = \mathbf{false}$.

Now a diagnosis candidate d is called a diagnosis candidate for the combination of a system $\langle C, V, S\rangle$, a description of that system, and a single observation *obs* iff

$$h_{d^+} \ \wedge \ h_{d^-} \ \wedge \ x_{obs^+} \ \wedge \ x_{obs^-} \ \wedge \ SD \ \not\models \bot,$$

where h_{d^+} denotes the conjunction of literals h_i for which $c_i \in d$, and where h_{d^-} denotes the conjunction of literals $\neg h_i$ for which $c_i \notin d$, and where x_{obs^+} and x_{obs^-} are as in Definition 6. The notion of a diagnosis for a single observation is extended to the notion of a diagnosis for a sequence of observations by requiring that the above condition holds for all observations in the sequence.

2.2 Observation-based Fault Localization

Given a system $\langle C, V, S\rangle$ and a sequence of observations $obs_1, \ldots, obs_m$, we define $\langle C, V', S'\rangle$ and $obs'_1, \ldots, obs'_m$ to be as follows.

- $V' := a_1, \ldots, a_n, e$, i.e., the number of variables in the modified system is equal to the number of components, plus one. All variables of V' have domain $\{\mathbf{true}, \mathbf{false}\}$.
- For $obs'_i := \langle a_1, \ldots, a_n, e\rangle$ we have
  - a_j indicates whether or not component c_j was involved in the computation that resulted in observation obs_i,
  - e indicates whether obs_i is a failure or not, i.e., e is equal to the truth value of the condition $obs_i \in S$.
- We define S' to contain those observations $obs'_i := \langle a_1, \ldots, a_n, e\rangle$ having $e = \mathbf{false}$.

The set of observations are stores in a so-called *observation matrix*, which is defined as follows

Definition 7 *Let M be the number of components, and N the number of execution runs. Let O denote the $N \times (M + 1)$* observation matrix*. For $j \leq M$, the element o_{ij} is equal to 1 (*true*) if component j was observed to be involved in the execution of run i, and 0 (*false*) otherwise. The element $o_{i,M+1}$ is equal to 1 (*true*) if run i failed, and 0 (*false*) otherwise. The rightmost column of O is also denoted as e (the error vector).*

From O it is also possible to derive the probability r that a component is actually executed in a run (expressing code coverage), and the probability g that a faulty component is actually exhibiting good behavior (expressing fault coverage, also known as the "goodness" parameter g from MBD [2]).

2.3 Bayes' Rule

Throughout this paper, components are assumed to fail independently. Therefore, in absence of any observation the probability a particular diagnosis $d(\Delta, C - \Delta)$ is correct is:

$$\Pr(d) = \prod_{c_m \in \Delta} \Pr(\neg h_m) \cdot \prod_{c_m \in C - \Delta} (1 - \Pr(\neg h_m))$$

where $\Pr(\neg h_n)$ is the given probability that component c_m is faulted (not healthy). The probability for diagnosis d being correct after an observation obs is given by Bayes' rule:

$$\Pr(d|SD \wedge obs) = \frac{\Pr(SD \wedge obs|d) \cdot \Pr(d)}{\Pr(SD \wedge obs)}$$

The denominator $\Pr(SD \wedge obs)$ is a normalizing term that is identical for all d and thus needs not to be computed directly. Thus,

$$\Pr(d|SD \wedge obs) = \alpha \cdot \Pr(SD \wedge obs|d) \cdot \Pr(d)$$

$\Pr(SD \wedge obs|d)$ is defined as

$$\Pr(SD \wedge obs|d) = \begin{cases} 0 & \text{if } d \text{ and } SD \wedge obs \text{ are inconsistent} \\ 1 & \text{if } d \text{ logically follows from } SD \wedge obs \\ \epsilon & \text{if neither holds} \end{cases}$$

where various policies ϵ are possible [1]: different values for ϵ will be considered in the subsequent sections of this paper.

For multiple observations, Bayes' rule can be applied in sequence. Thus, after a set of m observations $SO = \{obs_1, \ldots, obs_n\}$ the probability a particular diagnosis d is correct given by applying recursively the Bayes' rule, yielding

$$\Pr(d|SD \wedge SO) = \alpha \cdot \Pr(SD \wedge obs_1|d) \cdot \ldots \cdot Pr(SD \wedge obs_n|d) \cdot \Pr(d)$$

As SD does not change, we use $\Pr(d|obs)$ instead of $\Pr(d|SD \wedge obs)$ for simplicity.

2.4 3-inv

The circuit in Figure 2 will be the running example throughout this paper.

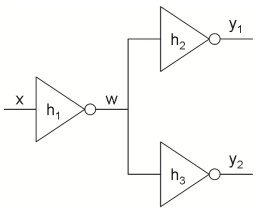


Figure 2. 3-inverters example

3 Model-Based Diagnosis

A weak model of an inverter component c is given by

$$h \Rightarrow y = \neg x$$

Consequently, the circuit is modeled by

$$h_1 \Rightarrow w = \neg x$$
$$h_2 \Rightarrow y_1 = \neg w$$
$$h_3 \Rightarrow y_2 = \neg w$$

Consider the observation $obs = ((x, y_1, y_2) = (1, 1, 0))$. It follows

$$h_1 \Rightarrow \neg w$$
$$h_2 \Rightarrow \neg w$$
$$h_3 \Rightarrow w$$

which equals

$$(\neg h_1 \vee \neg w)$$
$$(\neg h_2 \vee \neg w)$$
$$(\neg h_3 \vee w)$$

Resolution yields

$$(\neg h_1 \vee \neg h_3) \wedge (\neg h_2 \vee \neg h_3)$$

also known as conflicts [4], meaning that (1) at least c_1 or c_3 is at fault, and (2) at least c_2 or c_3 is at fault.

The minimal diagnoses are the minimal hitting set [7], given by

$$\neg h_3 \vee (\neg h_1 \wedge \neg h_2)$$

Thus either c_3 is at fault (single fault), or c_1 and c_2 are at fault (double fault). Given the weak model, any other fault combination that is subsumed by the above, two minimal diagnoses, is a valid diagnosis.

Assuming all inverters have equal a priori fault probability, clearly, the single fault has higher probability, i.e., should rank higher than the double fault candidate. The posterior probability of the diagnoses, given the observations, is computed using Bayes' rule, updating the prior probability according to the extent the observation is explained by the candidate diagnosis as explained in Section 2.3. Thus,

$$\Pr(\neg h_3|obs) = \alpha \cdot \Pr(obs|\neg h_3) \cdot \Pr(\neg h_3)$$
$$\Pr(\neg h_1 \wedge \neg h_2|obs) = \alpha \cdot \Pr(obs|\neg h_1 \wedge \neg h_2) \cdot \Pr(\neg h_1 \wedge \neg h_2)$$

Let $\Pr(\neg h_c) = p$ and assume components fail independently, the prior probabilities are given by

$$\Pr(\neg h_3) = p$$
$$\Pr(\neg h_1 \wedge \neg h_2) = p^2$$

If ϵ is defined to be one divided by the number of observations that can be explained by a given diagnosis, and since there are 4 possible observations that can be explained by $\neg h_3$, and 8 possible observations that can be explained by $\neg h_1 \wedge \neg h_2$ it follows that

$$\Pr(obs|\neg h_3) = \frac{1}{4}$$
$$\Pr(obs|\neg h_1 \wedge \neg h_2) = \frac{1}{8}$$

Consequently,

$$\Pr(\neg h_3|obs) = \alpha \cdot \frac{1}{4} \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|obs) = \alpha \cdot \frac{1}{8} \cdot p^2$$

As the two minimal diagnoses are independent (weak fault model), both must sum up to 1, determining α.

For $p = 0.01$ it follows

$$\Pr(\neg h_3|obs) \approx 0.995$$
$$\Pr(\neg h_1 \wedge \neg h_2|obs) \approx 0.005$$

Instead of accounting for the scaling constant α such that the posterior probabilities sum up to 1, we can also explicitly compute $\Pr(obs)$. By definition, as explained in Section 2.3, the conditional probability is calculated as follows

$$\Pr(\neg h_3|SD \wedge obs) = \frac{\Pr(\neg h_3 \wedge (SD \wedge obs))}{\Pr(SD \wedge obs)}$$
$$\Pr(\neg h_1 \wedge \neg h_2|SD \wedge obs) = \frac{\Pr((\neg h_1 \wedge \neg h_2) \wedge (SD \wedge obs))}{\Pr(SD \wedge obs)}$$

For $obs = ((x, y_1, y_2) = (1, 1, 0))$, the solution $\varsigma = SD \wedge obs = \neg h_3 \vee (\neg h_1 \wedge \neg h_2)$ holds,

$$\Pr(\neg h_3|\varsigma) = \frac{\Pr(\neg h_3)}{\Pr(\neg h_3) + \Pr(\neg h_1) \cdot \Pr(\neg h_2) - \Pr(\neg h_1) \cdot \Pr(\neg h_2) \cdot \Pr(\neg h_3)}$$
$$\Pr(\neg h_1 \wedge \neg h_2|\varsigma) = \frac{\Pr(\neg h_1 \wedge \neg h_2)}{\Pr(\neg h_3) + \Pr(\neg h_1) \cdot \Pr(\neg h_2) - \Pr(\neg h_1) \cdot \Pr(\neg h_2) \cdot \Pr(\neg h_3)}$$

Thus, for $p = 0.01$

$$\Pr(\neg h_3|\varsigma) = \frac{p}{p + p^2 - p^3} = 0.99$$
$$\Pr(\neg h_1 \wedge \neg h_2|\varsigma) = \frac{p^2}{p + p^2 - p^3} = 0.01$$

