

Probabilistic Diagnostics with P-Graphs*

Balázs Polgár[†] and Endre Selényi[†]

Abstract

This paper presents a novel approach for solving the probabilistic diagnosis problem in multiprocessor systems. The main idea of the algorithm is based on the reformulation of the diagnostic procedure as a P-graph model. The same, well-elaborated mathematical paradigm—originally used to model material flow—can be applied in our approach to model information flow. This idea is illustrated by deriving a maximum likelihood diagnostic decision procedure. The diagnostic accuracy of the solution is considered on the basis of simulation measurements, and a method of constructing a general framework for different aspects of a complex problem is demonstrated with the use of P-graph models.

Introduction

Diagnostics is one of the major tools for assuring the reliability of complex systems in information technology.

In such systems the test process is often implemented on system-level: the “intelligent” components of the system test their local environment and each other. The test results are collected, and based on this information the good or faulty state of each system-component is determined. This classification procedure is known as *diagnostic process*.

The early approaches that solve the diagnostic problem employed oversimplified binary fault models, could only describe homogeneous systems, and assumed the faults to be permanent. Since these conditions proved to be impractical, lately much effort has been put into extending the limitations of traditional models [1]. However, the presented solutions mostly concentrated on only one aspect of the problem. In this paper we introduce a novel modeling approach based on P-graphs that can integrate these extensions in one framework, while maintaining a good diagnostic performance. With this model, we formulate diagnosis as an optimization problem and apply the idea to the well-known multiprocessor testing problem, whose structure is one of the simplest.

*This research has been supported partly by the Hungarian National Research Foundation Grants OTKA T038027.

[†]Department of Measurement and Information Systems, Budapest University of Technology and Economics, Magyar Tudósok krt. 2, Budapest, Hungary, H-1117, e-mail: {polgar,selenyi}@mit.bme.hu

Furthermore, we have not only integrated existing solution methods, but proceeding from a more general base we have extended the set of solvable problems with new ones.

The paper is structured as follows. First an overview is given about the traditional aspects of system-level diagnosis and the way we have generalized the test invalidation model. Then the elements and the solution method of a P-graph model are introduced. In the main part the diagnostic problem of a multiprocessor system is formulated with the use of P-graphs. Afterwards, an important aspect, the extensibility of the model is demonstrated via examples. Moreover, the generation and the solution method of a P-graph model is clarified on a small example. The diagnostic accuracy of the decoding algorithm is presented on the basis of simulation results and it is compared to other approaches taken from the literature. Finally, we conclude and sketch the direction of future work.

1 System-level Diagnosis

System-level diagnosis considers the *replaceable units* of a system, and does not deal with the exact location of faults within these units. A *system* consists of an interconnected network of independent but cooperating *units* (typically processors). The state of each unit is either *good* when it behaves as specified, or *faulty*, otherwise. The *fault pattern* is the collection of the states of all units in the system. A unit may test the *neighboring* units connected with it via direct links. The network of the units testing each other determines the *test topology*. The outcome of a test can be either *passed* or *failed* (denoted by 0/1 or G/F); this result is considered *valid* if it corresponds to the actual physical state of the tested unit.

The collection of the results of every completed test is called the *syndrome*. The test topology and the syndrome are represented graphically by the *testing graph*. The vertices of a testing graph denote the units of the system, while the directed arcs represent the tests originated at the *tester* and directed towards the *tested unit* (UUT). The result of a test is shown as the label of the corresponding arc. Label 0 represents the passed test result, while label 1 represents the failed one. See Figure 1 for an example testing graph with three units.

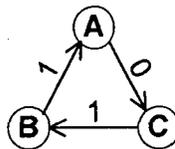


Figure 1: Example testing graph (test topology with syndrome)

1.1 Traditional approach

Traditional diagnostic algorithms [2, 3] assume that

- faults are permanent,
- states of units are binary (*good, faulty*),
- the test results of good units are always valid,
- the test results of faulty units can also be invalid. The behavior of faulty tester units is expressed in the form of *test invalidation models*.

Table 1 covers the possible test invalidation models where the selection of *c* and *d* values determines a specific model. The most widely used example is the so-called PMC (Preparata, Metze, Chien) test invalidation model, (*c = any, d = any*) which considers the test result of a faulty tester to be independent of the state of the tested unit. Another well-known test invalidation model is the BGM (Barsi, Grandoni, Maestrini) model (*c = any, d = faulty*) where a faulty tester will always detect the failure of the tested unit, as it is assumed that the probability of two units failing the same way is negligible.

