ON CLASSIFICATION HEURISTICS OF PROBABILISTIC SYSTEM-LEVEL FAULT DIAGNOSTIC ALGORITHMS

Tamás Bartha, Endre Selényi
Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13–17, H-1111 Budapest, Hungary
bartha@sztaki.hu,selenyi@mit.bme.hu

Abstract System-level fault diagnosis of massively parallel computers requires efficient algorithms, handling a many processing elements in a heterogeneous environment. Probabilistic fault diagnosis is an approach to make the diagnostic problem both easier to solve and more generally applicable. The price to pay for these advantages is that the diagnostic result is no longer guaranteed to be correct and complete in every fault situation. In an earlier paper [2] the authors presented a novel methodology, called *local information diagnosis*, and applied it to create a family of probabilistic diagnostic algorithms. This paper examines the identification of fault-free and faulty units in detail by defining three heuristic methods of fault classification and comparing the diagnostic accuracy provided by these heuristics using measurement results.

Keywords: multiprocessor systems, system-level fault diagnosis, probabilistic algorithms

Introduction

Massively parallel computing systems are built up of a large amount of functionally identical *processing elements* (PEs). PEs execute the user application in a distributed manner, and cooperate using a communication medium to unify the partial results in complete solution. The probability of an error occurrence during application execution in an massively parallel system is significant due to the large number of components and the long continuous time of operation. Therefore, keeping the delivered system service uninterrupted by tolerating the effects of occurring errors is very important for parallel systems. This aim can be achieved by a fault tolerant architecture. *Automated fault diagnosis*

is an integral part of multiprocessor fault tolerance. Its task is to locate the faulty units in the system. Identified faulty units are stopped, and physically or logically excluded from the set of available resources, and the computer is reconfigured to use only the fault-free system devices.

Existing methods for system-level fault diagnosis can be categorized into *deterministic* and *probabilistic* methods. Deterministic diagnosis algorithms guarantee the correct and complete identification of the fault set, provided that certain a priori requirements on the structure of the test arrangement and the behavior of the faulty units are satisfied. These requirements are usually strict and often impractical. The resulting deterministic algorithms are too complex and not efficient enough to handle large systems. Probabilistic diagnostic algorithms only attempt to provide correct diagnosis with *high probability*. This implies that the created diagnostic image can be either *incorrect* (fault-free processors are misdiagnosed as faulty, or vice versa) or *incomplete* (the fault state of certain processors cannot be classified). The benefits of the probabilistic approach are simpler, faster algorithms, and no restrictive assumptions on the test arrangement or on the fault sets.

1. SYSTEM-LEVEL FAULT DIAGNOSIS

System-level fault diagnosis uses a simplified fault model. The system is built of a set of $u_i \in U$ units $(i = 1, 2, \ldots, n)$, connected by a set of $v_j \in V$ interconnection links $(j = 1, 2, \ldots, m)$. The units and links form a graph $S = (U, V)$. A unit u_i can either be *fault-free* (written as f_i^0) or *faulty* (f_i^1). It may test one or more other fault-free or faulty units. The complete collection of test assignments is a digraph $T = (U, E)$, where $E \subseteq V$ contains the set of $t_{ij} = (u_i, u_j)$ tests between units u_i and u_j. Two sets can be associated with each u_i unit: (1) the set of units tested by u_i, $\Gamma(u_i) = \{u_j | t_{ij} \in E\}$, and (2) the set of testers of u_i, $\Gamma^{-1}(u_i) = \{u_j | t_{ji} \in E\}$. The union of tested and tester units is the set of neighbors $N(u_i) = \Gamma(u_i) \cup \Gamma^{-1}(u_i)$. The set of units, that are reachable from u_i via directed edge sequences consisting of at most k edges are called the k-neighbors: $N_k(u_i)$. The cardinality of these sets are denoted by $\nu(u_i)$ and $\nu_k(u_i)$, respectively. Edges of the T digraph or *testing graph* are labeled by the $a_{ij} \in A$ test results. Tests have a binary (pass/fail) outcome. The A set of test results is called the *syndrome*.