Now suppose that there is a second observation $obs' = ((x, y_1, y_2) = (1, 1, 1))$, which does not reveal any faulty behavior. Using the same reasoning as for the first obs, all possible diagnoses explain obs'

$$\begin{aligned} d_1 &= (h_1 \wedge h_2 \wedge h_3) \\ d_2 &= (h_1 \wedge h_2 \wedge \neg h_3) \\ &\dots \\ d_7 &= (\neg h_1 \wedge \neg h_2 \wedge h_3) \\ d_8 &= (\neg h_1 \wedge \neg h_2 \wedge \neg h_3) \end{aligned}$$

As mentioned before, probabilities are updated as follows

$$\Pr(d_k|\{obs, obs'\}) = \alpha \cdot \Pr(obs'|d_k) \cdot \Pr(obs|d_k). \Pr(d_k)$$

Due to the first observation, we only consider the two minimal diagnoses d_2 and d_7. Thus

$$\Pr(\neg h_3|\{obs, obs'\}) = \alpha \cdot \Pr(obs'|\neg h_3) \cdot \Pr(obs|\neg h_3).p$$
$$\Pr(\neg h_1 \wedge \neg h_2|\{obs, obs'\}) = \alpha \cdot \Pr(obs'|\neg h_1 \wedge \neg h_2) \cdot \Pr(obs|\neg h_1 \wedge \neg h_2).p^2$$

Similarly to the previous observation, it follows that

$$\Pr(obs'|\neg h_3) = \frac{1}{4}$$
$$\Pr(obs'|\neg h_1 \wedge \neg h_2) = \frac{1}{8}$$

Consequently

$$\Pr(\neg h_3|\{obs, obs'\}) = \alpha \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot p = \alpha \cdot \frac{1}{16} \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|\{obs, obs'\}) = \alpha \cdot \frac{1}{8} \cdot \frac{1}{8} \cdot p^2 = \alpha \cdot \frac{1}{64} \cdot p^2$$

4 Observation-based Diagnosis

In the following we assume that we cannot apply model-based techniques to derive diagnoses. Consider the same circuit. However, now we necessarily abstract from system structure and component behavior. Observations are associated with pass or fail information. Hence, the following observation matrix O is obtained:

1	1	0	0	obs_1 (c_1 and c_2 are involved, leading to y_1 = 1, i.e., a pass)
1	0	1	1	obs_2 (c_1 and c_3 are involved, leading to y_2 = 0, i.e., a fail)

There are generally two approaches towards diagnosing the above problem. The first approach, SFL, is popular in software. The second approach is based on logic reasoning, similar to MBD, but without the knowledge that comes from modeling component behavior and interconnection structure.

4.1 SFL

Returning to the example presented in the previous section, suppose we have the following O

$$\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{array}$$

The data in O can be compactly represented using four counters for each diagnosis. Let d be a diagnosis, S_F be the set of indices of non healthy components in d, e be the index of the error detection information:

- $a_{11}(d) = \sum_{n=1..N} [(\bigvee_{c_m \in S_F} o_{nm}) \wedge e_n]$
- $a_{10}(d) = \sum_{n=1..N} [(\bigvee_{c_m \in S_F} o_{nm}) \wedge \neg e_n]$
- $a_{01}(d) = \sum_{n=1..N} [(\bigwedge_{c_m \in S_F} \neg o_{nm}) \wedge e_n]$
- $a_{00}(d) = \sum_{n=1..N} [(\bigwedge_{c_m \in S_F} \neg o_{nm}) \wedge \neg e_n]$

where $[\cdot]$ is the Iverson's operator [5].

Diagnosis in SFL consists in identifying the diagnoses that are more probable explanations for the errors. This is done by means of *similarity coefficients* taken from data clustering techniques [6], which are defined using the four counters just defined. As an example, the Ochiai similarity coefficient is defined as follows

$$s(j) = \frac{a_{11}(j)}{\sqrt{(a_{11}(j) + a_{01}(j)) * (a_{11}(j) + a_{10}(j))}} \tag{1}$$

For the three single faults it follows

	a_{01}	a_{10}	a_{11}	s
$\neg h_1$	0	1	1	0.7
$\neg h_2$	1	1	0	0
$\neg h_3$	0	0	1	1

Compared to MBD, the second conflict is missing. Including this information would yield (3^{rd} row)

$$\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array} \quad obs_3 \text{ (c_2 and c_3 are involved, leading to a fail)}$$

For the three single faults it would follow

	a_{01}	a_{10}	a_{11}	s
$\neg h_1$	1	1	1	0.5
$\neg h_2$	1	1	1	0.5
$\neg h_3$	0	0	2	1

For double faults it follows

	a_{01}	a_{10}	a_{11}	s
$\neg h_1 \wedge \neg h_2$	0	1	2	0.82
$\neg h_2 \wedge \neg h_3$	0	1	2	0.82
$\neg h_1 \wedge \neg h_3$	0	1	2	0.82

Note that this way of counting multiple diagnoses start converging to a value that is merely dependent on the pass/fail ratio of the matrix as typically all component combinations will be involved in any computation (the triple fault $\neg h_1 \wedge \neg h_2 \wedge \neg h_3$ is involved in all - for the example above $s(\neg h_1 \wedge \neg h_2 \wedge \neg h_3) = 0.82$. Here, SFL does not provide a sound basis for multiple fault diagnosis.

4.2 Logic Reasoning

In this section we describe the logic reasoning (LR) method to compute diagnoses and present several approaches to compute the associated probabilities.

4.2.1 Compute Diagnoses

Unlike the MBD approach mentioned earlier, which statically deduces information from the program source, O is the *only*, dynamic source of information, from which *both* a model, and the input-output observations are derived. Apart from exploiting dynamic information, this approach only requires a generic component model, avoiding the need for detailed functional modeling or relying, e.g., on invariants or pragmas for model information. Note, however, that this default model can easily be extended when more detailed information is available.

Abstracting from particular component behavior, each component c_j is modeled by the weak model

$$h_j \Rightarrow (x_j \Rightarrow y_j)$$

where h_j models the health state of c_j and x_j, y_j model its input and output variable value *correctness* (i.e., we abstract from actual variable *values*, in contrast to the earlier example). This weak model implies that a healthy component c_j translates a correct input x_j to a correct output y_j. However, a faulty component or input *may* lead to an erroneous output.

As each row in O specifies which components were involved, we interpret a row as a "run-time" model of the program as far as it was considered in that particular run. Consequently, O is interpreted as a sequence of typically different models of the program, each with its particular observation of input/output correctness. The overall diagnosis can be viewed as a sequential diagnosis approach that incrementally takes into account new structural program (and pass/fail) evidence with increasing N. A single row $O_{n,*}$ corresponds to the (sub)model

$$\begin{aligned} &h_m \Rightarrow (x_m \Rightarrow y_m), \text{ for } m \in I_n \\ &x_{s_i} = y_{s_{i-1}}, \text{ for } i \geq 2 \\ &x_{s_1} = \texttt{true} \\ &y_{s'} = \neg e_n \end{aligned}$$

where $I_n = \{m \in \{1, \ldots, M\} \mid o_{nm} = 1\}$ denotes the well-ordered set of component indices involved in computation n, s_i denotes the i^{th} element in this ordering, (i.e., for $i \leq j, s_i \leq s_j$), s' denotes its last element. The resulting component chain logically reduces to

$$\bigwedge_{m \in S_n} h_m \Rightarrow \neg e_n$$

For example, consider the row ($M = 5$)

c_1	c_2	c_3	c_4	c_5	e
1	0	0	1	0	1

This corresponds to a model where components c_1, c_4 are involved. As the order of the component invocation is not given (and with respect to our above weak component model is irrelevant), we derive the model

$$\begin{aligned} &h_1 \Rightarrow (x_1 \Rightarrow y_1) \\ &h_4 \Rightarrow (x_4 \Rightarrow y_4) \\ &x_4 = y_1 \\ &x_1 = \texttt{true} \\ &y_4 = \neg e_n \end{aligned}$$

In this chain the first component c_1 is assumed to have correct input ($x_1 = \texttt{true}$, typical of a proper test), its output feeds to the input of the next component c_4 ($x_4 = y_1$), whose output is measured in terms of e_n ($y_4 = \neg e_n$). This chain logically reduces to

$$h_1 \wedge h_4 \Rightarrow \texttt{false}$$

If this were a passing computation ($h_1 \wedge h_4 \Rightarrow \texttt{true}$) we could not infer anything (apart from the exoneration when it comes to probabilistically rank the diagnosis candidates as explained in next section). However, as this run failed this yields

$$\neg h_1 \vee \neg h_4$$

which, in fact, is a conflict. In summary, each failing run in O generates a conflict according to

$$\bigvee_{m \in S_n} \neg h_m$$

As in the former MBD approach, the conflicts are then subject to a hitting set algorithm that generates the diagnostic candidates.

To illustrate this concept, again consider the example program. For the purpose of the spectral approach we assume the program to be run two times where the first time we consider the correctness of y_1 and the second time y_2. This yields the observation matrix O below

c_1	c_2	c_3	e	
1	1	0	0	obs_1
1	0	1	1	obs_2

From obs_2, it follows

$$\neg h_1 \vee \neg h_3$$

which equals the first conflict from the earlier MBD approach, and the diagnosis trivially comprises the two single faults $\{1\}$ ($\neg h_1$) and $\{3\}$ ($\neg h_3$). Compared to the earlier MBD approach, the second conflict ($\neg h_2 \vee \neg h_3$) is missing due to the fact that no additional knowledge is available on component behavior and component interconnection. Although this would suggest that the dynamic approach yields lower diagnostic performance than the earlier MBD approach, note that the example program is ideally suited to static analysis, whereas real programs feature extensive control flow, rendering the previous approach extremely difficult. However, if, for some reason, we were able to capture the second conflict in terms of the execution trace according to

c_1	c_2	c_3	e	
0	1	1	1	obs_3

then our observation-based approach would yield exactly the same set of minimal diagnoses.

