System-Level Fault Diagnosis Using Comparison Models: An Artificial-Immune-Systems-Based Approach

Mourad Elhadef, Shantanu Das, and Amiya Nayak
School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada
Email: {*elhadef, shantdas, anayak*}*@site.uottawa.ca*

***Abstract*— The design of large dependable multiprocessor systems requires quick and precise mechanisms for detecting the faulty nodes. The problem of system-level fault diagnosis is computationally difficult and no efficient and generic deterministic solutions are known, motivating the use of heuristic algorithms. In this paper, we show how artificial immune systems (AIS) can be used for fault diagnosis in large multiprocessor systems containing several hundred nodes. We consider two models—the simple comparison model and the generalized comparison model (GCM), and we propose AIS-based algorithms for identifying faults in diagnosable systems, based on comparisons among units. We performed experimental analysis of these algorithms by simulating them on randomly generated diagnosable systems of various sizes under various fault scenarios. The simulation results indicate that the AIS-based approach provides an effective solution to the system-level fault diagnosis problem.**

***Index Terms*— Fault tolerance, System-level diagnosis, Multiprocessor and multicomputer systems, Comparison models, Artificial Immune Systems.**

I. Introduction

Large-scale self-diagnosable distributed systems are useful in providing dependable computing platforms for critical applications. These so-called loosely coupled multiprocessor systems are sometimes composed of hundreds (or even thousands) of interconnected processing units[1]. In order to detect faults at the processor level, a set of diagnostic tests are performed by the units. From the results of the tests, processors need to be diagnosed as faulty or fault-free. This problem is known as system-level fault diagnosis problem.

The system-level diagnosis problem has been extensively studied in the last three decades (the reader is referred to the following surveys for more details [1], [2]). The classical model, known as the PMC model, was introduced by Preparata, Metze, and Chien, in 1967. In [3], Preparata et al. studied the fault diagnosis in a directed-graph-based model in which processors are represented by the graph vertices and links along which tests can be conducted are represented by the graph edges. It was assumed that processors perform tests on each other and diagnosis is based on the collection of test outcomes. The PMC model and its variations are known as *invalidation models*. A second approach, known as the *comparison model*, has been introduced independently by Malek [4] and by Hakimi and Chwa [5] giving rise to two models. The Malek's model is known as the *asymmetric comparison* model and that of Hakimi and Chwa is called the *symmetric comparison* model. In comparison models, the system is modeled by an undirected graph, and it is assumed that pairs of processors are assigned the same job to be performed. The agreements and disagreements among the processors are the basis for identifying the set of faulty processors. In both models it is assumed that two fault-free processors give matching results while a faulty and a fault-free processor give mismatching outcomes. The two models differ in the assumption on tests involving a pair of faulty processors. In the symmetric model, both test outcomes are possible in this case, while in the asymmetric model two faulty processors always give mismatching outputs.

Maeng and Malek extended next the comparison model by allowing the comparisons to be conducted by the processors themselves [6]. Their extended model is known as the *Meang/Malek* (MM) model. Furthermore, in [6], it was assumed that a comparison is performed by each processor for each pair of distinct neighbors with which it can communicate directly; this special case of the MM model is referred to as the MM^* model. In [7], Sengupta and Dahbura presented a generalization of invalidation and comparison models by introducing a new model, known as the *generalized comparison model*, in which the comparator processor can be one of the two processors under comparison. Blough and Brown introduced next in [8] a combination of a distributed diagnosis and the generalized comparison model in systems having weak reliable broadcast capacity. They developed the first *broadcast comparison model*, in which two processors under comparison broadcast their outputs to all processors in the system. Recently, Chessa and Santi applied the comparison-based system-level fault diagnosis approach to ad-hoc networks [9].

Identifying the complete and correct set of faulty

This paper is based on "A Novel Artificial-Immune-Based Approach for System-Level Fault Diagnosis", by M. Elhadef, S. Das and A. Nayak, which appeared in the Proceedings of the First Int. Conference on Availability, Reliability and Security, (ARES'06), Austria, April 2006.

[1]We will use interchangeably the terms unit, node, processing unit, and processor.

processors using the comparison approach is shown to be NP-Hard [10], but if the system is t-diagnosable, the problem is solvable in polynomial time. This problem has been extensively studied leading to elegant and efficient solutions [1], [2]. In this paper, we consider a totally different approach based on Artificial Immune Systems (AIS) for solving the System-level diagnosis problem. We present AIS-based algorithms for identifying faults in all t-diagnosable multiprocessor and multicomputer systems based on the comparison approach. Artificial immune systems have been used in many applications, including fault diagnosis [11], [12], [13]. Our AIS-based diagnosis approach is shown to be efficient, in that, it does not suffer from a loss in diversity, and hence, allows for faster diagnosis in worst-case situations or when very large systems are considered.

The remainder of this paper is organized as follows. We first describe the comparison models for system-level diagnosis and related notations and definitions in Section II. Section III describes artificial immune systems (AIS) and then presents an analogy between AIS and the system-level diagnosis problem, based on which the diagnosis algorithm is designed. The algorithm is described in detail in Section IV followed by a discussion on the correctness and complexity of the algorithm, in Section V. Experimental results for the algorithms are presented and analyzed in Section VI and finally Section VIII concludes the discussion and motivates future investigations on the system-level fault diagnosis problem.

II. The Comparison Models

A. Basic Concepts

A comparison model for system-level fault diagnosis in multiprocessor systems can be described by two graphs, a *communication graph* and a *comparison* (or test) *graph*. The communication graph represents the interconnection topology of the multiprocessor system; an undirected edge $e = (u, v)$ represents a communication link between the two processors u and v. Whereas, the comparison graph shows the comparison tests that are performed in order to identify the set of faulty processors once a faulty situation is detected, i.e., when the system deviates from its expected behavior due to faults in the processors.

Faults can be classified either as *permanent* faults, *intermittent* faults, or *transient* faults. A transient fault occurs once and disappears. An intermittent fault is a transient one which occurs repeatedly, whereas a permanent fault continues to exit until the faulty unit is repaired. In this paper, we consider only permanent faults. If a faulty node is unable to communicate with the rest of the system, then the fault is known as *hard*, whereas if the faulty unit continues to operate and to communicate, with altered behaviors, with the other nodes in the system, then the fault is said to be *soft*. For a diagnosis to be possible, the behavior of soft faulty nodes should somewhat be constrained (or invalidated). Various comparison invalidation rules, which are used to diagnose the state of the units in the system, have been defined leading to different comparison models. In this paper, we consider two of these comparison models—the simple comparison model (SCM) as in [5] and the generalized comparison model (GCM) introduced by Sengupta and Dahbura [7]. The difference between the two models is that in GCM, the comparator processor is one of the processors being compared, whereas in SCM, all comparison tests are performed by a central observer that monitors the system. However, in both models, the diagnosis of faults based on the comparison outcomes, is performed by the central observer. We describe below both models.

Definition 1: A system is *t-diagnosable* if each node can be correctly identified as fault-free or faulty based on a valid collection of comparison results, assuming that the number of faulty nodes does not exceed a given bound t.

If the system deviates from its specified logical behavior, and if the faulty situation is detected, then the first step consists in diagnosing the state of the system, i.e., identifying which nodes are faulty and which are fault-free. The fault identification process is based on the comparison syndrome output by the system's nodes. A diagnosis is said to be *correct* if there are no fault-free units mistakenly diagnosed as faulty; otherwise, it is an *incorrect* diagnosis. A diagnosis is said to be *complete* if all faulty units are correctly identified; otherwise, the diagnosis is *incomplete*. In this paper, we consider only the deterministic diagnosis approach in which the input is a comparison syndrome and the output is the set of processors diagnosed as faulty.