Table 1: Traditional test invalidation models

State of tester	State of UUT	Test result
<i>good</i>	<i>good</i>	<i>passed</i>
<i>good</i>	<i>faulty</i>	<i>failed</i>
<i>faulty</i>	<i>good</i>	$c \in \{passed, failed, any\}$
<i>faulty</i>	<i>faulty</i>	$d \in \{passed, failed, any\}$

The purpose of system-level diagnostic algorithms is to determine the state of each unit from the syndrome. The difficulty comes from the possibility that a fault in the tester processor invalidates the test result. As a consequence, multiple “candidate” diagnoses can be compatible with the syndrome. To provide a complete diagnosis and to select from the candidate diagnoses, the so-called *deterministic* algorithms use extra information in addition to the syndrome, such as assumptions on the size or on the topology of the fault pattern.

Alternatively, *probabilistic* algorithms try to determine the most probable diagnosis assuming that a unit is more likely good than faulty [4]. Frequently, this maximum likelihood strategy can be expressed simply as “many faults occur less frequently than a few faults.” Thus, the aim of diagnostics is to determine the minimal set of faulty elements of the system that is consistent with the syndrome.

1.2 Generalized approach

In our previous work [5] we used a generalized test invalidation model, introduced by Blount [6]. In this model probabilities are assigned to both possible test outcome

for each combination of the states of tester and tested unit (shown in Table 2). Since the passed and failed results are complementary events, the sum of the probabilities in each row is 1. The assumption of the complete fault coverage can be relaxed in the generalized model by setting probability p_{b1} to the fault coverage of the test. Probabilities p_{c0} , p_{c1} , p_{d0} and p_{d1} express the distortion of the test results by a faulty tester. Moreover, the generalized model is able to encompass *false alarms* (a good tester finds a good unit to be faulty) by setting probability p_{a1} to nonzero.

Table 2: Generalized testing model

State of tester	State of UUT	Probability of test result	
		0	1
<i>good</i>	<i>good</i>	p_{a0}	p_{a1}
<i>good</i>	<i>faulty</i>	p_{b0}	p_{b1}
<i>faulty</i>	<i>good</i>	p_{c0}	p_{c1}
<i>faulty</i>	<i>faulty</i>	p_{d0}	p_{d1}

Naturally, the generalized test invalidation model also covers the traditional models. Setting the probabilities as $p_{a0} = p_{b1} = 1$, $p_{c0} = p_{c1} = p_{d0} = p_{d1} = 0.5$, and $p_{a1} = p_{b0} = 0$, the generalized model will have the characteristics of the PMC model, while the configuration $p_{a0} = p_{b1} = p_{d1} = 1$, $p_{c0} = p_{c1} = 0.5$ and $p_{a1} = p_{b0} = p_{d0} = 0$ will make it behave like the BGM model. Analogously, every traditional test invalidation model can be mapped as a special case to our model by assigning suitable probabilities to each element of the related test invalidation relation. In this sense the generalized test invalidation model covers the traditional models.

2 Diagnosis Based on P-Graphs

2.1 Definition of P-Graph Model of the Diagnostic System

The name 'P-graph' originates from the name 'Process-graph' from the field of Process Network Synthesis problems (PNS problem for short) in chemical engineering. In connection with this field the mathematical background of the solution methods of PNS problems have been elaborated well, see [7], [8] and [9].

A P-graph is a directed bipartite graph. Its vertices are partitioned into two sets, with no two vertices of the same set being adjacent. In our interpretation one of the sets contains *knowledge* (the knowledge about the states of units union the knowledge about the possible test results), the other one contains *logical relations* between the pieces of knowledge. The edges of the graph point from the *premisses*¹ 'through' the logical relation to the *consequences*. The set of premisses contains both good and faulty states of each unit (e.g., 'unit A is good', 'unit A is faulty', 'unit B is good', denoted by A_g, A_f, B_g), and the set of consequences contains the

¹premiss = preliminary condition

measured test results (e.g. 'unit A finds unit B to be good', 'unit B finds unit C to be faulty', denoted by AB_G , BC_F). Logical relations determine the possible premisses of each possible test result. Namely, there are 8 logical relations for each test according to the states of tester and tested unit and the possible test results. Probabilities in Table 2 are assigned to relations expressing the uncertainty of the consequences, see Figure 2.