4.2.2 Classical Model for Computing Probabilities

Computing probabilities is done in much the same way as in MBD. For every diagnosis candidate, we update the posteriors by the extent that the observation is explained by the candidate diagnosis. In contrast to the MBD case, an observation is not an input or output value, but pass or fail information e_m (as the input and outputs are already taken into account by e).

Suppose the following two observations

1	1	0	0	obs_1
1	0	1	1	obs_2

After obs_1, all diagnoses are still possible (8 in total)

$$\begin{aligned} d_1 &= (h_1 \wedge h_2 \wedge h_3) \\ d_2 &= (h_1 \wedge h_2 \wedge \neg h_3) \\ &\dots \\ d_7 &= (\neg h_1 \wedge \neg h_2 \wedge h_3) \\ d_8 &= (\neg h_1 \wedge \neg h_2 \wedge \neg h_3) \end{aligned}$$

and their probabilities are updated according to Bayes' rule

$$\Pr(d_i|obs_1) = \alpha \cdot \Pr(obs_1|d_i) \cdot \Pr(d_i)$$

where ϵ is defined as follows

$$\epsilon = \begin{cases} \frac{E_P}{E_P+E_F} & \text{if run passed} \\ \frac{E_F}{E_P+E_F} & \text{if run failed} \end{cases} \quad (2)$$

where $E_P = 2^M$ and $E_F = (2^l - 1) \cdot 2^{M-l}$ are the number of passed and failed observations that can be explained by diagnosis d_k, respectively, and $l = |S_N|$ is the number of faulty components in the diagnosis. Although this observation does not help much in pinpointing the fault (all diagnoses are still valid), its update makes single faults more probable than multiple faults.

As mentioned before, when considering obs_2, the minimal set of diagnoses is $\neg h_1$ or $\neg h_3$, and their probabilities are updated by

$$\Pr(\neg h_1|\{obs_1, obs_2\}) = \alpha \cdot \Pr(obs_2|\neg h_1) \cdot \Pr(obs_1|\neg h_1).p$$
$$\Pr(\neg h_3|\{obs_1, obs_2\}) = \alpha \cdot \Pr(obs_2|\neg h_3) \cdot \Pr(obs_1|\neg h_3).p$$

yielding

$$\Pr(\neg h_1|\{obs_1, obs_2\}) = \alpha \cdot \frac{4}{12} \cdot \frac{8}{12} \cdot p = 0.5$$
$$\Pr(\neg h_3|\{obs_1, obs_2\}) = \alpha \cdot \frac{4}{12} \cdot \frac{8}{12} \cdot p = 0.5$$

However, if we assume O also includes the second MBD conflict, i.e., obs_3, then the set of consistent diagnoses is the same as for MBD, i.e., $\neg h_3$ and $\neg h_1 \wedge \neg h_2$.

$$\Pr(\neg h_3|O) = \alpha \cdot \Pr(obs_3|\neg h_3) \cdot \Pr(obs_2|\neg h_3) \cdot \Pr(obs_1|\neg h_3).p$$
$$\Pr(\neg h_1 \wedge \neg h_2|obs_3) = \alpha \cdot \Pr(obs_3|\neg h_1 \wedge \neg h_2) \cdot \Pr(obs_2|\neg h_1 \wedge \neg h_2) \cdot \Pr(obs_1|\neg h_1 \wedge \neg h_2).p^2$$

Consequently

$$\Pr(\neg h_3|O) = \alpha \cdot \frac{4}{12} \cdot \frac{4}{12} \cdot \frac{8}{12} \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot \frac{6}{14} \cdot \frac{6}{14} \cdot \frac{8}{14} \cdot p^2$$

Note that when the two failed observations are available, the minimal diagnosis $\neg h_1$ is no longer a valid explanation.

4.2.3 Intermittency Model for Computing Probabilities

A disadvantage of the classical probability model is that components involved in passed runs are not exonerated, and there is not a way to distinguish between diagnoses with the same cardinality. An approach to account for the fact that, similar to SFL, components involved in passed computations should be exonerated, by extending the component model with an intermittent failure model, as introduced in MBD [2].

We include statistical information on the probability that a faulty component c will exhibit correct behavior (i.e., produce correct output). Let $g(c)$ denote this probability. In the following we will distinguish three different Bayesian update schemes (ϵ), which we refer to as Method 1, Method 2, and Method 3.

4.2.4 Method 1

In this method, the observations made during passed runs are also taken into account by extending the ϵ definition as follows

$$\Pr(obs|D) = \begin{cases} 0 & \text{if } d \text{ and } obs \text{ are inconsistent} \\ 1 & \text{if } d \text{ logically follows from } obs \\ 1 & \text{if neither holds, run passed, and } a_{10}(d) = 0 \\ g(d) & \text{if none of the above and run passed} \\ 1 - g(d) & \text{if none of the above and run failed} \end{cases}$$

where $g(d) = \frac{a_{10}(d)}{a_{10}(d)+a_{11}(d)}$ (i.e, the fraction of involvement of the faulty component(s) that did *not* lead to a failure).

Again, considering the following two observations

$$\begin{array}{ccc|cc} 1 & 1 & 0 & 0 & obs_1 \\ 1 & 0 & 1 & 1 & obs_2 \end{array}$$

The hitting set for the weak model is equal to $\neg h_1 \vee \neg h_3$. After obs_1, the probabilities of $\neg h_1$ and $\neg h_3$ are updated as follows

$$\begin{aligned} \text{Pr}(\neg h_1|obs_1) &= \alpha \cdot \text{Pr}(obs_1|\neg h_1) \cdot p \\ \text{Pr}(\neg h_3|obs_1) &= \alpha \cdot \text{Pr}(obs_1|\neg h_3) \cdot p \end{aligned}$$

where, from the definition, $\text{Pr}(obs_1|\neg h_1) = g(\neg h_1)$ and $\text{Pr}(obs_1|\neg h_3) = 1$. Thus

$$\begin{aligned} \text{Pr}(\neg h_1|obs_1) &= \alpha \cdot g(\neg h_1) \cdot p \\ \text{Pr}(\neg h_3|obs_1) &= \alpha \cdot 1 \cdot p \end{aligned}$$

Similarly, after obs_2 the probabilities are updated as follows

$$\begin{aligned} \text{Pr}(\neg h_1|\{obs_1, obs_2\}) &= \alpha \cdot \text{Pr}(obs_2|\neg h_1) \cdot \text{Pr}(obs_1|\neg h_1) \cdot p \\ \text{Pr}(\neg h_3|\{obs_1, obs_2\}) &= \alpha \cdot \text{Pr}(obs_2|\neg h_3) \cdot \text{Pr}(obs_1|\neg h_3) \cdot p^2 \end{aligned}$$

where, from the definition, $\text{Pr}(obs_2|D) = 1 - g(d)$, and $\text{Pr}(obs_2|D) = 1 - g(d)$ is as previously defined. Consequently

$$\begin{aligned} \text{Pr}(\neg h_1|\{obs_1, obs_2\}) &= \alpha \cdot (1 - g(\neg h_1)) \cdot g(\neg h_1) \cdot p \\ \text{Pr}(\neg h_3|\{obs_1, obs_2\}) &= \alpha \cdot (1 - g(\neg h_3)) \cdot 1 \cdot p \end{aligned}$$

Since $g(\neg h_1) = 0.5$ and $g(\neg h_3) = 0$

$$\begin{aligned} \text{Pr}(\neg h_1|\{obs_1, obs_2\}) &= \alpha \cdot 0.5 \cdot 0.5 \cdot p = \alpha \cdot 0.25 \cdot p \\ \text{Pr}(\neg h_3|\{obs_1, obs_2\}) &= \alpha \cdot 1 \cdot p = \alpha \cdot p \end{aligned}$$

yielding

$$\begin{aligned} \text{Pr}(\neg h_1|\{obs_1, obs_2\}) &= 0.2 \\ \text{Pr}(\neg h_3|\{obs_1, obs_2\}) &= 0.8 \end{aligned}$$

which means that $\neg h_3$ is more probable to be the diagnostic explanation as $\neg h_1$ is partially exonerated. Compared to LR without intermittency, this method distinguishes between the two diagnoses, whereas in the previous method $\neg h_3$ and $\neg h_1$ were considered equally likely equal for explaining the fault.