B. The Simple Comparison Model (SCM)

In the simple comparison there is a central observer (comparator) which performs comparisons between pairs of processors by assigning them some tasks from the set of tasks $T = \{T_1, T_2, \ldots\}$. Each pair of processors v_i and v_j is assigned a task $T_l \in T$. Once the task T_l is completed by both processors, their results are compared. The comparison graph in this case, is an undirected graph $G = (V, C)$, where V denotes the set of processors and $C = \{(v_i, v_j) : (v_i, v_j)$ is a pair of processors performing the same task $T_l \in T\}$. We denote a processor pair (v_i, v_j) or (v_j, v_i) by c_{ij}.

The set of all comparison outcomes is called the *syndrome*. The notations used for the SCM is as follows:

- Γ_i denotes the set of processors with which a processor $v_i \in V$, is compared, and is given by $\Gamma_i = \{v_j : c_{ij} \in C\}$.
- Ω is a comparison syndrome.
- Ω^{ij} refers to the comparison outcome of the processor pair c_{ij}.
- $\Omega(v_i)$ defines the set of results of the comparison tests that are carried out between the processor v_i and all its neighbors, and is given by $\Omega(v_i) = \{(c_{ij}, \Omega_{ij}) : v_j \in \Gamma_i \;\&\; c_{ij} \in C\}$.
- $\widetilde{F}$ denotes the actual fault set in a faulty situation.
- $\Omega_{\widetilde{F}}$ refers to any comparison syndrome that can be generated under the fault set $\widetilde{F}$.

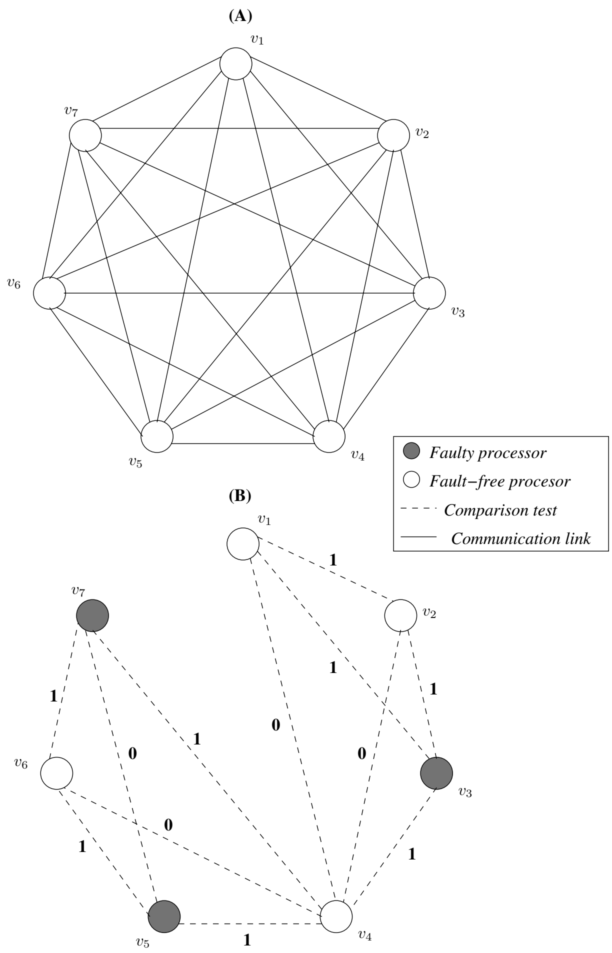

Figure 1. A 3-Diagnosable Comparison-Based System: (A) Communication Graph. (B) A Comparison Assignment and a Symmetric Comparison Syndrome.

Definition 2: A comparison syndrome is said to be *consistent* (or compatible) with a fault set F under SCM if for any $c_{ij} \in C$, such that v_i is fault-free, i.e., $v_i \in V - F$, $\Omega^{ij} = 1$ iff $v_j \in F$.

Figure 1 (A) shows an example of a small system composed of seven fully connected processors. In Figure 1 (B) a comparison assignment is shown in dashed lines, and a symmetric comparison syndrome corresponding to the actual fault set $\widetilde{F} = \{v_3, v_5, v_7\}$ is also provided. This example shows a 3-diagnosable system [14].

C. The Generalized Comparison Model (GCM)

The GCM is known to be the most general model, i.e., it generalizes both comparison models and invalidation models. Recall that under the invalidations models [3], [15], [16], units test each other directly, i.e., the comparator node is one of the nodes under comparison. Figure 2 depicts the invalidation rules of GCM. According to GCM, if the comparator node is fault-free, then the comparison outcome is 0 if none of the compared nodes is faulty, and it is 1 if one of them is faulty. However, if the comparator itself is faulty, then the comparison outcome is unreliable, and hence, may be 0 or 1. Two types of comparison models have been studied in the literature: symmetric and asymmetric comparisons. Symmetric and asymmetric comparison models differ in the assumption on comparisons involving a pair of faulty compared nodes, once the comparator node is non-faulty. In the symmetric model, both comparison outcomes are possible in this case (0 or 1), while in the asymmetric model two faulty compared nodes always give mismatching outputs, and hence, the comparison outcome is 1.

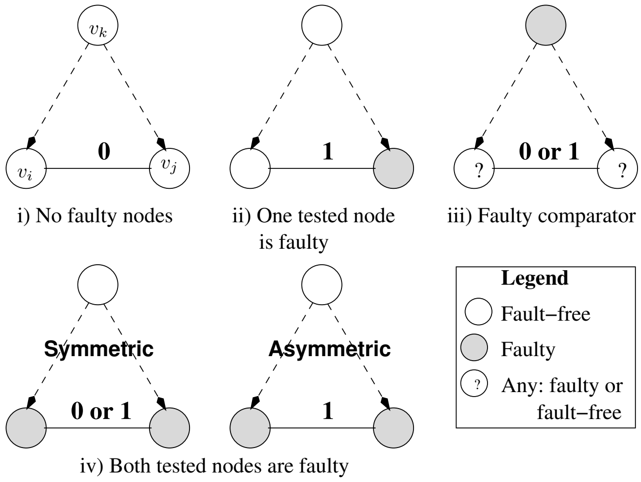

Figure 2. GCM's Invalidation Rules: v_k is the comparator node, and v_i and v_j are the compared nodes. The value besides the undirected edge denotes the comparison outcome.

In this model, the communication graph and the comparison graph are defined as follows. The communication graph is represented by the graph $G(V, E)$, where $V = \{v_1, v_2, \cdots, v_N\}$ denotes the set of N processing units and E refers to the set of communication links. The comparison graph is a multigraph whose edges represent comparison tests performed by the processing units on pairs of processors. Figure 3 (A), adopted from [7], shows an example of a five-units interconnection graph, and Figure 3 (B) depicts an example of a comparison graph. From now on we will refer only to the comparison graph.

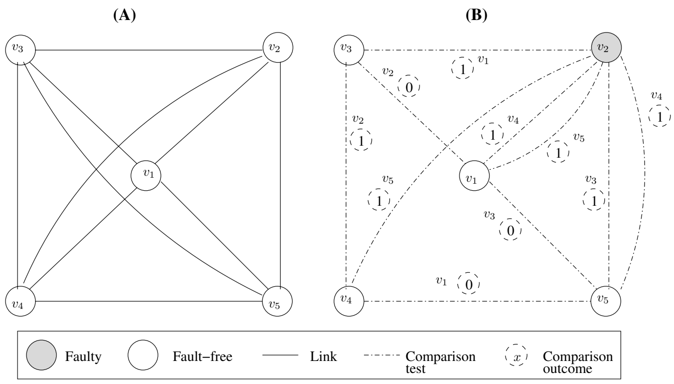

Figure 3. (A) An Interconnection Graph and (B) a Comparison Multigraph. The unit and the value beside each undirected edge of the multigraph denote, respectively, the comparator unit and the comparison outcome.