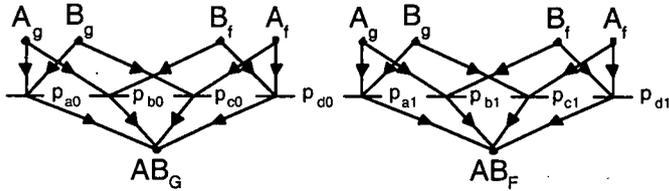


Figure 2: P-graph model of a single test (vertices with same label represent a single vertex; multiple instances are only for better arrangement)

A *solution structure* is defined as a subgraph of the original P-graph, which deduces the consequences back to a subset of premisses.

Function $X()$ is a membership function, $X(A)$ is 1 if unit A is in the solution structure, and 0 otherwise. With the use of this function *constraints* can be defined assuring that in a solution structure a unit should have one and only one state. Formally, for each unit U $X(U_G) + X(U_F) = 1$. A P-graph is *contradictionless* if all constraints are satisfied.

The *probability of the syndrome* (P_S) is the product of probabilities of relations in a solution structure. This is the occurring probability of the known consequences under the conditions of the given subset of system premisses.

Because of probabilities are assigned to relations, more contradictionless solution structures can exist having different subsets of system premisses and having different P_S values. The object is to find a solution structure containing that subset of system premisses which implies the known consequences with the maximum likelihood. This is an optimization task.

In principle, this task can be solved by general mathematical programming methods like mixed integer non-linear programming (MINLP), however, they are unnecessary complex. Friedler et al. ([7, 8, 9]) developed a new framework for solving PNS problems effectively by exploiting the special structure of the problem and the corresponding mathematical model.

2.2 Steps of the Solution Algorithm

1. The maximal P-graph structure is generated. It contains only the relevant pieces of knowledge and the relevant logical relations, but constraints are not yet satisfied. It contains all possible fault patterns being consistent with the given syndrome.

2. Every combinatorially feasible solution structure is obtained. These are the structures that satisfy the constraints and draw the known consequences—the syndrome—back to a subset of the system premisses. Each of these subsets determines a possible fault pattern.
3. For each combinatorially feasible solution structure the probability of syndrome is calculated. This is the conditional probability of the syndrome under the condition of a particular fault pattern.
4. The structure having the highest probability is selected; this solution structure contains the diagnosis with maximum likelihood.

Steps 2–4 can be completed either by a general solver for linear programming (since the generated maximal structure is a special flat P-graph), or with an adapted SSG algorithm [7] using the branch and bound technique.

3 Extensions of the Model

The main contribution of this novel modeling approach is its generality. With its use several aspects of system-level diagnosis can be handled in the same framework. Furthermore, it also became possible to formulate new aspects of diagnosis. So, it is possible to model and diagnose for instance

- *systems with heterogeneous elements*

To achieve this, different generalized test invalidation models with appropriate probabilities should be assigned to units with different behavior.

- *multiple fault states*

It is able to construct and handle a finer model of the state of a unit, than the binary one (containing the *good* and *faulty* states). This also means that the result of a test can be more than binary.

- *intermittent faults*

These are permanent faults that become activated only in special circumstances. Because these circumstances are usually independent from the testing process, these type of faults are diagnosed on the basis of multiple syndromes.

- *failures occurring during the test process*

It is a new aspect of the diagnostics. Traditional models all have the restrictive assumption that the state of units must be unchanged from the beginning of the test process to the end. But it is not acceptable if the time of test is comparable to the mean time between failures.

The model of the last two items are presented in details in the next subsections. The model constructed for the test process of intermittent faults is equivalent to the model of a system having more than two possible test results. Accordingly, for details of the handle of multiple fault states see the handle of intermittent faults.

3.1 Modeling Intermittent Faults

Although handle of intermittent faults is one of the difficult to manage diagnostic problems, a possible solution is the use of multiple syndromes, as mentioned above. In this approach two or more testing rounds are performed in a row, and the possible differences between the subsequent syndromes are used to detect intermittent faults.