Again, when compared to MBD approach the second conflict ($\neg h_2 \vee \neg h_3$) is missing. However, as explained in the previous section, if obs_3 were available, this approach would result in the same diagnostic performance as MBD

$$\neg h_3 \vee (\neg h_1 \wedge \neg h_2)$$

The probabilities are calculated according to

$$\begin{aligned} \text{Pr}(\neg h_3|O) &= \alpha \cdot \text{Pr}(obs_3|\neg h_3) \cdot \text{Pr}(obs_2|\neg h_3) \cdot \text{Pr}(obs_1|\neg h_3) \cdot p \\ \text{Pr}(\neg h_1 \wedge \neg h_2|O) &= \alpha \cdot \text{Pr}(obs_3|\neg h_1 \wedge \neg h_2) \cdot \text{Pr}(obs_2|\neg h_1 \wedge \neg h_2) \cdot \text{Pr}(obs_1|\neg h_1 \wedge \neg h_2) \cdot p^2 \end{aligned}$$

(Note that the diagnosis $\neg h_1 \wedge \neg h_2$ was previously discarded because it was not a minimal diagnosis. However, have we not discarded non-minimal diagnoses, its probability would be updated as follows $\text{Pr}(\neg h_1 \wedge \neg h_2|\{obs_1, obs_2\}) = \alpha \cdot (1 - g(\neg h_1 \wedge \neg h_2)) \cdot g(\neg h_1 \wedge \neg h_2) \cdot p^2$). From the definition, it follows

$$\begin{aligned} \text{Pr}(obs_1|\neg h_3) &= 1 \\ \text{Pr}(obs_1|\neg h_1 \wedge \neg h_2) &= g(\neg h_1 \wedge \neg h_2) \\ \text{Pr}(obs_3|\neg h_3) &= \text{Pr}(obs_2|\neg h_3) = 1 - g(\neg h_3) \\ \text{Pr}(obs_3|\neg h_1 \wedge \neg h_2) &= \text{Pr}(obs_2|\neg h_1 \wedge \neg h_2) = 1 - g(\neg h_1 \wedge \neg h_2) \end{aligned}$$

Hence,

$$\Pr(\neg h_3|O) = \alpha \cdot (1 - g(\neg h_3)) \cdot (1 - g(\neg h_3)) \cdot 1 \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot (1 - g(\neg h_1 \wedge \neg h_2)) \cdot (1 - g(\neg h_1 \wedge \neg h_2)) \cdot g(\neg h_1 \wedge \neg h_2) \cdot 1 \cdot p^2$$

yielding

$$\Pr(\neg h_3|O) = \alpha \cdot 1^2 \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot (1 - 0.33)^2 \cdot 0.33 \cdot p^2$$

thus,

$$\Pr(\neg h_3|O) = \alpha \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot 0.15 \cdot p^2$$

Meaning that the $\neg h_3$ is more probable than $\neg h_1 \wedge \neg h_2$.

Generalizing, in terms of a_{11}, a_{10}, a_{01}, and a_{00}, the probability of diagnosis d after O is observed equals

$$\Pr(d|O) = \alpha \cdot g(d)^{a_{10}(d)} \cdot (1 - g(d))^{a_{11}(d)} \cdot \Pr(d)$$

4.2.5 Method 2

This method is essentially the same as Method 1, except that it also takes into account the number of faulty components involved in the observation (in contrast to Method 1) by taking

$$\Pr(obs|D) = \begin{cases} 0 & \text{if } d \text{ and } obs \text{ are inconsistent} \\ 1 & \text{if } d \text{ logically follows from } obs \\ 1 & \text{if neither holds, run passed, and } a_{10}(D) = 0 \\ g(d)^{ct} & \text{if none of the above and run passed} \\ 1 - g(d)^{ct} & \text{if none of the above and run failed} \end{cases}$$

where ct is the number of faulty components involved in the observation, and $g(d)$ is defined as in the previous section. The rationale is that if more faulty components are involved, it is more likely the run will fail.

For the two single fault diagnoses that follow from LR on observations obs_1 and obs_2 this method yields the same results as Method 1 (as $ct = 1$)

$$\Pr(\neg h_1|\{obs_1, obs_2\}) = 0.2$$
$$\Pr(\neg h_3|\{obs_1, obs_2\}) = 0.8$$

However, for multiple fault diagnoses this method may give different results. Suppose again the following O

1	1	0	0	obs_1
1	0	1	1	obs_2
0	1	1	1	obs_3

As mentioned before, the hitting set for the weak model equals $\neg h_3 \vee (\neg h_1 \wedge \neg h_3)$. It follows

$$\Pr(\neg h_3|O) = \alpha \cdot \Pr(obs_3|\neg h_3) \cdot \Pr(obs_2|\neg h_3) \cdot \Pr(obs_1|\neg h_3) \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot \Pr(obs_3|\neg h_1 \wedge \neg h_2) \cdot \Pr(obs_2|\neg h_1 \wedge \neg h_2) \cdot \Pr(obs_1|\neg h_1 \wedge \neg h_2) \cdot p^2$$

From the definition, it follows

$$\begin{aligned} \Pr(obs_1|\neg h_3) &= 1 \\ \Pr(obs_1|\neg h_1 \wedge \neg h_2) &= g(\neg h_1 \wedge \neg h_2)^2 \\ \Pr(obs_3|\neg h_3) &= \Pr(obs_2|\neg h_3) = 1 - g(\neg h_3)^1 \\ \Pr(obs_3|\neg h_1 \wedge \neg h_2) &= pr(obs_2|\neg h_1 \wedge \neg h_2) = 1 - g(\neg h_1 \wedge \neg h_2)^1 \end{aligned}$$

Hence,

$$\Pr(\neg h_3|O) = \alpha \cdot (1 - g(\neg h_3)) \cdot (1 - g(\neg h_3)) \cdot 1 \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot (1 - g(\neg h_1 \wedge \neg h_2)^1) \cdot (1 - g(\neg h_1 \wedge \neg h_2)^1) \cdot g(\neg h_1 \wedge \neg h_2)^2.p^2$$

Thus, by evaluating $g(d)$,

$$\Pr(\neg h_3|O) = \alpha \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot 0.049 \cdot p^2$$

Similarly to Method 1, this method considers the single explanation more probable than the double fault.

Generalizing, the probability of diagnosis d after O is observed is updated according to

$$\Pr(d|O) = \alpha \cdot \prod_{i \in \{1..|S_F|\}} (g(d)^i)^{pr(d,i)} \cdot (1 - g(d)^i)^{fr(d,i)} \cdot \Pr(d)$$

where pr and fr count the number of passed and failed runs where it was observed that i faulty components where involved, respectively, i.e.,

$$pr(d,i) = \sum_{m \in \{1..M\}} [|\{n|o_{mn} \wedge n \in S_F \wedge e_m\}| = i]$$
$$fr(d,i) = \sum_{m \in \{1..M\}} [|\{n|o_{mn} \wedge n \in S_F \wedge \neg e_m\}| = i]$$

where S_F is the set of indices of faulty components in d, and $[\cdot]$ is the Iverson's operator.

4.2.6 Method 3

In this variant, the updates are computed based on [3], where ϵ is defined as

$$\Pr(obs|D) = \begin{cases} 1 & \text{if } d \text{ and } obs \text{ are inconsistent} \\ 0 & \text{if } d \text{ logically follows from } obs \\ \epsilon = 1 - g(d) & \text{if neither holds for passed and failed runs} \end{cases}$$

where $g(d)$ is defined as in the previous methods. Therefore, in terms of a_{11}, a_{10}, a_{01}, and a_{00}, ϵ can be re-written as follows

$$\epsilon = \frac{a_{11}(D)}{a_{11}(D) + a_{10}(D)}$$

Consequently for

1	1	0	0	obs_1
1	0	1	1	obs_2

we obtain

$$\Pr(\neg h_1|\{obs_1, obs_2\}) = \alpha \cdot \Pr(\{obs_1, obs_2\}|\neg h_1) \cdot p$$
$$\Pr(\neg h_3|\{obs_1, obs_2\}) = \alpha \cdot \Pr(\{obs_1, obs_2\}|\neg h_3) \cdot p$$

where

$$\Pr(\{obs_1, obs_2\}|\neg h_1) = \left(\frac{a_{11}(\neg h_1)}{a_{11}(\neg h_1) + a_{10}(\neg h_1)}\right)^{a_{11}(\neg h_1)+a_{10}(\neg h_1)}$$
$$\Pr(\{obs_1, obs_2\}|\neg h_3) = \left(\frac{a_{11}(\neg h_3)}{a_{11}(\neg h_3) + a_{10}(\neg h_3)}\right)^{a_{11}(\neg h_1)+a_{10}(\neg h_1)}$$

resulting in,

$$\Pr(\neg h_1|\{obs_1, obs_2\}) = \alpha \cdot 0.5^2 \cdot p = 0.20$$
$$\Pr(\neg h_3|\{obs_1, obs_2\}) = \alpha \cdot 1^2 \cdot p = 0.80$$

If the second failed observation is considered, the approach diagnostic results equals to $\neg h_3 \vee (\neg h_1 \wedge \neg h_2)$, and the probabilities are updated according to:

$$\Pr(\neg h_3|O) = \alpha \cdot 1^2 \cdot p = \alpha \cdot p$$
$$\Pr(\neg h_1 \wedge \neg h_2|O) = \alpha \cdot 0.67^3 \cdot p^2 = \alpha \cdot 0.30 \cdot p^2$$

4.3 Summary

	Classical	Method 1	Method 2	Method 3
$\Pr(\neg h_1)$	0.5	0.2	0.2	0.2
$\Pr(\neg h_3)$	0.5	0.8	0.8	0.8

(a) After obs_1 and obs_2

	Classical	Method 1	Method 2	Method 3
$\Pr(\neg h_3)$	0.98	0.99	0.999	0.77
$\Pr(\neg h_1 \wedge \neg h_2)$	0.02	0.01	0.001	0.23

(b) After obs_1, obs_2, and obs_3

Figure 3. Probabilities updates

Let $\Pr(\neg h_m) = 0.01$, Figure 3 lists the probabilities resulting from the various ϵ policies for the diagnoses obtained after obs_1 and obs_2 only (Figure 3(a)) and after obs_3 (Figure 3(b)). In the first case, the classic policy cannot distinguish between c_1 and c_3 while the g policies exploit the additional information provided by the exonerating observation obs_1. When obs_3 is included c_1 is no longer a valid diagnosis by itself, and is eliminated from the (hitting) set of valid diagnosis candidates. Hence, all policies favor c_3 as most likely candidate, due to (1) the lower prior probability of the double fault (all policies) and (2) the exoneration by passed runs (methods 1, 2, and 3).