The syndrome can be interpreted according to various test invalidation models. Test invalidation is the effect of the behaviour of a faulty unit on a test result. For example, a faulty tester unit may produce an nondeterministic pass/fail test result, independent on the state of the tested unit. This test invalidation scheme is called the *symmetric invalidation* or PMC model [5]. Other test invalidation schemes are also possible. In *heterogeneous* systems consisting of various functional units, test invalidation will likely be heterogeneous as well. The

generalized test invalidation scheme provides a unified framework to handle the differences of the invalidation models of system components [6]. The model is described in Table 1.1. Due to the complete test assumption fault-free units always test other units correctly. Test results of faulty tester unit can have three outcomes: always pass, always fail, or arbitrarily pass/fail independent on the fault state of the tested unit. These results correspond to the constants 0, 1, and X. Nine possible test invalidation models are encompassed by the generalized scheme, denoted by the respective C and D values. For example, symmetric invalidation is referred to as the $\mathrm{T_{XX}}$ test invalidation model.

Table 1.1 Generalized test invalidation

Tester unit	*Tested unit*	*Test result*
fault-free	fault-free	pass
fault-free	faulty	fail
faulty	fault-free	$C \in$ {pass, fail, or arbitrary}
faulty	faulty	$D \in$ {pass, fail, or arbitrary}

The relationship between tester and tested units encapsulated by generalized invalidation can be used to derive *parameterized one-step implication rules.* One-step implications have the form of "fault state a of unit u_i implies the fault state b of unit u_j" (denoted by $f_i^a \rightarrow f_j^b$). An implication rule is affected by three main parameters: (1) the test invalidation of the tester unit, (2) the (hypothesized) fault state of the tester/tested unit, and (3) the actual test outcome. Four types of one-step implication rules exist: tautology, forward implication, backward implication, and contradiction. A contradiction provides a sure implication: it expresses that either the fault-free or the faulty state of a certain unit is *incompatible* with the syndrome: $f_i^a \rightarrow f_i^{\bar{a}}$. The complete set of parameterized one-step implication rules derived from the general test invalidation model can be found in [2].

Two one-step implications can be combined into a two-step implication using the transitive property: if $f_i^a \rightarrow f_j^b$, and $f_j^b \rightarrow f_k^c$ are two valid one-step implications, then they imply $f_i^a \rightarrow f_k^c$. The set of all one-step and multiple-step implications obtained by repeated application of the transitive property is the *transitive closure.* It contains all information that can be extracted from the syndrome. In the following section we describe how the transitive closure can be utilized in the diagnostic procedure.

1.1 Local information diagnosis

The transitive closure is obtained using the implication rules derived from the generalized test invalidation model, and so it is the complete source of topology

and fault set independent diagnostic information. A diagnostic algorithm based on the transitive closure executes the following steps: first, one-step diagnostic implications are extracted using the parameterized implication rules and the actual syndrome. Then, multiple-step implications are obtained by transitively combining one-step implications. Inference propagation may continue until all possible implication chains are expanded in full length, that is, the transitive closure is created. All units involved in contradictions found in the transitive closure can be surely classified as fault-free or faulty. Finally, other units are diagnosed by a deterministic or probabilistic fault classification method.

There are two main performance bottlenecks in the above outlined procedure. First and foremost, generating the transitive closure of a large inference graph is a computation-intensive task. The underlying idea of *local information diagnosis* (LID) is that a probabilistic algorithm can achieve high probability of diagnostic correctness without expanding the implication chains in full length. Two main types of fault patterns can occur in a massively parallel system: (1) the faults are scattered throughout the system, separated from each other, and (2) the faults are located close to each other forming a group. In most practical cases both situations can be handled using just a portion of the diagnostic information [3]. The other performance bottleneck originates in the classification of those units which are not involved in a contradiction and whose fault state cannot be surely identified. Deterministic algorithms require complex methods for this task, since they must guarantee a correct and complete diagnosis (if only in a restricted set of cases). Probabilistic algorithms do not use the requirements necessary for correct operation of deterministic methods, and therefore can provide good diagnostic performance even beyond the traditional limits.