Each processing unit is assigned a subset of the other units to test. Testing is based on assigning a set of tasks $T = \{T_1, T_2, \cdots\}$ to the system's processors. A pair of processors (v_i, v_j) is assigned a task $T_l \in T$. Once the task T_l is executed by both processing units, the results are compared. We represent the comparison multigraph by an undirected graph $M(V, C)$, where V and C denote, respectively, the vertices (processors) and the edges (comparison tests). For every $v_i, v_j, v_k \in V$, $(v_i, v_j, v_k) \in C$, or simply $c_k^{ij} \in C$, iff processor v_k tests processors v_i and v_j by assigning them the same task. Each edge c_k^{ij} has a label associated with it. The binary value assigned to the label depends on the GCM's invalidation rules as discussed previously.

In the following, we will generalize the notation provided for SCM to GCM. Let $v_i, v_j, v_k \in V$ be any processing units in V. The subscript k will be used to designate a comparator node.

- c_k^{ij} denotes a comparison test performed by unit v_k between processors v_i and v_j. $c_k^{ij} \in C$.
- Γ_i^{-1} denotes to the set of units (neighbors) to which v_i is compared and is defined by $\Gamma_i^{-1} = \{v_j : \exists v_k \in V$ such that $c_k^{ij} \in C\}$.
- Γ_i defines the set of all comparators of processor v_i and all its neighbors, and is given by $\Gamma_i = \{v_k : \exists v_j \in \Gamma_i^{-1}$ such that $c_k^{ij} \in C\}$.
- Γ_k^{+1} denotes the set of processor pairs that are compared by the processing unit v_k, and is defined by $\Gamma_k^{+1} = \{(v_i, v_j) : c_k^{ij} \in C\}$.
- Ω denotes a comparison syndrome.
- Ω_k^{ij} refers to the outcome of the comparison test $c_k^{ij} \in C$ in the syndrome Ω, i.e., the result of the comparison test that is conducted by v_k on both processors v_i and v_j.
- Ω_F^{ijk} refers to the outcome of comparison test c_k^{ij} in the syndrome Ω_F, that is generated under the fault set F.
- $\Omega(v_k, \Gamma_k^{+1})$ refers to the subset of Ω containing only the outcomes of comparison tests executed by node v_k, and is defined by $\Omega(v_k, \Gamma_k^{+1}) = \{(v_i, v_j, \Omega_k^{ij}) : (v_i, v_j) \in \Gamma_k^{+1}\}$.
- $\Omega(v_i, \Gamma_i^{-1})$ refers to the subset of syndrome Ω containing only the comparison outcomes of tests performed between node v_i and all its neighbors, i.e., Γ_i^{-1}, and is given by $\Omega(v_i, \Gamma_i^{-1}) = \{(v_j, v_k, \Omega_k^{ij}) : v_j \in \Gamma_i^{-1}$ & $v_k \in \Gamma_i$ & $c_k^{ij} \in C\}$.
- F denotes the actual fault set in a faulty situation.
- $\Omega_{\widetilde{F}}$ refers to any comparison syndrome that can be generated, using the GCM's invalidation rules, under the fault set $\widetilde{F}$.

Definition 3: A fault set $F \subseteq V$ is *consistent* (or consistent) with the syndrome Ω under GCM if for each $\Omega_k^{ij} \in \Omega$ none of the following assertions holds:

i) $\Omega_k^{ij} = 0$, where $v_k \in V - F$ and v_i and/or $v_j \in F$
ii) $\Omega_k^{ij} = 1$ where $v_k, v_i, v_j \in V - F$

III. Artificial Immune Systems

An artificial immune system (AIS) is designed to mimic the operations of the human immune system which protects our body from the attacks of foreign organisms such as bacteria and viruses. These foreign organisms are called *antigens*. The main role of the immune system is to generate molecules, called *antibodies*, as a response to the detection of an antigen. The immune response is specific to each antigen. Once the antigen is detected, those antibodies that best recognize the antigen will proliferate by cloning. This process is known as the *clonal selection principle* [17] and is shown in Figure 4. The new cloned cells undergo *maturation* (or hypermutation), proportional to their affinity[2] to the antigen, in order to increase their receptor population (called repertoire). The highest affinity antibodies undergo the lowest mutation rates, whereas, the lowest affinity antibodies suffer the highest mutation rates. Since some of the maturated clones may be harmful for the body, they are eliminated. When the cloning and maturation processes have been completed, the immune system has improved the antibodies' affinity, resulting in the neutralization and/or the elimination of the antigen.

Once the antigen is neutralized, the immune system returns to its normal condition after eliminating the good cells. However, some cells remain circulating throughout the body as memory cells. When the immune system is attacked later by the same type of antigen or a similar one, these memory cells are activated, allowing for a better and more efficient response. This is known as *secondary response*.

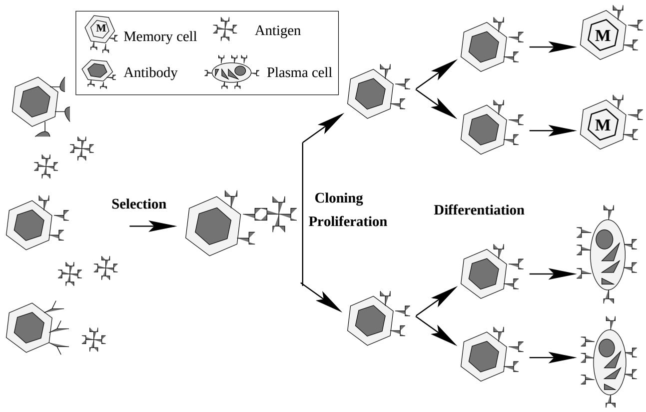

Figure 4. The Clonal Selection Principle.

By analogy we can show that the immune system and the fault identification problem share some common features. In fact, we can state that the system (the body) is exposed to a set of faults (the antigen). In diagnosable systems each fault set can be uniquely identified by one of its consistent syndromes (the antigen's molecular structure). In addition, the production of high "*affinity*" antibodies, pre-selected for the specific antigen, can be

[2]Affinity in biology is defined as a relationship or resemblance in structure between species that suggests a common origin.

considered as fault identification. Thus, artificial immune systems can be used as a basis for identifying the set of faulty processors. We shall now describe an AIS-based diagnosis algorithm using an affinity function for measuring the resemblance between the input syndrome and the one generated for a given potential solution.

The design of an artificial immune system is quite similar to the design of other traditional computational intelligence approaches, such us genetic algorithms, neural networks, etc. We first need to select a representation scheme of the search space. Then we need to define one or more evaluation functions to assess the behavior of the potential solutions. Finally, we have to choose or propose an immune algorithm which will be governing the dynamics of the search.

Our AIS-based diagnosis uses a binary scheme in order to represent potential solutions. A binary string of length $|V|$ is used to model an antibody (a potential solution) ab, defined such that for each $i \leq |V|, ab[i] = 1$ if the processor v_i is considered as faulty, and $ab[i] = 0$ if v_i is assumed to be fault-free. Since we are dealing with t-diagnosable systems, the number of bits at 1 in a potential solution should not exceed the bound t, and should not be less than one since in a faulty situation at least one processor is faulty. We need some evaluation functions that can provide a measure of the affinity between a given comparison syndrome and the syndrome generated by our algorithm from a potential solution. In the next section, we define the affinity functions that we used, based on the system-level fault diagnosis problem. We also give a detailed description of our immune diagnosis algorithm which is based on the clonal selection theory [18].