5 Analytic Model

In this section we derive a simple, approximate model to assess the influence of various parameters on the *wasted* debugging effort W. It is defined as the effort that is wasted on inspecting a component that was not faulty. In our computation of W we assume that after each inspection, the test set is rerun, possibly leading to a new ranking (without the most recently removed fault). For example, suppose a triple-fault program ($M = 6$, and c_1, c_2, and c_3 faulty) for which the following diagnosis $D = \{\{1, 2, 6\}, \{3, 4, 5\}\}$ is obtained. This diagnosis induces a wasted effort of $W = 33\%$ as c_6 in the first candidate is inspected in vain, as well as, on average two out of three inspections in the second candidate (in this example we assumed that rerunning the test set didn't change the second candidate). In contrast to related work, we measure W instead of effort so that the performance metric's scale is independent of the number of faults in the program.

The evaluated parameters are number of components M, number of test cases N, testing code coverage r, testing fault coverage g, and fault cardinality C. Consider the example O in Figure 4(a), with $M = 5$ components of which the first $C = 2$ components are faulty. As a faulty component can still produce correct behavior, and therefore not cause a run to fail, we use an extended encoding where '1' denotes a component that is involved, whereas '2' denotes a (faulty) component whose involvement actually produced a failure (and consequently a failing run).

c_1	c_2	c_3	c_4	c_5	e
1	0	1	0	1	0
0	2	1	0	0	1
0	2	1	1	0	1
1	1	1	1	0	0
2	1	0	1	0	1

(a) Example O

c_1	c_2	c_3	c_4	c_5	e
0	2	1	0	0	1
0	2	1	1	0	1
2	1	0	1	0	1

(b) O's failed runs only

Figure 4. Observation Matrix Example

In the following we focus on the hitting set since its constituents are primarily responsible for the asymptotic behavior of W. Although their individual ranking is influenced by component activity in passed runs, the hitting set itself is exclusively determined by the failing runs. Thus, we consider the sub-matrix shown in Figure 4(b).

From Figure 4(b) it can be seen that the first 2 columns together form a hitting set of cardinality 2 (which corresponds to our choice $C = 2$). This can be seen by the fact that in each row there is at least one set member involved, i.e., there is a so-called "chain" of c_1 and/or c_2 involvement that is "unbroken" from top row to bottom row.

While this chain exists by definition (given the fact that both are faulty there is always at least one of them involved in *every* failed run), other chains may also exist, and may cause W to increase. This occurs when those chains pertain to diagnostic candidates of equal or lower cardinality (B) than C. Generally, two types of chain can be distinguished: (1) chains (of cardinality $B < C$) within the faulty components set, called *internal* chains, and (2) chains (of cardinality $B \leq C$) completely outside the faulty components set, called *external* chains. In the above example after $N = 2$ (so considering only the first two failed runs), there is still one internal chain (corresponding to single fault c_2), and two external chains (corresponding to single fault $\{3\}$, and double fault $\{3, 4\}$). As their probability will be higher (due to the a priori probability computation) they will head the ranking. With respect to the internal fault this does not significantly influence W since this indicates a true faulty component (the real double fault $\{1, 2\}$ being subsumed by $\{2\}$). Consequently, there is no wasted debugging effort. With respect to $\{3\}$ however, this fault will induce wasted effort. After $N = 3$ both single faults has disappeared (both chain of '1's have been *broken* during the third failing run), while the double fault c_3, c_4 is still present. From the above example it follows that (1) W is primarily impacted by external chains, and (2) the probability of a B cardinality chain still "surviving" decreases with the number of failing runs. The latter is the reason why in the limit for $N \to \infty$ all external (and internal) chains will have disappeared, exposing the true fault as only diagnosis.

5.1 Number of Failing Runs

As the number of failing runs is key to the behavior of W in the following we first compute the fraction of failed runs f out of the total of N runs, given r and g. Consider C faulty components. Let f denote the probability of a run failing. A run passes when note of the C components induces a failure, i.e., does not generate a '2' in the matrix. Since the probability of the latter equals $1 - r \cdot (1 - g)$ and generating a '2' requires (1) being involved (probability r) and (2) producing a failure (probability $(1 - g)$), the probability of not generating a '2' in the matrix equals $(1 - r \cdot (1 - g))$. Consequently, the probability a run passes equals $(1 - r \cdot (1 - g))^C$, yielding

$$f = 1 - (1 - r \cdot (1 - g))^C$$

This implies that for high g (and/or low r) a very large number of runs N is required to generate a sufficient number $N_F = f \cdot N$ of failing runs in order to eliminate competing chains of equal of lower cardinality B. As r also affects the number of external chains which, however, is not affected by g, the effect of g can be seen orthogonal to r in that it only impacts the number of failed runs through f. Consequently, g and N are related in that a high g is compensated by a, possible huge, increase in N. In the sequel, we therefore only focus on the effect of r.

5.2 Behavior for Small Number of Runs

While for large N the determination of W depends on the probability that competing chains will have terminated, for small N a more simple derivation can be made. Consider the case of a single failing run ($N_F = f \cdot N = 1$). From the first (failing) row ($k = 1$) in the above example (Figure 4(b)) it can be seen that there are generally $r \cdot (M - C)$ external single-fault ($B = 1$) chains (c_3 and c_5) that induce wasted effort. As W denotes the ratio of wasted effort it follows

$$W = \frac{r \cdot (M - C)}{M} \tag{3}$$

which for large M approaches r. This is confirmed by the experiments discussed later.

After the second failed run ($k = 2$) the probability a $B = 1$ chain survives two failing runs equals r^2 (i.e., the probability of two '1's for a particular component). Consequently, the number of $B = 1$ chains equals $r^2 \cdot (M - C)$, which, in general, decreases negative-exponentially with the number of (failing) runs ($f \cdot N$). For $B = 2$ the situation is less restrictive as *any* combination of '1's of the first and second row qualifies as a double-fault chain. As on average there are $M' = \lfloor r \cdot (M - C) \rfloor$ '1's per row there are $\binom{M'}{2}$ double-faults.

After the third failing run ($k = 3$) the number of surviving $B = 1$ chains equals $r^3 \cdot (M - C)$, whereas the number of triple faults equals $\binom{M'}{3}$ As for sufficiently large M the higher-cardinality combinations outnumber the lower-cardinality combinations, W is dominated by the combinations that have the same cardinality as the fault cardinality C. Consequently, assuming $N_F \leq C$ it follows that the number of C-cardinality chains that compete with the actual C-cardinality diagnosis is approximated by $\binom{M'}{C}$. However, if there are more combinations than $M - C$ these combinations will overlap in terms of component indices. As W does not measure wasted effort on a component that was already previously inspected (and subsequently

removed from the next diagnosis), the average number of "effective" C-cardinality chains will never exceed $\frac{M}{C}$ (as there are C indices per candidate). Hence, the number of competing C-cardinality chains is approximated by $\texttt{min}\{\frac{M}{C}, \binom{M'}{C}\}$.

5.3 Behavior for Large Number of Runs

For large N_F the trend of W can also be approximated from the probability that competing chains will still have survived after N_F runs, which we derive as follows. Consider a B-cardinality external chain. At each row there is a probability that this chain does not survive. Similar to the derivation of f we consider the probability that *all* B components involved in the chain have a '0' entry, which would terminate that particular chain. This probability equals $(1-r)^B$. Hence, the probability that a B-cardinality chain does not break per run equals $1-(1-r)^B$. Consequently, the probability that a chain survives N_F failing runs equals

$$(1-(1-r)^B)^{N_F}$$

Similar to the derivation for small N_F, we only consider C-cardinality chains. The largest number of competing chains at the outset equals $\binom{M'}{C}$. As there always exists an N_F for which this number is less than $\frac{M}{C}$ (in the asymptotic case we consider only a few chains) the number of competing chains after N_F runs is given by

$$(1-(1-r)^C)^{N_F} \cdot \binom{M'}{C}$$

Consequently, W is approximated by

$$W \approx \frac{(1-(1-r)^C)^{N_F} \cdot \binom{M'}{C}}{M} \tag{4}$$

We observe a negative-exponential (geometric) trend with N_F (N) while C postpones that decay to larger N_F (N) as the term $1-(1-r)^C$ approaches unity for large C.

In the following we asymptotically approximate the number of failing test runs N_F needed for an optimal diagnosis (i.e., W approaches 0). Considering Eq. (4) a single diagnosis is approximately reached for

$$(1-(1-r)^C)^{N_F} \cdot \binom{M'}{C} = W \cdot M$$

which can be modeled as $(1-(1-r)^C)^{N_F} = K$. It follows $N_F = -\log K / \log 1-(1-r)^C$. Since for sufficiently large C the term $1-(1-r)^C$ approaches unity, and since $\log 1-\epsilon \approx -\epsilon$ it follows that $N_F \sim \log K/(1-r)^C$. As $(1-r) < 1$ it follows $N_F \sim \log K \cdot ((1-r)^{-1})^C$ of which the second term increases exponentially with C. Since $K = \binom{M'}{C}$ for large M this term also increases exponentially with C. However, as the term is included in a logarithm, the effect of this term is less than the previous.

6 An optimal similarity coefficient for single-faults

In this section we show how our above reasoning approach can be used to derive an optimal similarity coefficient for *single-fault* programs.

In the single-fault case we know that all failures relate to only one fault, which, by definition, is included in the minimal hitting set. Hence, any coefficient approach should consider the minimal hitting set only (i.e., only those c_j which consistently occur in failing runs). This implies that the optimal approach is to select only the failing runs and compute the similarity coefficient. Since for these components by definition $a_{01} = 0$, one only needs to consider a_{11} and a_{10}. This, in turn, implies that the ranking is only determined by the exonerating term a_{10}. Thus the ranking can be calculated as follows

$$sim(j) = \begin{cases} s(j) & \text{if } a_{01} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

In summary, once we only consider the components included in the hitting set, any of the coefficients that includes a_{10} in the denominator will produce the same, optimal ranking. Experiments using this "hitting set filter" combined with a simple similarity coefficient such as Tarantula indeed confirm that this approach leads to the best performance [8].