Along these guidelines we presented in an earlier paper [1] a family of probabilistic diagnostic algorithms based on the local information diagnosis methodology. These simple and efficient algorithms use the *generalized test invalidation* principle making them able to handle a class of heterogeneous systems. Here we outline the mechanism only of the *Limited Multiplication of Inference Matrix* (LMIM) algorithm, the interested reader can find the detailed definition of the other LID methods in [2]. In the initial phase of the LMIM algorithm one-step implications are collected and stored in the $2n \times 2n$ $\mathbf{M}$ *inference hypermatrix*. The $\mathbf{M}$ matrix consists of four $n \times n$ binary minor matrices: $\mathbf{M}^{00}$, $\mathbf{M}^{01}$, $\mathbf{M}^{10}$, and $\mathbf{M}^{11}$. The $\mathrm{m}^{xy}[i,j]$ element of the $\mathbf{M}^{xy}$ minor matrix ($x, y \in \{0,1\}$) equals to 1 if there exists an $f_i^x \rightarrow f_j^y$ one-step implication between units u_i and u_j, otherwise it is 0.

Transitive closure can be computed by the logical closure of the $\mathbf{M}$ matrix. This is achieved by the repeated application of the $\mathbf{M}^{(k+1)} \leftarrow \mathbf{M}^{(k)} \cdot \mathbf{M}^{(k)}$ iteration until no new implications appear in the matrix. In the LMIM algorithm the $\mathbf{M}$ matrix is multiplied only a few, constant times. Thus, the matrix will contain only a subset of the diagnostic inferences included in the transitive

closure. Nonzero elements in the main diagonal of the $\mathbf{M}^{01}$ and $\mathbf{M}^{10}$ minor matrices signify contradictions. For example, if $\mathrm{m}^{01}[i,i]$ equals to 1, then the $f_i^0 \rightarrow f_i^1$ implication holds, that is unit u_i is surely faulty. Similarly, all u_j units corresponding to the nonzero $\mathrm{m}^{01}[j,j]$ and $\mathrm{m}^{10}[j,j]$ elements can be surely classified. For other units a heuristic fault classification rule must be used to determine their fault state. The quality of the employed fault classification heuristic significantly affects diagnostic accuracy. Our previous paper used one of the possible heuristic rules. This paper introduces two additional fault classification heuristics, called *Election* and *Clique*, to the existing *Majority* heuristic described in [1]. The diagnostic performance of the three heuristics are compared using measurement results.

2. FAULT CLASSIFICATION HEURISTICS

The three fault classification heuristics called *Majority, Election,* and *Clique* presented in this section are all based on the assumption that the number of faulty units does not exceed the number of fault-free units in the system. However, each heuristic uses this assumption differently.

Majority heuristic

The idea of Majority heuristics is simple: since only the fault-free units produce reliable test results, only the implications from the fault-free states (stored in the $\mathbf{M}^{00}$ and $\mathbf{M}^{01}$ minor matrices) should be considered. The $f_j^0 \rightarrow f_i^0$ and $f_j^0 \rightarrow f_i^1$ implications ($j = 1, 2, \ldots, n$) can be interpreted as votes for the fault-free and faulty state of the u_i unit, respectively. The fault classification can be made as a majority decision between the votes for the fault-free/faulty state. The sum of votes, i.e., the sum of $f_j^0 \rightarrow f_i^0$ and $f_j^0 \rightarrow f_i^1$ implications can be calculated by counting the nonzero elements stored in the ith column of the $\mathbf{M}^{00}$ and $\mathbf{M}^{01}$ matrices (see Figure 1.1). Comparing the two sums $\Sigma^0[i] = \sum_j \mathrm{m}^{01}[j,i]$ and $\Sigma^1[i] = \sum_j \mathrm{m}^{00}[j,i]$, the unit is diagnosed as faulty if $\Sigma^0[i] < \Sigma^1[i]$, otherwise it is fault-free.

Election heuristic

The Election heuristic applies the mechanism of the CFT algorithm [2] to limited inference methods. The idea is to identify the faulty units sequentially one-by-one. Units are ranked according to the likelihood of them being faulty for the purpose of selection, and in each identification step the unit with the highest ranking is diagnosed as faulty. Then, the diagnostic uncertainty is decreased by removing the useless and confusing implications originating in the actually located faulty unit. Naturally, rankings must be recomputed each