IV. The AIS-Based Fault Diagnosis Algorithm

In the following, we first introduce the affinity function used by the GCM and that used by the simple comparison model. Then, we present the AIS-based fault identification algorithm.

A. Affinity Measures for Fault Diagnosis

We now describe the approach used for evaluating potential solutions (i.e. potential fault sets) generated by our algorithm. We consider the GCM model here. Note that we first need to generate for each potential fault set F one of its compatible syndromes that has the particularity of including a subset of the input syndrome $\Omega_{\widetilde{F}}$, i.e., that satisfies the following property:

$$\Omega_F^{ijk} = \Omega_{\widetilde{F}}^{ijk}, \forall c_k^{ij} \in C \text{ such that } v_k \in F.$$

In fact, the outcomes of faulty comparators are unreliable and may be 0 or 1. Since v_k is faulty according to the potential fault set F, then one can assume by default that all its comparison outcomes are exactly those of the input syndrome. This step will reduce[3] the search space considerably since we are not considering all possible compatible syndromes, but a subset of them depending on the number of faulty units in F. The remaining comparison tests are evaluated using the GCM's invalidation rules (see Figure 2). Note that the syndrome Ω_F contains a subset of the input syndrome $\Omega_{\widetilde{F}}$ and it is compatible with F. Consequently, if $\Omega_F = \Omega_{\widetilde{F}}$, then one can conclude that the actual fault set is F, i.e., $\widetilde{F} = F$.

Given the potential fault set F and the previously generated syndrome Ω_F, to check whether Ω_F is equal to $\Omega_{\widetilde{F}}$ we proceed as follows. First, we verify if both syndromes are identical, from v_i's viewpoints where $v_i \in V$, by computing an *affinity* measure. Recall that affinity in biology is defined as a resemblance in structure between species that suggests a common origin. The affinity measure of two syndromes is defined as the number of comparison outcomes in the first syndrome that are identical to their corresponding in the second syndrome. Since v_i can be either a comparator node or a compared node, it follows that we need to consider both v_i's views (its view as a comparator node and its view as a node involved in comparison tests).

The affinity of both syndromes, Ω_F and $\Omega_{\widetilde{F}}$, from v_i's viewpoint as a comparator is defined by *affinity*$^{+1}(\Omega_F, \Omega_{\widetilde{F}}, v_i) =$

$$\begin{cases} 1 & \text{if } \Gamma_i^{+1} = \emptyset, \\ \frac{|\Omega_F(v_i,\Gamma_i^{+1}) \cap \Omega_{\widetilde{F}}(v_i,\Gamma_i^{+1})|}{|\Gamma_i^{+1}|} & \text{otherwise.} \end{cases} \quad (1)$$

Note that when $\Gamma_i^{+1} = \emptyset$, i.e., v_i does not perform any comparison, *affinity*$^{+1}(\Omega_F, \Omega_{\widetilde{F}}, v_i) = 1$, meaning that both syndromes are identical from this point of view since there are no comparison outcomes that differ.

Similarly, the affinity of both syndromes, Ω_F and $\Omega_{\widetilde{F}}$, from v_i's viewpoint as a node that is part of comparison tests is given by *affinity*$^{-1}(\Omega_F, \Omega_{\widetilde{F}}, v_i) =$

$$\begin{cases} 1 & \text{if } \Gamma_i^{-1} = \emptyset, \\ \frac{|\Omega_F(v_i,\Gamma_i^{-1}) \cap \Omega_{\widetilde{F}}(v_i,\Gamma_i^{-1})|}{|\Gamma_i^{-1}|} & \text{otherwise.} \end{cases} \quad (2)$$

The affinity of both syndromes, Ω_F and $\Omega_{\widetilde{F}}$, from all v_i's viewpoints can now be defined as follows: *affinity*$(\Omega_F, \Omega_{\widetilde{F}}, v_i) =$

$$\frac{\textit{affinity}^{+1}(\Omega_F, \Omega_{\widetilde{F}}, v_i) + \textit{affinity}^{-1}(\Omega_F, \Omega_{\widetilde{F}}, v_i)}{2} \quad (3)$$

affinity$(\Omega_F, \Omega_{\widetilde{F}}, v_i) \in [0, 1]$ and can be seen as the correctness probability of the potential status of node v_i. The affinity of both syndromes from a point of view of all nodes can be computed as follows:

$$\textit{affinity}(\Omega_F, \Omega_{\widetilde{F}}) = \frac{\sum_{v_i \in V} \textit{affinity}(\Omega_F, \Omega_{\widetilde{F}}, v_i)}{N} \quad (4)$$

When considering the simple comparison model, the affinity computation would be much easier. Since each processor v_i would participate in a comparison, only as a candidate and not as a comparator, it would have only one viewpoint. So, the affinity between syndromes Ω_F and $\Omega_{\widetilde{F}}$ from v_i's viewpoint is defined by:

$$\textit{affinity}(\Omega_{\widetilde{F}}, \Omega_F, v_i) = \frac{|\Omega_{\widetilde{F}}(v_i) \cap \Omega_F(v_i)|}{|\Gamma(v_i)|} \quad (5)$$

[3]Specially when large fault sets are considered.

The overall affinity of the syndromes Ω_F and $\Omega_{\widetilde{F}}$, would be defined similarly as before (see Eq. 4).

Note that *affinity*$(\Omega_F, \Omega_{\widetilde{F}}) \in [0,1]$ and can be considered as the correctness probability of the potential solution F being the actual fault set. It follows that if *affinity*$(\Omega_F, \Omega_{\widetilde{F}}) = 1$, then $F = \widetilde{F}$. Since we used the particular syndrome Ω_F, generated as described above, we can state that there exists only a unique fault set F that satisfies $\Omega_F = \Omega_{\widetilde{F}}$. The problem of system-level diagnosis now becomes the problem of finding a potential fault set F such that *affinity*$(\Omega_F, \Omega_{\widetilde{F}}) = 1$. This idea is used for designing our immune diagnosis algorithm, which is described below.

B. Algorithm Description

The flowchart of the AIS-based algorithm is given in Figure 5. The immune diagnosis algorithm accepts as an input the comparison graph $G(V,C)$ and an input syndrome $\Omega_{\widetilde{F}}$ generated after the occurrence of faults is detected. The AIS-based diagnosis proceeds as follows:

1) **Initial Population:** The initial population POP is created by generating in a bitwise fashion potential antibodies (solutions) using the following heuristic, where POP_SIZE is the population size and $GenRandom(x)$ generates randomly a potential solution that corresponds to a fault set of cardinality x:

```
Heuristic InitializePopulation (POP) {
  nbFaults = 1;
  for(i = 0; i < POP_SIZE; i++) {
    do {
      anIndividual = GenRandom(nbFaults);
    }while (anIndividual ∈ POP)
    POP[i] = anIndividual;
    nbFaults = (nbFaults modulo t) + 1;
  }
}
```

This heuristic generates an initial population that includes at least one fault set of each cardinality. This helps increasing the diversity of the initial population, and hence, improve the search capabilities of our algorithm instead of just relying on the use of a randomly generated initial population. Note however, that the generated solutions for the initial population are still random.

2) **Affinity Computation:** Compute for each individual in the population its affinity with respect to the input syndrome $\Omega_{\widetilde{F}}$, using Eq. (4).

3) **Highest Affinity Selection:** Select α highest affinity antibodies (potential solutions) from POP composing a new set SUB_POP of high affinity individuals with relation to the input syndrome $\Omega_{\widetilde{F}}$.

4) **Cloning:** The α previously selected antibodies are cloned independently by generating identical copies of each individual. The number of clones is proportional to the antigenic affinity. Let $CLONES$ denote the set of clones.