Note that the above filter is only optimal for programs that have only 1 fault as applying this filter to any multiple-fault program would be overly restrictive. It would fail to detect faults that are not always involved in failed runs. For example,

the diagnosis for the O in the beginning of Section 4.1 when using the filtering approach would yield $D = \{\{1\}\}$, entirely ignoring two of the three faults. Hence, instead of considering a single-fault hitting set filter, we modify this approach in order to also allow application to multiple-fault programs. Taking the Ochiai coefficient as (best) starting point (for $\kappa = 1$, Eq. 5 follows from Eq. 1 by squaring, and factoring out $a_{11}(j)$, none of which changes the ranking) and applying the above filtering approach, we derive the following similarity coefficient, coined Zoltar-S, according to

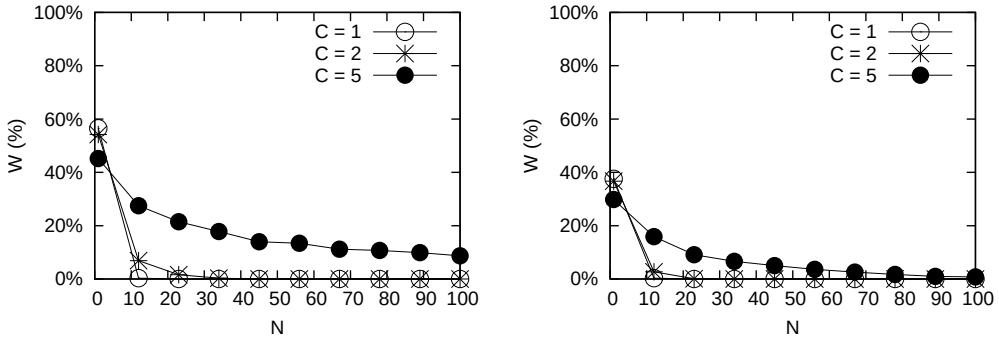
$$s_{\text{Z-S}} = \frac{a_{11}(j)}{a_{11}(j) + a_{10}(j) + a_{01}(j) + \kappa \cdot \frac{a_{01}(j) \cdot a_{10}(j)}{a_{11}(j)}} \tag{5}$$

where $\kappa > 0$ is a constant factor that exonerates a component c_j that was either seldom executed in failed runs or often in passed runs. We empirically verified that the higher the κ the more identical the diagnosis becomes with the one obtained by the hitting set filter [8]. In the context of this paper we limit κ to $10,000$ to avoid round-off errors.

A Synthetic Results

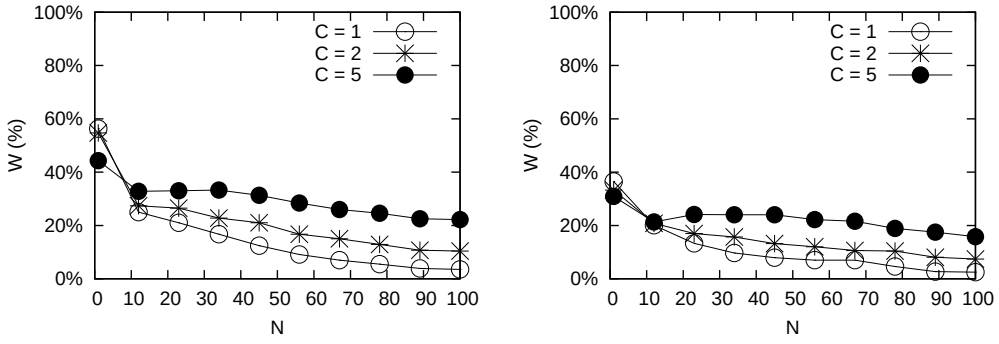
A.1 W vs. N

Figures 5, 6, and 7 plot W vs. N for several parameters, such as number of faults C, test set coverage r, and failure coverage g. To obtain the data, we use a simple, probabilistic model of program behavior that is directly based on C, N, M, r, and g. Without loss of generality we model the first C of the M components to be at fault.



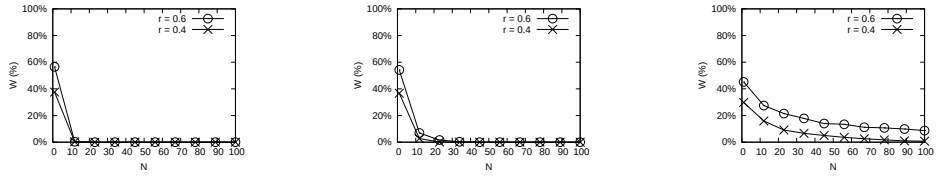
(a) $g = 0.1$ and $r = 0.6$ (b) $g = 0.1$ and $r = 0.4$

Figure 5. W **vs.** N **for** $g = 0.1$

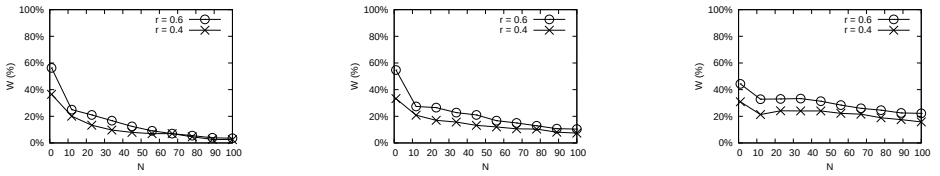


(a) $g = 0.9$ and $r = 0.6$ (b) $g = 0.9$ and $r = 0.4$

Figure 6. W **vs.** N **for** $g = 0.9$

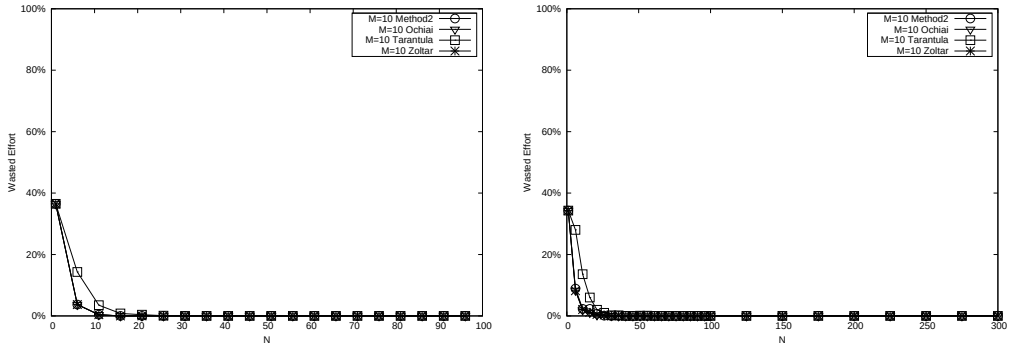


(a) $g = 0.1$ and $C = 1$ (b) $g = 0.1$ and $C = 2$ (c) $g = 0.1$ and $C = 5$



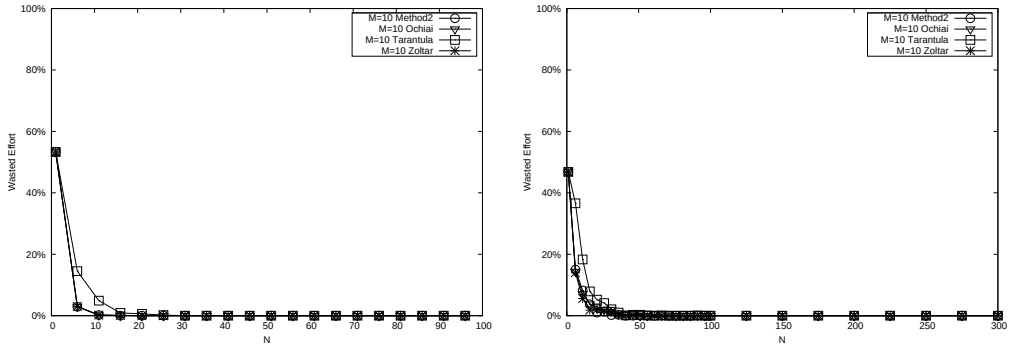
(d) $g = 0.9$ and $C = 1$ (e) $g = 0.9$ and $C = 2$ (f) $g = 0.9$ and $C = 5$