The adaptation of diagnostic P-graph model to this approach is quite simple. Considering the case of double syndromes (for simplicity), there are four possible result combinations for each test:

- 'both results are passed' (denoted by GG),
- 'first result is passed, the other is failed' (denoted by GF),
- 'first result is failed, the other is passed' (denoted by FG) and
- 'both results are failed' (denoted by FF).

This means that the P-graph model of a single test should contain 4×4 logical relations according to the 4 possible test results and the 4 possible state-combinations of tester and tested unit (Figure 3). The probabilities of relations are calculated from the original probabilities (Table 2) and can be seen in Table 3.

Table 3: Probabilities of test results for pairs of syndrome

State of tester	State of UUT	Probability of test results			
		GG	GF	FG	FF
good	good	$p_{A0} = p_{a0}^2$	$p_{A1} = p_{a0}p_{a1}$	$p_{A2} = p_{a1}p_{a0}$	$p_{A3} = p_{a1}^2$
good	faulty	$p_{B0} = p_{b0}^2$	$p_{B1} = p_{b0}p_{b1}$	$p_{B2} = p_{b1}p_{b0}$	$p_{B3} = p_{b1}^2$
faulty	good	$p_{C0} = p_{c0}^2$	$p_{C1} = p_{c0}p_{c1}$	$p_{C2} = p_{c1}p_{c0}$	$p_{C3} = p_{c1}^2$
faulty	faulty	$p_{D0} = p_{d0}^2$	$p_{D1} = p_{d0}p_{d1}$	$p_{D2} = p_{d1}p_{d0}$	$p_{D3} = p_{d1}^2$

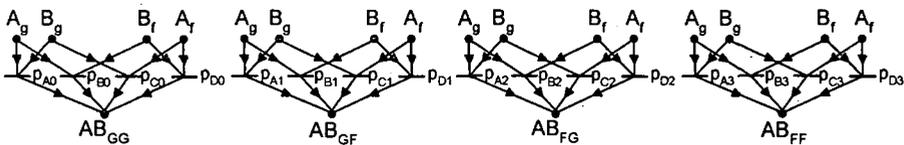


Figure 3: P-graph model of a single test in case of two syndromes

The case of diagnostics on the basis of more than two syndromes can be handled a similar way having more and more test result combinations.

A good property of this model is the following: after the 1st solution step—namely cutting the irrelevant parts of the graph to be solved—the P-graph model based on multiple syndromes is exactly of the same size as the P-graph model based on a single syndrome. This is because only the number of possible test results (or result combinations) grows, but the measured result (or result combination) of a test will be always a particular one.

3.2 Modeling Failures Occurring During the Test Process

Properties of the system to be modelled:

- faults are still permanent,
- units can fail during test process, i.e. a unit which was assumed to be good in a test can be faulty later. (Repairing is not included in the model, that is a faulty unit cannot become good in a later test.)

The second property implies that an order between tests should be defined and the states of a unit in different tests should be distinguished.

Let's define the *test order graph* $TO(V_{TO}, E_{TO})$, where

- each $t_i \in V_{TO}$ vertex represents a test in the system, i.e. it corresponds to an edge in the testing graph
- a $(t_i, t_j) \in E_{TO}$ directed edge defines a *preceding relation* between tests meaning that test t_i is performed earlier than test t_j .

For instance, consider a system with toroidal mesh topology, where each unit tests its four neighbors (Figure 4.a). The TO-graph of this system can be seen on Figure 4.b if only the order of those tests are known, which are performed by the same tester.

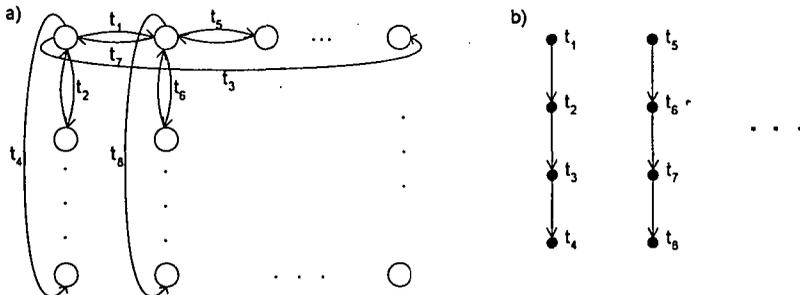


Figure 4: Example a) testing graph with toroidal mesh topology b) a possible test order graph of it

The definition of the P-graph model corresponds to the former one (Section 2.1) with the following changes.