5) **Maturation:** The newly generated clones undergo an affinity maturation process inversely proportional to the antigenic affinity, generating a new population $CLONES^\star$ of maturated clones. The clones suffer a mutation where a number of genes of each clone are selected for mutation. The higher the affinity, the smaller the mutation rate. The number of genes to be mutated in a potential antibody ab is computed as follows:

$$\#GenesToBeMutated_{ab} = \frac{N}{2}\Big(1 - \frac{affinity(\Omega_{F_{ab}}, \Omega_{\widetilde{F}})}{\epsilon}\Big)$$

where $\epsilon = (rand() \ modulo \ t) + 1$, and $rand()$ generates a random integer. The term ϵ has been introduced in order to force clones to undergo various degrees of mutation. This will ensure that the number of genes that will be flipped will not be the same for all clones, and hence, forcing their offspring to explore different regions of the search space. This will increase the diversity of the population, and hence allow the immune algorithm to investigate multiple potential candidates.

Flipping genes randomly without taking into account the fact that they may be correct is unpractical and may lead to high computation time. Hence, we adopted adaptive mutation [19]. In adaptive mutation, each gene is given an individual performance, and hence, the overall performance of the antibody is given by the normalized sum of performances of all of its genes. In our case, Eq. (3) (for GCM) and (Eq. 5) (for SCM) are used to evaluate the affinity of genes. Furthermore, to enhance diversity three types of mutation operators have been used:

i) $FlipOneGene$: performs the mutation of one gene.

ii) $ExchangeTwoGenes$: exchanges two genes. If both genes are identical, they are simply flipped using the first mutation operator.

iii) $ExhancgeThreeGenes$: performs the exchange of three genes. This mutation operator is in fact the combination of the first two.

The type of mutation is randomly chosen. However, genes to suffer mutation are selected regarding their individual affinities. In fact, Eq. (3) (for GCM) or (Eq. 5) (for SCM) defines the individual affinity of each gene, and can be seen as the correctness probability of the potential status represented by this gene.

6) **Affinity Re-computation:** Since clones are mutated, their previously computed affinities are no longer valid. Hence, this step computes the affinities of the maturated clones $CLONES^\star$ with relation to the antigen represented by the input syndrome $\Omega_{\widetilde{F}}$.

7) **Highest Affinity Re-selection:** the α highest affinity antibodies are selected from the maturated clones to form the new population $POP^\star$.

8) **Replacement:** The γ lowest affinity antibodies from $POP^\star$ are replaced by new individuals generated

randomly.

9) **Stopping Criteria:** The immune diagnosis algorithm stops when it finds an antibody $ab^\star \in POP^\star$ that matches the antigen (the actual faut set), i.e., *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}) = 1$.

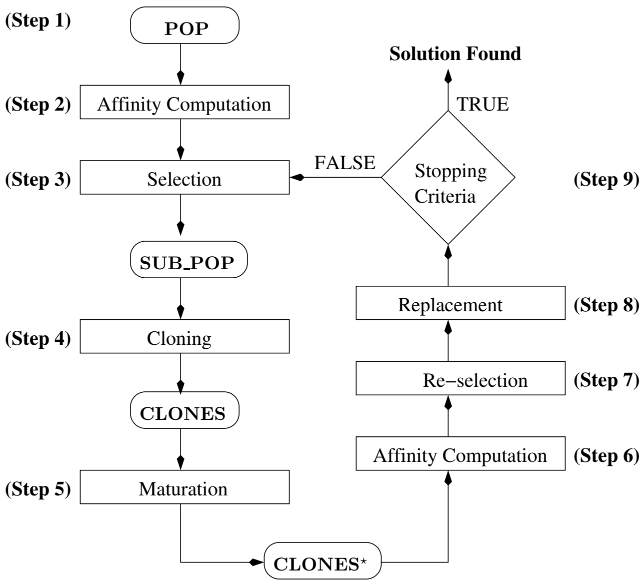

Figure 5. Flowchart of the AIS-based fault identification algorithm.

Note that the values of all the parameters mentioned above, i.e., α, γ, and POP_SIZE are determined experimentally. A complete sensitivity analysis of the clonal selection algorithm with relation to these user-defined parameters can be found in [18], where the authors suggest initial settings that may be considered when using a clonal-selection based AIS.

V. Algorithm Analysis and Discussion

A. Proof of Correctness

A diagnosis algorithm is said to be *correct* if all fault-free processors are correctly identified. The correctness proof of our algorithm follows trivially from the definition of the affinity measure given in Eq. (4). In the following, we provide the proof of correctness for GCM. Note that the proof for SCM can be easily deduced from that of GCM.

Theorem 1: For any t-diagnosable multiprocessor system with the comparison assignment $G(V, C)$ and an input syndrome $\Omega_{\widetilde{F}}$ generated under a symmetric or asymmetric comparison model, the AIS-based diagnosis algorithm is correct, i.e., identifies the actual set $\widetilde{F}$ of faulty processors.

Proof: The AIS-based algorithm stops once it finds an antibody $ab^\star \in POP^\star$ such that *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}) = 1$. Let $F_{ab^\star}$ denote the fault set found by the diagnosis algorithm. Assume that $F_{ab^\star}$ is not the actual fault set, i.e., $F_{ab^\star} \neq \widetilde{F}$, and hence, $\Omega_{\widetilde{F}} \neq \Omega_{ab^\star}$. Since the optimal solution found satisfies *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}) = 1$ it follows by Eq. (4) that for each $v_i \in V$, *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}, v_i) = 1$, since by definition *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}, v_i) \in [0, 1]$. Hence, for each $v_i \in V, \Omega_{\widetilde{F}}(v_i) = \Omega_{ab^\star}(v_i)$ by definition of $\Omega(v_i)$ according to SCM. Similarly, for the GCM model, $\Omega_{ab^\star}(v_i, \Gamma_i^{-1}) = \Omega_{\widetilde{F}}(v_i, \Gamma_i^{-1})$ and $\Omega_{ab^\star}(v_i, \Gamma_i^{+1}) = \Omega_{\widetilde{F}}(v_i, \Gamma_i^{+1})$. It follows that $\Omega_{\widetilde{F}} = \Omega_{ab^\star}$ which contradicts our assumption, and hence, $F_{ab^\star} = \widetilde{F}$, i.e., the fault set $F_{ab^\star}$ found by the AIS-based diagnosis algorithm is the actual fault set. ■

B. Convergence of the AIS-Based Diagnosis Algorithm

Recently, the convergence of artificial immune systems has gained much more attention [20]–[22]. In [20], de Castro and Timmis studied the convergence properties of the AINET (an artificial immune network) model. They concluded that the dynamics within the network makes it very difficult to formally prove the convergence properties of AINET. Hence, they instead investigated the convergence empirically, in order to derive reasonable convergence criterion for the network learning process. In [21], Ruochen et al. proved using Markov chain that the immunity clonal strategy algorithm is convergent. In [22], Villalobos-Arias et al. presented the first proof of convergence for the multiobjective artificial immune system algorithm. Our AIS-based diagnosis algorithm can also be seen as a multiobjective AIS-based algorithm, since we are trying to optimize the affinity function. It follows that the proof of convergence of our algorithm follows trivially from that provided by Villalobos-Arias et al. in [22].

C. Time Complexity of the AIS-Based Diagnosis Algorithm

In [18], de Castro presented an analysis of the complexity of the clonal selection algorithm on which our AIS-based diagnosis algorithm is based. However, the time complexity provided assumed that the number of iterations (generations) is bounded. In our diagnosis algorithm such bound does not exist, but may be estimated experimentally. We believe that the estimation of the time complexity of AIS-based algorithms can be accomplished using previous work done for genetic algorithms [23], which is beyond the scope of this paper and is matter of ongoing research.