Figure 7. Impact of N on W



(a) $C = 1$ (b) $C = 2$

Figure 8. W vs. N for $M = 10$, $g = 0.1$, and $r = 0.4$

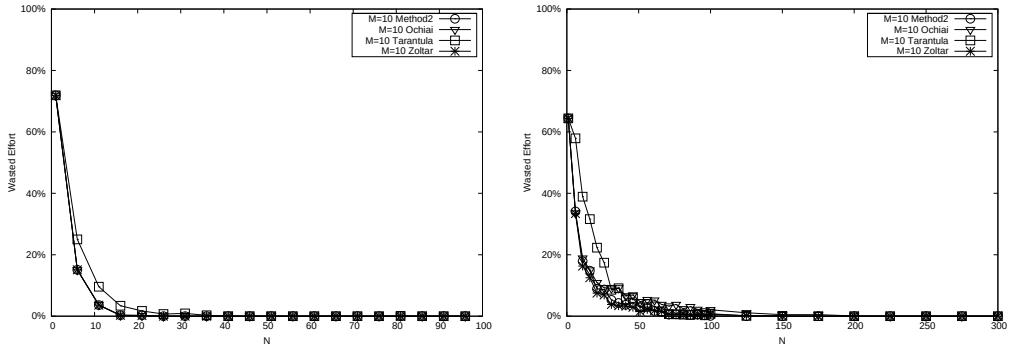


(a) $C = 1$ (b) $C = 2$

Figure 9. W vs. N for $M = 10$, $g = 0.1$, and $r = 0.6$

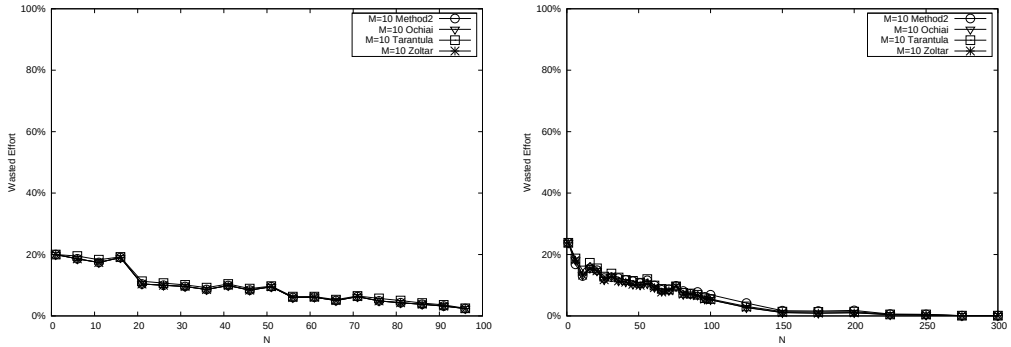
A.2 W vs. P

The following figures, up to Figure 24, plot W vs. P, showing that the observation-based technique (Zoltar-M using Method 2 as policy) may be of added value in order to employ several developers (P) to find the bugs. The plots were generated by fixing $M = 20$ and $N = 100$, and each point represents an average of 1,000 matrices.



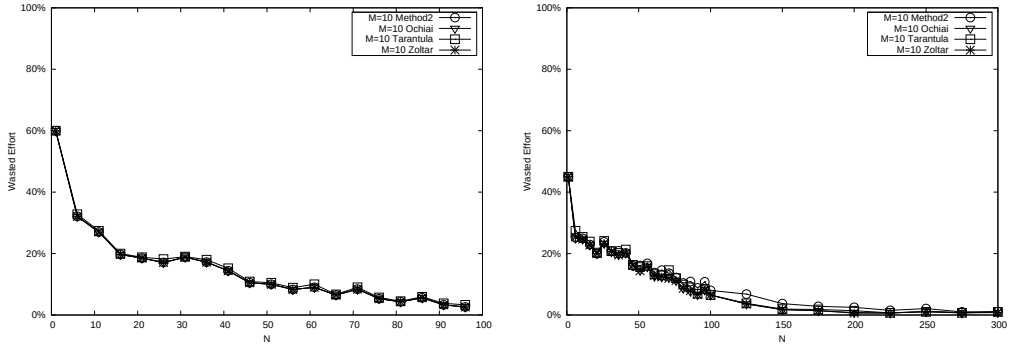
(a) $C = 1$ (b) $C = 2$

Figure 10. W **vs.** N **for** $M = 10$**,** $g = 0.1$**, and** $r = 0.8$



(a) $C = 1$ (b) $C = 2$

Figure 11. W **vs.** N **for** $M = 10$**,** $g = 0.9$**, and** $r = 0.4$

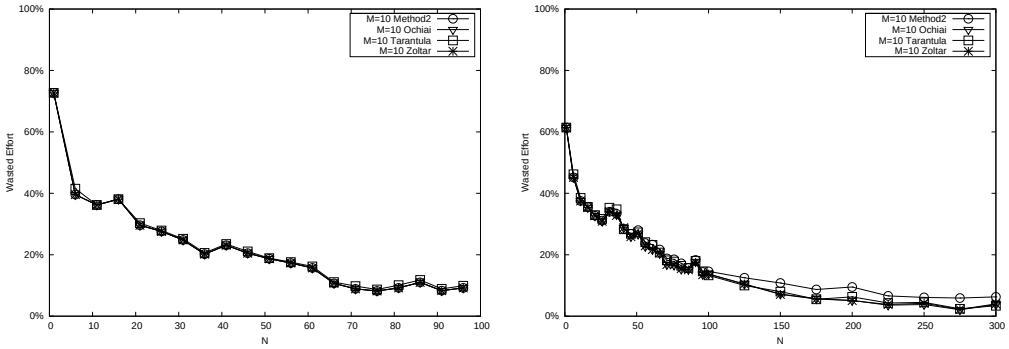


(a) $C = 1$ (b) $C = 2$

Figure 12. W **vs.** N **for** $M = 10$**,** $g = 0.9$**, and** $r = 0.6$

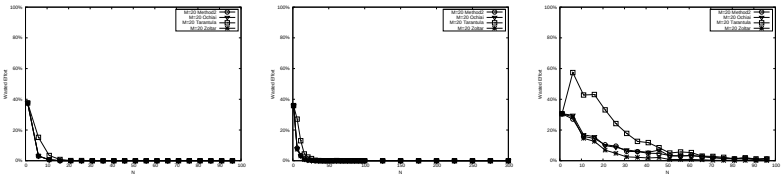
A.3 Probability/Similarity Distribution

The plots in Figure 24 contain the probability/similarity distribution for the rankings obtained with the several techniques. As can be seen, the observation-based approach (coined Zoltar-M) does give extra information on the number of faults in the code, when compared with SFL techniques (Ochiai, Tarantula, Zoltar-S).



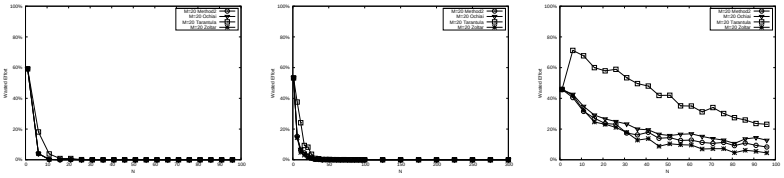
(a) $C = 1$ (b) $C = 2$

Figure 13. W **vs.** N **for** $M = 10$**,** $g = 0.9$**, and** $r = 0.8$



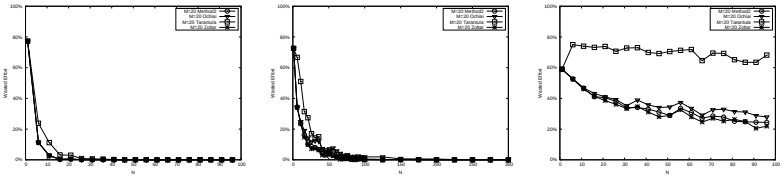
(a) $C = 1$ (b) $C = 2$ (c) $C = 5$

Figure 14. W **vs.** N **for** $M = 20$**,** $g = 0.1$**, and** $r = 0.4$



(a) $C = 1$ (b) $C = 2$ (c) $C = 5$

Figure 15. W **vs.** N **for** $M = 20$**,** $g = 0.1$**, and** $r = 0.6$



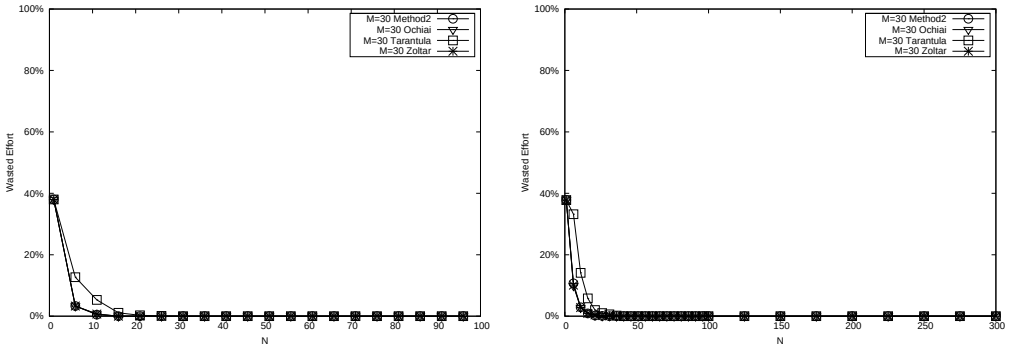
(a) $C = 1$ (b) $C = 2$ (c) $C = 5$

Figure 16. W **vs.** N **for** $M = 20$**,** $g = 0.1$**, and** $r = 0.8$

References

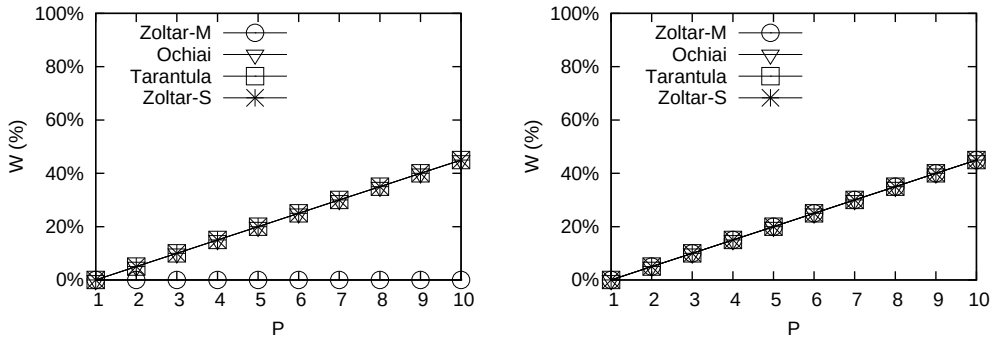
[1] J. de Kleer. Getting the probabilities right for measurement selection. In *Proceedings of the Seventeenth International Workshop on Principles of Diagnosis (DX-06), Burgos, Spain*, May 2006.

[2] J. de Kleer. Diagnosing intermittent faults. In *Proceedings of the Seventeenth International Workshop on Principles of Diagnosis (DX-07), Nashville, Tennessee, USA*, May 2007.