- The set of *system premisses* contains each possible state of each unit in each such test, where the given unit is affected (for each U unit and t_i test U_iG and U_iF are included, where unit U is either a tester or a tested unit in test t_i).
- *Constraints* formulate that
 - each unit U in each test t_i where it is either a tester or a tested unit has one and only one state, i.e. $X(U_iG) + X(U_iF) = 1$.
 - for each unit U and tests t_i, t_j , where unit U is either a tester or a tested unit in tests t_i and t_j , and there exists a directed path from t_i to t_j in the TO-graph: $X(U_iF) + X(U_jG) \leq 1$.

Expectedly, the more information known about the dependencies of tests results the more accurate diagnosis. And reversely, the less edges in the test order graph can imply the more and more misdiagnosed processor in the diagnosis.

4 Example

Consider the testing graph and syndrome given on Figure 1. Eight logical relations belong to each of the three tests, but the maximal structure contains only four for each test depending on the test results as Figure 5 shows.

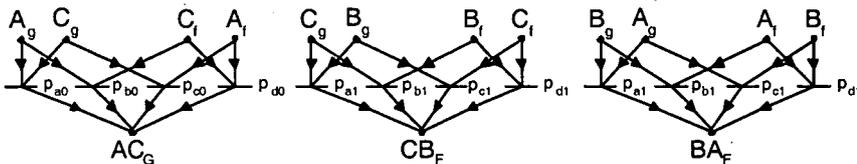


Figure 5: P-graph-model of testing graph and syndrome given on Figure 1

Eight combinatorially feasible solution structures exist because of the constraints and each of it contains three logical relations. The eight structures correspond to the 2^3 possible fault patterns of the three units. A part of these can be seen on Figure 6 with the corresponding diagnoses and probabilities. Finally, such a fault pattern is selected, which produces the syndrome with the highest probability.

Table 4 contains three test invalidation models, the first one corresponds to the PMC model, the second is a PMC model with incomplete fault coverage and the third is a more general model converging to the BGM model. The conditional probabilities of the syndrome under the conditions of different fault patterns, that is the redundant probabilities of the structures can be found in Table 5.

In case of PMC model the probability of syndrome is the highest when only unit B is faulty. This is still the case when the assumption of 100 percent test coverage

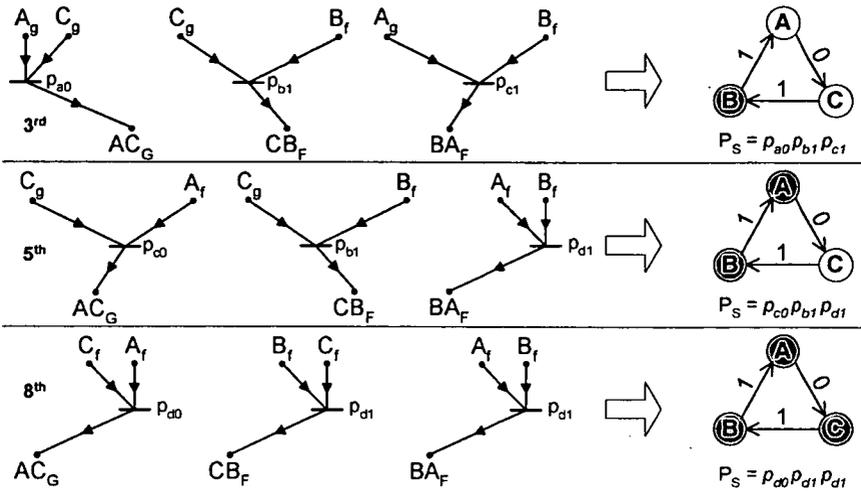


Figure 6: Some of solution structures of the P-graph model

Table 4: Test invalidation models with different probabilities

State of tester	State of tested unit	Test result					
		PMC		incomplete PMC		incomplete BGM-like	
		0	1	0	1	0	1
<i>good</i>	<i>good</i>	1	0	1	0	1	0
<i>good</i>	<i>faulty</i>	0	1	0.1	0.9	0.1	0.9
<i>faulty</i>	<i>good</i>	0.5	0.5	0.5	0.5	0.7	0.3
<i>faulty</i>	<i>faulty</i>	0.5	0.5	0.5	0.5	0.1	0.9

is given up but with smaller probability and with the possibility that unit C can be faulty although a good unit tested it to be good. If we assume that faults in the testers eventuate in valid test results more frequently than in invalid ones—as in the third model—then logically it seems to be probable that unit A is also faulty beside unit B and the algorithm provides this diagnosis.