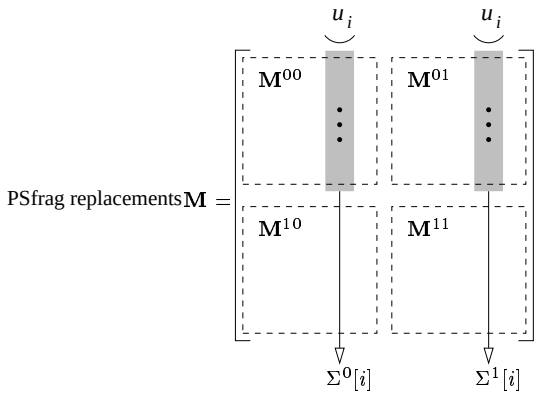

Figure 1.1 Calculation of the $\Sigma^0[i]$ and $\Sigma^1[i]$ values

Election heuristic
{ initialization }
for each $u_i \in U$ **do**
  $\mathrm{LF}[i] \leftarrow \Sigma^1[i] - \Sigma^0[i]$
  $\mathrm{NLF}[i] \leftarrow \sum_j \mathrm{LF}[j],\ u_j \in \Gamma^{-1}(u_i)$
end for
{ election }
$\Upsilon \leftarrow U, \Phi \leftarrow \emptyset$
while $\exists j, k, \mathrm{m}^{01}[j,k] \neq 0$ **do**
  find u_m with:
    maximum $\mathrm{LF}[m]$, and
    minimum $\mathrm{NLF}[m]$
  $\Phi \leftarrow \Phi \cup u_m$
  $\Upsilon \leftarrow \Upsilon - u_m$
  $\forall u_i \in U, \mathrm{m}^{01}[m,i] \leftarrow 0$
  recalculate $\mathrm{LF}[i]$ and $\mathrm{NLF}[i]$
end while { classification }
$\forall u_i \in \Phi$, u_i is faulty
$\forall u_j \in \Upsilon$, u_j is fault-free

Clique heuristic
{ initialization }
for each $u_i \in U$ **do**
  $\mathrm{C}^0[i] \leftarrow \mathrm{C}^0[i] \cup u_j$, if $\mathrm{m}^{00}[i,j] \neq 0$
  $\mathrm{C}^1[i] \leftarrow \mathrm{C}^1[i] \cup u_j$, if $\mathrm{m}^{01}[i,j] \neq 0$
end for
{ clique closure }
for each $u_i \in U$ **do**
  for each $u_j \in \mathrm{C}^0[i]$ **do**
    $\mathrm{C}^0[i] \leftarrow \mathrm{C}^0[i] \cup \mathrm{C}^0[j]$
    $\mathrm{C}^1[i] \leftarrow \mathrm{C}^1[i] \cup \mathrm{C}^1[j]$
  end for
end for
{ classification }
find u_m with:
  maximum $|\mathrm{C}^0[m]|$, and
  minimum $|\mathrm{C}^1[m]|$
$\forall u_i \in \mathrm{C}^0[m]$, u_i is fault-free
$\forall u_j \in \mathrm{C}^1[m]$, u_j is faulty
other units are unknown

Figure 1.2 Pseudo code of the Election and Clique heuristics

time the set of diagnostic implications is changed. The procedure is outlined in Figure 1.2.

The likelihood $\mathrm{LF}[i]$ of the faulty state of unit u_i is estimated as $\mathrm{LF}[i] = \Sigma^1[i] - \Sigma^0[i]$. For ranking units with identical LF values, the likelihood $\mathrm{NLF}[i]$ of the faulty state of units testing u_i is also counted: $\mathrm{NLF}[i] = \sum_j \mathrm{LF}[j]$ for

each $u_j \in \Gamma^{-1}(u_i)$. The units are sorted to find the unit u_m most likely to be faulty with the most reliable testers, i.e., having the maximum $\mathrm{LF}[m]$ and the minimum $\mathrm{NLF}[m]$ values. The u_m unit is then added to the Φ set of faulty units. The unit and its $f_m^0 \rightarrow f_i^1$ implications are removed from the $\mathbf{M}$ inference matrix, and the entire selection procedure starts again. When there are no more implications in the $\mathbf{M}^{01}$ minor matrix the remaining units are classified as fault-free.

Clique heuristic

The Clique heuristic is based on the diagnostic algorithm by Maestrini et al. [4]. The concept is similar to the Majority heuristic: if some fault-free units could be located, then their test results could reliably identify the fault state of other units. However, instead of comparing the feasibility of the fault-free/faulty states individually, the algorithm tries to group the units into two separate cliques. The *friendly* clique $\mathrm{C}^0[i]$ of unit u_i contains units with a fault state identical to u_i (they are either all fault-free or faulty), while the *foe* clique $\mathrm{C}^1[i]$ groups units with a fault state opposite to u_i (if u_i is fault-free, then they can only be faulty, and vice versa). Obviously, the clique sets of neighbor fault-free units are identical.