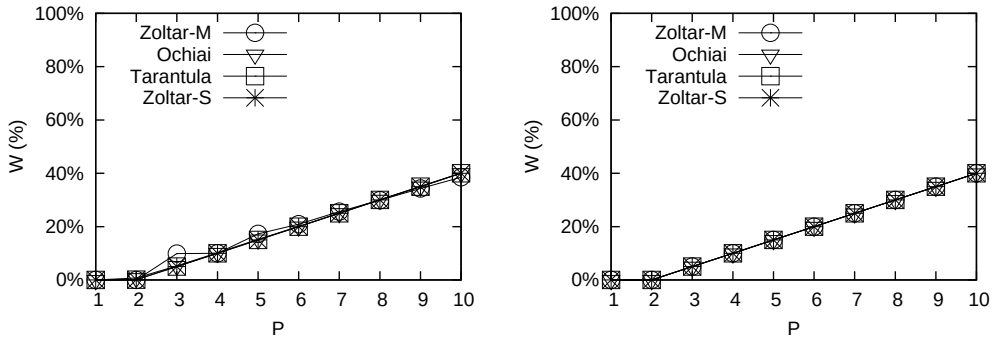
(a) $C = 1$ (b) $C = 2$

Figure 17. W **vs.** N **for** $M = 30$, $g = 0.1$, **and** $r = 0.4$



(a) $g = 0.1$, and $r = 0.6$ (b) $g = 0.1$ and $r = 0.4$

Figure 18. W **vs.** P **for** $C = 1$ **and** $g = 0.1$



(a) $g = 0.1$, and $r = 0.6$ (b) $g = 0.1$ and $r = 0.4$

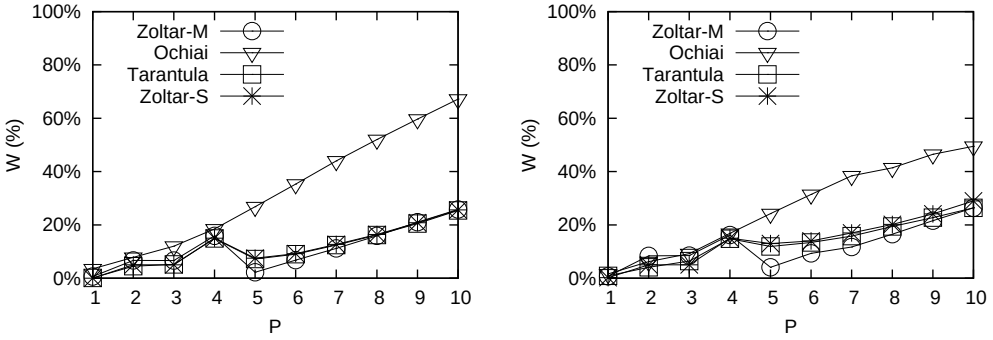
Figure 19. W **vs.** P **for** $C = 2$ **and** $g = 0.1$

[3] J. de Kleer. *Personal Communication*. TU Delft, Delft, June 2007.

[4] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artif. Intell.*, 32(1):97–130, 1987.

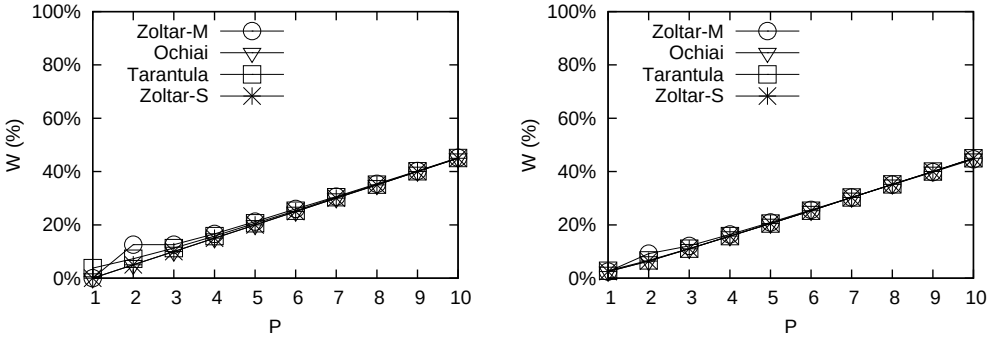
[5] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, Massachusetts, 1990.

[6] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.



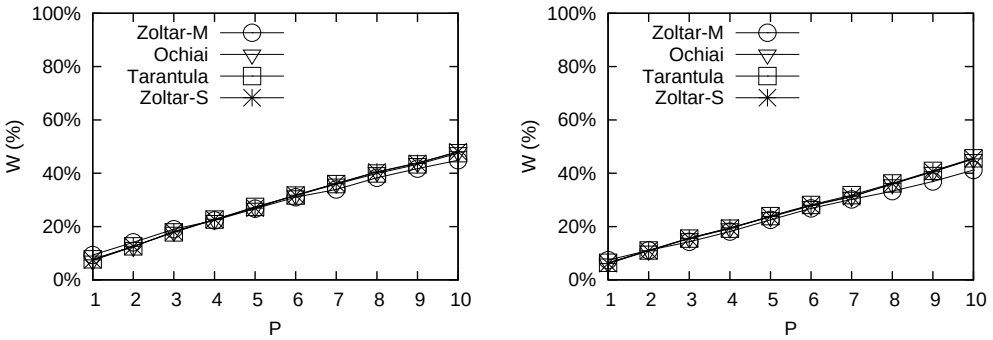
(a) $g = 0.1$, and $r = 0.6$ (b) $g = 0.1$ and $r = 0.4$

Figure 20. W **vs.** P **for** $C = 5$ **and** $g = 0.1$



(a) $g = 0.9$, and $r = 0.6$ (b) $g = 0.9$ and $r = 0.4$

Figure 21. W **vs.** P **for** $C = 1$ **and** $g = 0.9$

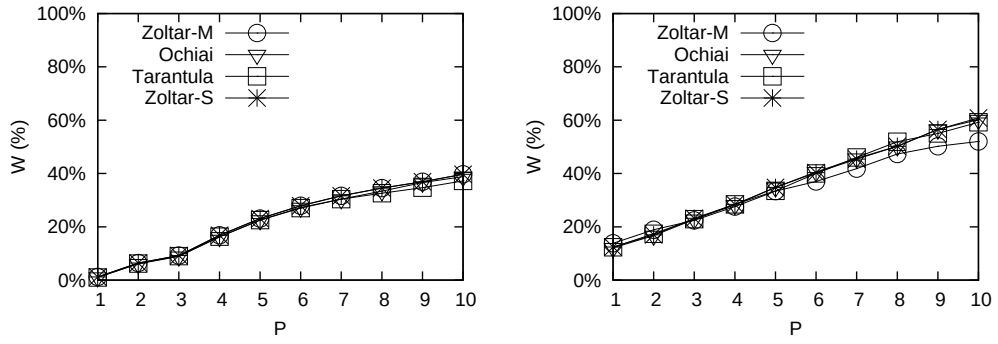


(a) $g = 0.9$, and $r = 0.6$ (b) $g = 0.9$ and $r = 0.4$

Figure 22. W **vs.** P **for** $C = 2$ **and** $g = 0.9$

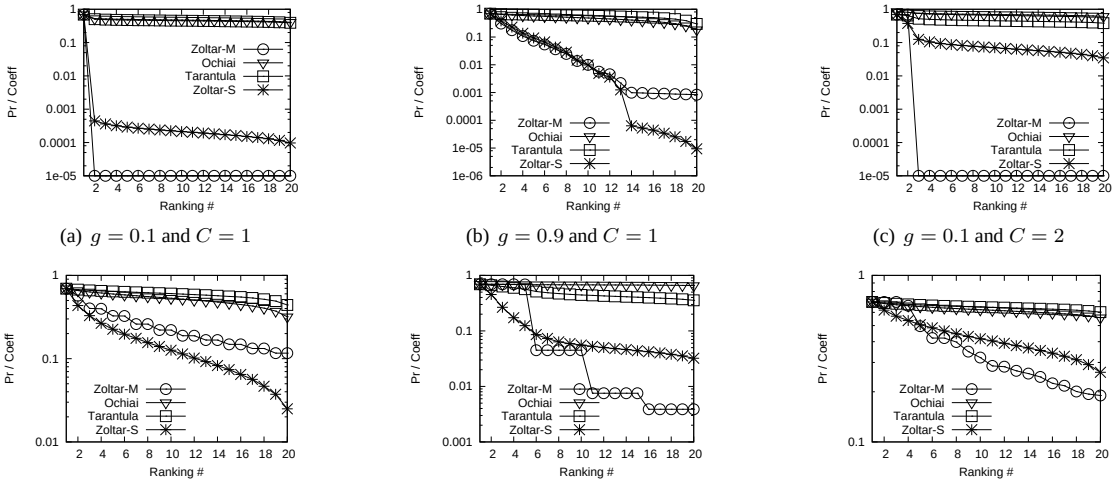
[7] R. Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[8] R. Vayani. Improving automatic software fault localization, July 2007. Master's thesis. Faculty of EEMCS, Delft University of Technology.



(a) $g = 0.9$, and $r = 0.6$ (b) $g = 0.9$ and $r = 0.4$

Figure 23. W vs. P for $C = 5$ and $g = 0.9$



(a) $g = 0.1$ and $C = 1$ (b) $g = 0.9$ and $C = 1$ (c) $g = 0.1$ and $C = 2$

(d) $g = 0.9$ and $C = 2$ (e) $g = 0.1$ and $C = 5$ (f) $g = 0.9$ and $C = 5$

Figure 24. Probability/Similarity distribution

TUD-SERG-2008-014
ISSN 1872-5392