Table 5: Probabilities of the syndrome (P_S) assuming different fault patterns and test invalidation models

Solution	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th
Faulty units		A	B	C	A B	A C	B C	A B C
PMC	0	0	0.5	0	0.25	0.25	0	0.125
inc. PMC	0	0	0.45	0	0.225	0.225	0.025	0.125
inc. BGM-like	0	0	0.27	0	0.567	0.027	0.027	0.081

5 Simulation Results

In order to measure the efficiency of the P-graph based modeling technique a simulation environment was developed, which generates the fault pattern and the corresponding syndrome for the most common topologies with various parameters. The P-graph model of the syndrome-decoding problem was solved as a linear programming task using a commercial program called CPLEX. Other diagnostic algorithms with different solution methods taken from the literature were also implemented for comparison. First, the accuracy of the developed algorithm is demonstrated for varying parameters, then its relation to other algorithms for fixed parameters.

The simulations were performed in a two-dimensional toroidal mesh topology, where each unit is tested by its four neighbors and each unit behaved according to the PMC test invalidation model. Statistical values were calculated on the basis of 100 diagnostic rounds. In every round the fault pattern was generated by setting each processor to be faulty with a given probability, independently from others.

Accuracy of the solution algorithm: measurements were performed with system sizes of 4×4 , 6×6 , 8×8 , 10×10 units, and the failure probability of units varied from 10% to 100% in 10% steps. From the diagrams in Figure 7 it can be observed that the algorithm has a very good diagnostic accuracy. Even if half of the units were faulty, the rate of rounds containing misdiagnosed units did not exceed 20 percent, and the rate of misdiagnosed units relative to the system size was under 1 percent.

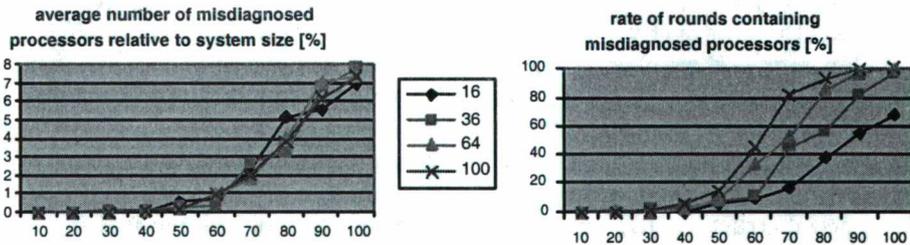


Figure 7: Simulation results depending on failing probability of units

Comparison to other algorithms: measurements were performed with system size 8×8 and the unit failure probability varied from 10% to 100% in 10% steps. The well-known algorithms taken from the literature were the LDA1 algorithm of Somani and Agarwal [10], the Dahbura, Sabnani and King (DSK) algorithm [11], and the limited multiplication of inference matrix (LMIM) algorithm developed by Bartha and Selenyi [12] from the area of local information diagnosis. It can be seen on the diagrams in Figure 8 that only the LMIM- algorithm approximates the accuracy of P-graph-algorithm.

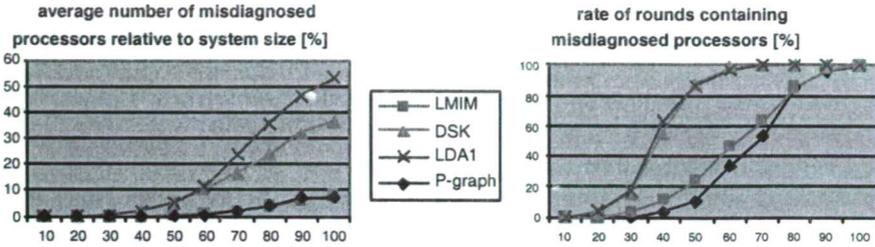


Figure 8: Comparison of probabilistic diagnostic algorithms

6 Conclusions

Application of P-graph based modeling in system-level diagnosis provides a general framework that supports the solution for several different fields, which previously needed several different modeling approaches and solution algorithms. Because the P-graph model takes into consideration more properties of the real system than previous models, its diagnostic accuracy is also better; it provides almost good diagnosis still in the situation, when half of the processors are faulty.