Cliques are initialized using the implications in the $\mathbf{M}^{00}$ and $\mathbf{M}^{01}$ minor matrices. Clique membership is then extended using the following two rules: (1) "my friend's friend is my friend", and (2) "my friends foe is my foe". The other two possible rules: (3) "my foe's friend is my foe", and (4) "my foe's foe is my friend" are not used, since they could lead to inconsistent cliques due to faulty units. Then the algorithm searches for the u_m unit with a maximum cardinality $\mathrm{C}^0[m]$ set and minimum cardinality $\mathrm{C}^1[m]$ set. The units belonging to the $\mathrm{C}^0[m]$ set are called the *Fault-Free Core*, they are classified as fault-free. Units in the $\mathrm{C}^1[m]$ set are diagnosed as faulty. Since some parts of the system can be separated by faulty units, there can be units neither contained in the $\mathrm{C}^1[m]$ set nor in the $\mathrm{C}^1[m]$ set. These units get the *unknown* classification, i.e., the Clique heuristic may lead to an *incomplete* diagnostic image.

3. MEASUREMENT RESULTS

The presented methods were compared using measurement results. For the purpose of measurement they were implemented in a dedicated simulation environment. The measurements examined many characteristics of the algorithms, including the effect of fault set size, fault groups, number of iterations, and system topology on diagnostic accuracy. The simulations were performed on a 2-dimensional toroidal mesh topology containing 12×12 processing elements. Random fault patterns of various size were injected and the system was diagnosed in 512 subsequent simulation rounds. Although several homogeneous

and heterogeneous invalidation schemes were involved in the simulation, here we can present only the results for the symmetric (PMC) test invalidation model due to volume constraints.

The effect of the fault set size on diagnosis accuracy is shown in Table 1.2. The first two columns contain the number of faults injected in the system and the percentage of faulty units. For the Majority and Election heuristics the number of simulation rounds with incorrect diagnosis (**MDR**), and the maximum number of incorrectly classified fault-free (**MGM**), and faulty (**MFM**) units per round are presented. According to the results the Majority heuristic gives a better overall performance than the Election heuristic, although the latter is less prone to misdiagnose a fault-free unit as a faulty unit. The Clique heuristic did not make any diagnostic mistakes, therefore the number of simulation rounds with incomplete diagnosis (**ICR**) and the maximum number of unknown units (**UM**) is given. Clearly, the number of unknown units considerably exceeds the total amount of units misdiagnosed by the other two methods, this is the price of the accurate diagnostic performance of the Clique heuristic.

Table 1.2 Diagnosis accuracy versus number of faults

Faults		**Majority**		**Election**		**Clique**	
		MDR	**MGM / MFM**	**MDR**	**MGM / MFM**	**ICR**	**UM**
4	(2.7%)	0	0 / 0	0	0 / 0	0	0
16	(11%)	0	0 / 0	0	0 / 0	10	1
36	(25%)	13	1 / 1	106	0 / 2	171	13
72	(50%)	222	5 / 6	463	0 / 8	510	58
96	(66%)	454	6 / 8	507	2 / 12	512	63

The higher degree of inference propagation (increasing the length of implication chains) improves diagnostic performance. Figure 1.3 presents this effect in the case of randomly injected fault patterns consisting of 36 and 72 faulty units. The number of simulation rounds with incorrect diagnosis is shown in the function of inference propagation iterations. Recall, that the length of implication chains doubles in each iteration, i.e., numbers 1, 2, 3, and 4 correspond to one-, two-, four-, and eight-step implications. The results justify our assumption: in the simulated system for random fault sets subsequent iterations improve diagnostic accuracy less and less.