D. Practical Considerations

In order to avoid rounding-off errors that may occur while computing the affinities of potential solutions using Eq. (4), we used a non-normalized version of the affinity function given by:

$$\text{affinity}(\Omega_{\widetilde{F}}, \Omega_F, v_i) = |\Omega_{\widetilde{F}}(v_i) \cap \Omega_F(v_i)| \quad (6)$$

$$\text{affinity}(\Omega_{\widetilde{F}}, \Omega_F) = \sum_{v_i \in V} \text{affinity}(\Omega_{\widetilde{F}}, \Omega_F, v_i) \quad (7)$$

As a consequence, the stopping criteria should be changed to *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab^\star}) = N$.

In order to time-optimize the computation of the affinity function we used an *affinity interpolation* heuristic. The interpolation process requires the following steps:

1) A *reference* population of t individuals, denoted by $\overline{POP}$, is randomly generated using the heuristic $InitializePopulation$ provided in Section IV. For $1 \leq i \leq t$, $\overline{POP}[i]$ denotes an antibody ab_i corresponding to a fault set of cardinality i.
2) Affinities of individuals in the reference population are computed using Eq. (6) and (7).

Now, in order to interpolate the affinity of a potential solution ab we proceed as follows. Let F_{ab} denote the potential fault set represented by the potential antibody ab.

3) The first step consists in choosing the individual from which interpolation will be performed. Here, we simply choose $\overline{POP}[|F_{ab}|]$, denoted by $\overline{ab}$, as a basis for affinity interpolation. Off course, this may not be the best choice from the reference population, but it is accessed directly without wasting time in seeking for the best candidate.
4) The second step consists in deriving the affinity of the potential solution ab from that of the reference individual $\overline{ab}$. The derivation process proceeds as follows. For each $v_i \in V$,
   i) if $ab[v_i] = \overline{ab}[v_i]$, then compute *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab}, v_i) =$ *affinity*$(\Omega_{\widetilde{F}}, \Omega_{\overline{ab}}, v_i)$
   ii) if $ab[v_i] \neq \overline{ab}[v_i]$, then compute *affinity*$(\Omega_{\widetilde{F}}, \Omega_{ab}, v_i)$ as before, and adjust the affinities of all neighbors $v_j \in \Gamma_i$.

VI. Experimental Results

The AIS-based fault identification algorithm has been implemented in C++ and evaluated using a PC equipped with a Pentium IV CPU 2.4GHz and 256MB of RAM. We developed two versions of the algorithm—one for the simple comparison model (SCM) and the other for the generalized comparison model (GCM). The performance evaluation of the algorithms is based on randomly generated comparison assignment graphs, and using all possible fault sets that may occur in a t-diagnosable situation.

A. Performance Under the Simple Comparison Model

For the symmetric comparison model we used basically t-diagnosable systems from the special design $D_t(N)$ with $t \leq \lceil \frac{N}{2} \rceil$, where $\lceil x \rceil$ denotes the largest integer not larger than x. A comparison assignment graph $G(P, C)$ is a $D_\alpha(N)$ design iff for all $p_i \in P, |\Gamma_i| \geq \alpha$, i.e., each processor is at least compared with α other processors. The example given in Figure 3 (B) is in fact a $D_3(7)$ comparison graph. For the asymmetric comparison model, we considered the special design $D_{2t}(N)$ with $t \leq \lceil \frac{N}{2} \rceil$. $D_t(N)$ and $D_{2t}(N)$ designs have been shown to be t-diagnosable in [14], and they can be easily generated (the only reason why we have adopted them).

The parameters for the algorithm have been set to the following values. POP_SIZE, the population size, is set to 10 if $N \leq 25$, and to 20 otherwise. α, the number of highest affinity antibodies selected in Step 3 of the diagnosis algorithm to undergo cloning, is $\alpha = POP\_SIZE$. This was suggested by de Castro [18] for optimization tasks. This mean that we are not applying the affinity proportionate cloning, hence, the number of clones generated for each of the POP_SIZE antibodies is assumed to be the same. No antibody (potential solution) is privileged for its affinity. Number of clones for each selected antibody (Step 4) = POP_SIZE. γ, the number of lowest affinity maturated clones that are replaced by new individuals, is $\gamma = 0.25 * POP\_SIZE$ (25% of the population).

1) First Set of Experiments: In the first set of experiments, the performance of the AIS-based diagnosis under the symmetric comparison model has been analyzed by considering two diagnosable systems $D_{25}(50)$ and $D_{50}(100)$. We considered all possible types of faults, i.e., the number of faults has been varied from 1 to t. For each value 1000 fault sets have been generated randomly. Figure 9 shows the average-case and the worst-case behaviors of the immune diagnosis algorithm. As one may expect, the more the number of faults increases, the more the diagnosis algorithm requires additional time to diagnose a faulty situation. However, the increase is not too significant. For example, for $D_{50}(100)$ and for a single fault the diagnosis requires in the average-case 0.52 seconds, and 8 seconds in the worst-case. For the extreme situation, i.e., half of the processors are faulty, the diagnosis requires in the average and worst cases, respectively, 20.83 and 22 seconds. Furthermore, the worst-case behavior is interesting, in that it gives an idea about the convergence of the diagnosis algorithm.

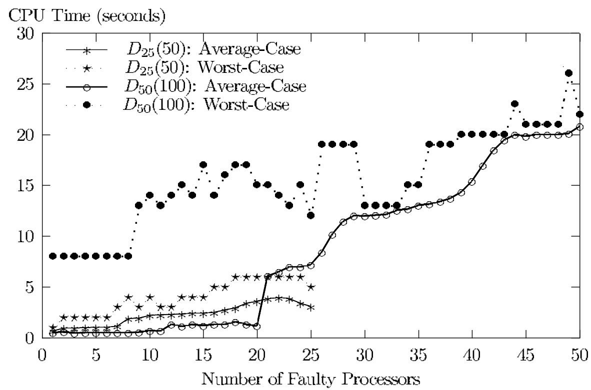

Figure 6. CPU Time vs. Number of Faulty Processors, in the Average and Worst Cases.

2) Second Set of Experiments: Our second set of simulations aimed at analyzing the performance of our immune fault identification algorithm when the number of processors in the systems increases. In Figure 7, the size of the system is varied from 5 to 100. For each size x, a comparison assignment $D_{\lceil \frac{x}{2} \rceil}(x)$ is randomly generate and 1000 fault sets, also generated randomly, have been experimented. As one can easily see from Figure 7, the CPU time, in the average and worst cases, needed to diagnose a faulty situation increases slowly. For large systems composed of hundreds of processors, Figure 8 depicts the behavior of the AIS-based diagnosis. Again, the worst-case behavior is still near that of the average-case.

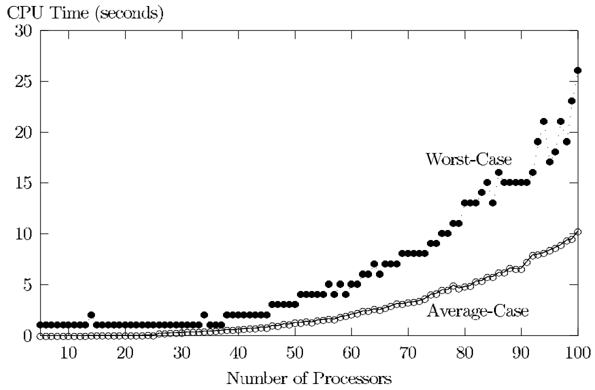

Figure 7. CPU Time vs. Number of Processors, in the Worst and Average Cases, with $|V| \leq 100$.