The results presented in this paper arose from solving the model with a general LP-problem solver and not from solving with a method specialized for PNS problems. Therefore its complexity was incomparable with traditional ones. But combinatorial approach for solving PNS problems is based on rigorous mathematical foundation, which –on the basis of experiences– can result in effective solution algorithm. Creating such an adapted algorithm is one of the subjects of our future work. Furthermore, we plan to examine the P-graph model of a diagnostic system with transient faults.

The favorable properties of the approach are achieved by considering the diagnostic system as a structured set of knowledge with well-defined relations. As mentioned previously, the syndrome-decoding problem in multiprocessor systems has a special structure, namely the direct manifestation of internal fault states in the syndromes. In more complex systems the states of the control logic have to be taken into account in the model to be analyzed [13]. These straightforward extensions to the modelling of integrated diagnostics can be well incorporated into the P-graph based models. Our current work aims at generalization of the results into this direction by extending previous results on the qualitative modeling of dependable systems with quantitative optimization [14].

References

- [1] T. Bartha, E. Selényi. Probabilistic System-Level Fault Diagnostic Algorithms for Multiprocessors, *Parallel Computing*, vol. 22, no. 13, pp. 1807–1821, Elsevier Science, 1997.

- [2] T. Bartha. Efficient System-Level Fault Diagnosis of Large Multiprocessor Systems, *Ph.D thesis*, BME-MIT, 2000.
- [3] M. Barborak, M. Malek, and A. Dahbura. The Consensus Problem in Fault Tolerant Computing, *ACM Computing Surveys*, vol. 25, pp. 171–220, June 1993.
- [4] S. N. Maheshwari, S. L. Hakimi. On Models for Diagnosable Systems and Probabilistic Fault Diagnosis. *IEEE Transactions on Computers*, vol. C-25, pp. 228–236, 1976.
- [5] B. Polgár, Sz. Nováki, A. Pataricza, F. Friedler. A Process-Graph Based Formulation of the Syndrome-Decoding Problem, In *4th Workshop on Design and Diagnostics of Electronic Circuits and Systems*, pp. 267–272, Hungary, 2001.
- [6] M. L. Blount. Probabilistic Treatment of Diagnosis in Digital Systems, In *Proc. of 7th IEEE International Symposium on Fault-Tolerant Computing (FTCS-7)*, pp. 72–77, June 1977.
- [7] F. Friedler, K. Tarjan, Y. W. Huang, L. T. Fan. Combinatorial Algorithms for Process Synthesis. *Comp. in Chemical Engineering*, vol. 16, pp. 313–320, 1992.
- [8] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan. Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems, *Chemical Engineering Science*, 47(8), pp. 1973–1988, 1992.
- [9] F. Friedler, L. T. Fan, and B. Imreh. Process Network Synthesis: Problem Definition. *Networks*, 28(2), pp. 119–124, 1998.
- [10] A. Somani, V. Agarwal. Distributed syndrome decoding for regular interconnected structures, In *19th IEEE International Symposium on Fault Tolerant Computing*, pp. 70–77, IEEE 1989.
- [11] A. Dahbura, K. Sabnani, and L. King. The Comparison Approach to Multiprocessor Fault Diagnosis, *IEEE Transactions on Computers*, vol. C-36, pp. 373–378, Mar. 1987.
- [12] T. Bartha, E. Selényi. Probabilistic Fault Diagnosis in Large, Heterogeneous Computing Systems, *Periodica Polytechnica*, vol. 43/2, pp. 127–149, 2000.
- [13] A. Pataricza. Algebraic Modelling of Diagnostic Problems in HW-SW Co-Design. In *Digest of Abstracts of the IEEE International Workshop on Embedded Fault-Tolerant Systems*. Dallas, Texas, Sept. 1996.
- [14] A. Pataricza. Semi-decisions in the validation of dependable systems In *Proc. of IEEE International Conference on Dependable Systems and Networks*, pp. 114–115, Göteborg, Sweden, July 2001.