We also examined the effect of system topology on diagnostic accuracy. Three regular communication topologies were simulated: (a) hexagonal toroidal grid with three connections, (b) 2-dimensional toroidal mesh with four connections, and (c) triangular toroidal grid with six connections. Figure 1.4 plots the number of simulation rounds with incorrect diagnosis in the function

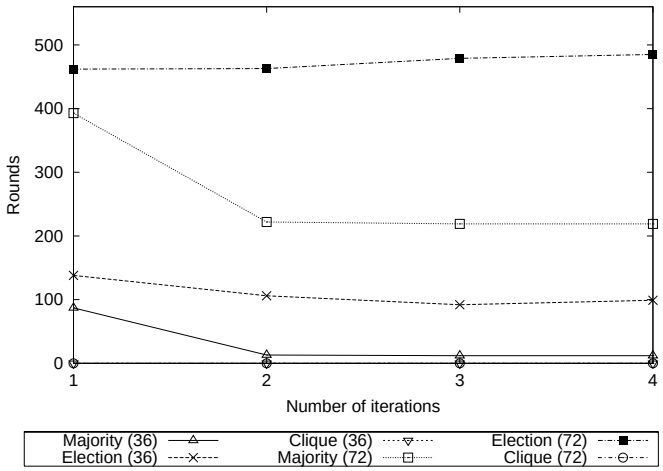

Figure 1.3 The effect of inference propagation

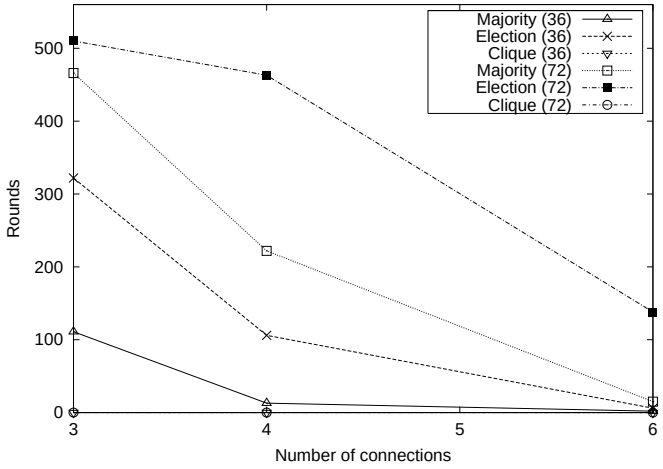

Figure 1.4 The effect of system topology

of connectivity. Random fault patterns consisting of 36 and 72 faulty units were injected in the system. As it can be seen, all of the heuristics perform better regardless of fault set size as the number of connections increases. In a completely connected system each heuristic would provide a correct and complete classification.

4. CONCLUSIONS

This paper described the concept of local information diagnosis, a novel approach to efficient probabilistic system-level diagnosis of massively parallel systems. The LID method uses a generalized test invalidation model to comply with heterogeneous structures and converts the syndrome into a topology and invalidation independent implication set. The paper demonstrated the legitimacy of the limited inference approach: it is possible to achieve high diagnostic accuracy even using only a portion of diagnostic information contained in the transitive closure by evaluating the implication chains in the inference graph only in a limited length.

Three different fault classification heuristics were presented. These heuristics apply ideas of existing successful algorithms in the LID framework. The main characteristics of the heuristics were compared by simulation and measurements. The Majority and Election heuristics have similar diagnostic performance and complexity. They provide a complete diagnostic image, but make a low amount of diagnostic mistakes in the case of large fault sets. The Clique heuristic produces correct diagnosis even for many faulty units, but as a disadvantage the more units remain unknown than are misdiagnosed by the other two methods.

References

[1] T. Bartha and E. Selényi. Efficient algorithms for system-level diagnosis of multiprocessors using local information. In *Proc. of the DAPSYS '96 Workshop on Distributed and Parallel Systems*, pages 183–190, Miskolc, October 1996.

[2] T. Bartha and E. Selényi. Probabilistic system-level fault diagnostic algorithms for multiprocessors. *Parallel Computing*, 22:1807–1821, 1997.

[3] D. Blough, G. Sullivan, and G. Masson. Fault diagnosis for sparsely interconnected multiprocessor systems. In *19th Int. IEEE Symp. on Fault-Tolerant Computing*, pages 62–69. IEEE Computer Society, 1989.

[4] P. Maestrini and P. Santi. Self diagnosis of processor arrays using a comparison model. In *Symposium on Reliable Distributed Systems (SRDS '95)*, pages 218–228, Los Alamitos, Ca., USA, September 1995. IEEE Computer Society Press.

[5] F. Preparata, G. Metze, and R. Chien. On the connection assignment problem of diagnosable systems. *IEEE Trans. Electronic Computers*, EC-16(6):848–854, December 1967.

[6] E. Selényi. *Generalization of System-Level Diagnosis.* D. Sc. thesis, Hungarian Academy of Sciences, Budapest, 1984.