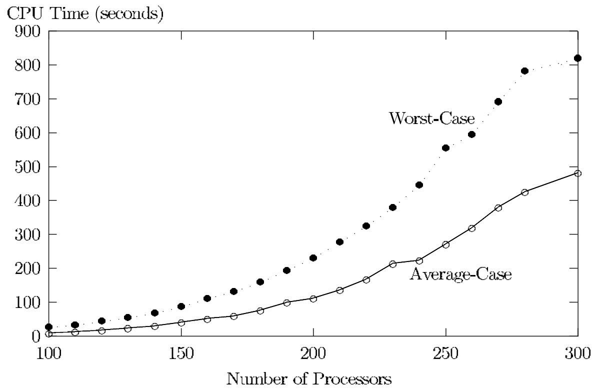

Figure 8. CPU Time vs. Number of Processors for Large Systems ($|V| \geq 100$), in the Worst and Average Cases.

B. Performance Under the Generalized Comparison Model

Due to the fact that both characterization of GCM-based t-diagnosable systems, the one provided by Sengupta and Dahbura [7] and that developed by Wang, Blough, and Alkalaj in [24], are rather theoretical and difficult to implement specially for large systems, we used a special class of diagnosable systems. Since the immune diagnosis approach does not rely on the comparison multigraph topology, then it follows that the following results remain valid for general graph structures.

Definition 4: A system with interconnection topology represented by the undirected graph $G = (V, E)$ is a $D_\alpha(N)$ *design* if for each pair of vertices $v_i, v_j \in V$ such that $j - i = N$ modulo k, where $k = 1, 2, \ldots, \alpha$, an undirected edge between v_i and v_j exists.

Definition 5: A multigraph $M = (V, C)$ is a $D_t^*(N)$ *design* if it is produced from a $D_{t+1}(N)$ system topology under the MM* model.

We used multigraphs from the special design $D_t^*(N)$ which have been shown in [7] to be t-diagnosable. More importantly, they can be easily generated even when very large systems are considered.

The settings of the immune parameters that have been used are as follows. The population size has been set to: $POP\_SIZE = 5$ if $N \leq 25$, $POP\_SIZE = 10$ if $25 \leq N \leq 50$, $POP\_SIZE = 15$ if $50 \leq N \leq 75$, otherwise $POP\_SIZE = 20$. As before, the number of highest affinity antibodies α is set to POP_SIZE, (i.e., $\alpha = POP\_SIZE$) and the number of lowest affinity maturated clones γ is set to 25% of the population (i.e., $\gamma = 0.25 * POP\_SIZE$). The number of clones generated in Step 5 is equal to the number of individuals in the population, i.e., $NumberOfClones = POP\_SIZE$.

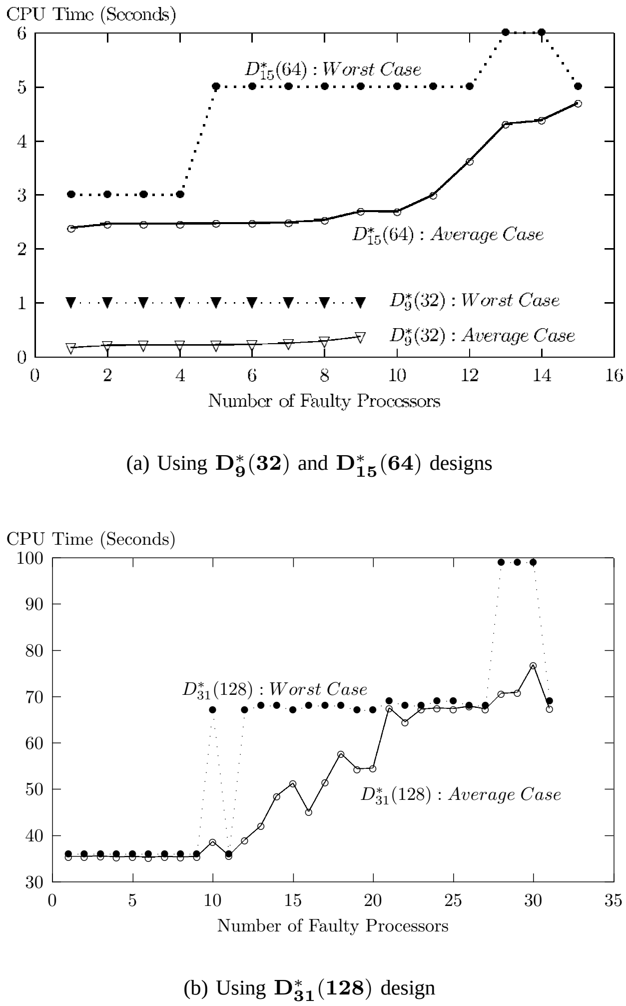

Figure 9. CPU Time vs. Number of Faulty Processors, in the Average and Worst Cases Using $D_9^*(32)$, $D_{15}^*(64)$, and $D_{31}^*(128)$ multigraphs.

A series of simulations were conducted in order to evaluate the performance of the immune diagnosis algorithm. In the first set of experiments, we considered the comparison multigraphs $D_9^*(32)$, $D_{15}^*(64)$, and $D_{31}^*(128)$. The number of faulty nodes was varied from 1 to t, and 1000 randomly generated fault sets have been ex-

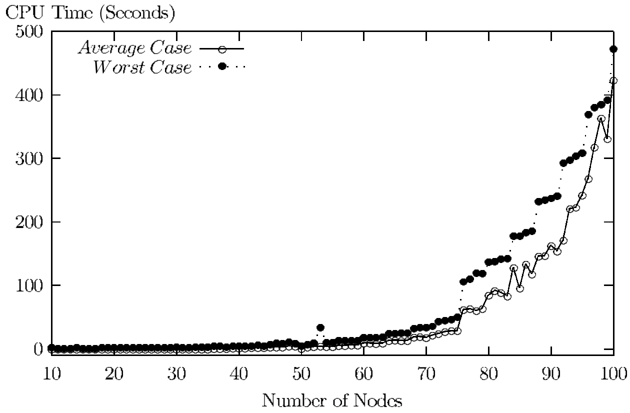

Figure 10. CPU Time vs. Number of Processors, in the Average and Worst Cases.

perimented with. Figure 9 shows the CPU time in the average and worst cases. As one may expect, the CPU time is proportional to the number of faulty nodes.

The second set of experiments involved varying the number of nodes from 10 to 100. For each value of $N \in [10, 100]$, a $D^*_{t-1}(N)$ system was generated. The value of t was set to $\lceil \frac{N}{2} \rceil$ if $N < 50$, and to $\lceil \frac{N}{4} \rceil$ otherwise. For each system, 1000 randomly generated fault sets have been experimented with. Figure 10 depicts the average and the worst-case CPU-times for the immune diagnosis algorithm. Note that the CPU time in the worst case is quite near to the average behavior, hence suggesting an efficient overall performance in all cases.

VII. Comparison with Related Work

The fault identification problem has been extensively studied resulting in the development of various diagnosis algorithms [1], [2]. Hakimi and Chwa developed an $O(|C|)$ diagnosis algorithm for their symmetric comparison model [5]. For the asymmetric comparison model, various fault identification algorithms have been proposed. The Ammann and Cin's $O(|V|^2)$ sequential diagnosis algorithm has been designed for a subset of t-diagnosable systems [25].

Among previous solutions to this problem, there have been only a few results that consider the system-level diagnosis problem under GCM. Sengupta and Dahbura's [7] diagnosis algorithm provides a correct and complete solution under GCM. Unfortunately, this algorithm has a time complexity of $O(N^5)$, which makes it unpractical specially when considering large systems composed of hundreds or even thousands of nodes.

In [10], Blough and Pelc studied the complexity of fault diagnosis under comparison models and they provided efficient algorithms for diagnosing systems for which the comparison assignment is a bipartite graph. A diagnosis algorithm has also been proposed, by Blough and Brown, for their broadcast comparison model which requires $O(|C| + t^2|V|)$ steps under asymmetric assumptions. In [9], Chessa and Santi presented a new comparison-based diagnostic model based on one-to-many communication which takes advantage of the shared nature of ad-hoc networks. They introduced a diagnosis protocol and gave two implementations of their model based on whether or not the network topology can change during diagnosis.

The need for more efficient solutions for diagnosis of large-scale systems, motivated the use of heuristic algorithms and evolutionary approaches. Genetic algorithms were used to solve the self-diagnosis problem, both under SCM [26] and under GCM [27], [28]. The genetic approach used by Abrougui and Elhadef in [28] has been shown to be quite efficient in identifying the set of faulty processors, in the average case. However, the disadvantage of these approaches is that in the worst case, the running time of these algorithms can be very large. In fact, the genetic diagnosis algorithm suffers from a loss in population diversity due especially to the use of an adaptive mutation operator rather than a random one. This has resulted in high-worst case behavior, specially for large systems composed of hundreds or thousands of processors. On the other hand, the artificial immune-based approach proposed in this paper avoids such erratic worst case behavior because the AIS-based algorithms do not suffer from a loss of population diversity. As we saw, the artificial immune-based diagnosis algorithm performs efficiently both in the average and worst cases. In fact, in our experiments, the worst case running time was never more than twice the average case behavior (for example see Figures 8 and 10).

VIII. Conclusion

The problem of fault identification in diagnosable systems based on an input syndrome, bears certain similarities to the process by which the immune system generates antibodies against specific antigens. Thus, artificial immune systems can be used to design solutions to the fault-diagnosis problem as was shown in this paper. Artificial immune-based algorithms were designed for fault identification under various comparison based models. The experimental results from extensive simulations showed that the AIS-based diagnosis approach can correctly identify the faulty processors. Moreover, the simulation results indicate that the AIS-based diagnosis algorithm is efficient, in both the worst and average cases, when considering large systems, or when the number of faulty processors is high.

Our results showed that the AIS-based approaches are an attractive and viable alternative to present fault diagnosis techniques. Further experimental analysis and comparisons with existing solutions would be helpful in understanding the pros and cons of using artificial immune systems in designing solutions to the diagnosis problem. We believe that given the features of the immune diagnosis approach, a natural extension would be to apply this new approach to the probabilistic models for fault diagnosis. It would also be interesting to experiment and analyze the use of alternative mechanisms, such as *swarm intelligence* and *memetic algorithms*, for solving the system-level fault diagnosis problem.

REFERENCES

[1] M. Barborak, M. Malek, and A. Dahbura, "The Consensus Problem in Fault-Tolerant Computing," *ACM Computing Surveys*, vol. 25, no. 2, pp. 171–220, June 1993.

[2] S. Lee and K. Shin, "Probabilistic Diagnosis of Multiprocessor Systems," *ACM Computing Surveys*, vol. 26, no. 1, pp. 121–139, March 1994.

[3] F. Preparata, G. Metze, and R. Chien, "On the Connection Assignment of Diagnosable Systems," *IEEE Trans. on Electron. Comput.*, vol. 16, no. 6, Dec. 1967.

[4] M. Malek, "A Comparison Connection Assignment for Diagnosis of Multiprocessor Systems," in *Proc. 7th Int. Symp. on Comput. Architecture, New York*. Association for Computing Machinery Publ., 1980, pp. 31–35.

[5] S. Hakimi and K. Chwa, "Schemes for Fault Tolerant Computing: A Comparison of Modularly Redundant and t-Diagnosable Systems," *Inform. Contr.*, vol. 49, pp. 212–238, June 1981.

[6] J. Maeng and M. Malek, "A Comparison Connection Assignment for Self-Diagnosis of Multiprocessor Systems," in *Proc. 11th Int. Symp. on Fault-Tolerant Comput.*, 1981, pp. 173–175.

[7] A. Sengupta and A. Dahbura, "On Self-Diagnosable Multiprocessor Systems: Diagnosis by the Comparison Approach," *IEEE Trans. on Computers*, vol. 41, no. 11, pp. 1386–1395, Nov. 1992.

[8] D. Blough and H. Brown, "The Broadcast Comparison Model for On-Line Fault Diagnosis in Multiprocessor Systems: Theory and Implementation," *IEEE Trans. on Computers*, vol. 48, no. 5, pp. 470–493, May 1999.

[9] S. Chessa and P. Santi, "Comparison-Based System-Level Fault Diagnosis in Ad Hoc Networks," in *Proc. of the 20th IEEE Symp. on Reliable Dist. Systems*, 2001, pp. 257–266.

[10] D. Blough and A. Pelc, "Complexity of Fault Diagnosis in Comparison Models," *IEEE Trans. on Computers*, vol. 41, no. 3, pp. 318–324, March 1992.

[11] J. Amaral, J. Amaral, R. Tanscheit, and M. Pacheco, "An immune inspired fault diagnosis system for analog circuits using wavelet signatures," in *Proc. of the 2004 NASA/DoD Conference on Evolvable Hardware (EH'04)*. IEEE Comput. Soc. Publ., 2004, pp. 138–141.

[12] D. Dasgupta, K. KrishnaKumar, D. Wong, and M. Berry, "Negative Selection Algorithm for Aircraft Fault Detection," in *Proc. of the 3rdt Inter. Conf. on Artificial Immune Systems (ICARIS)*, 2004, pp. 1–13.

[13] Y. Ishida, "Active Diagnosis by Self-Organization: An Approach by The Immune Network Metaphor," in *Proc. of the 15th Inter. Joint Conf. on Artificial Intelligence (IJCAI 97)*. Morgan Kaufmann Pub., 1997, pp. 1084–1091.

[14] A. Pelc, "Undirected Graph Models for System Level Fault Diagnosis," *IEEE Trans. on Electron. Comput.*, vol. 40, no. 11, pp. 1271–1276, Nov. 1991.

[15] F. Barsi, F. Grandoni, and P. Maestrini, "A Theory of Diagnosability Without Repairs," *IEEE Trans. on Computers*, vol. C-25, pp. 585–593, June 1976.

[16] S. Kreutzer and S. Hakimi, "Adaptive Fault Identification in Two New Diagnostic Models," in *Proc. 21st Allerton Conf. on Commun., Cont. and Comput.*, Univ. of Illinois, Urbana, Ill, 1983, pp. 353–362.

[17] L. N. D. Castro and J. I.Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. London: Springer-Verlag, 2002.

[18] L. N. D. Castro and F. J. V. Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evolut. Comput.*, vol. 6, no. 3, pp. 239–251, 2002.

[19] M. Elhadef and D. Coley, "Adaptive Mutation for Semi-Separable Problems," in *Proc. of the Genetic and Evol. Comput. Conf.*, CA, USA, 2001, pp. 306–312.

[20] L. N. D. Castro and J. Timmis, "Hierarchy and Convergence of Immune Networks: Basic Ideas and Preliminary Results," in *Proc. of the 1st Inter. Conf. on Artificial Immune Systems (ICARIS)*, 2002, pp. 231–240.

[21] L. Ruochen, D. Haifeng, and J. Licheng, "Immunity Clonal Strategies," in *Proc. of the 5th Int. Conf. on Computational Intell. and Multimedia Appl.*, 2003, pp. 290–295.

[22] M. Villalobos-Arias, C. A. C. Coello, and O. Hernández-Lerma, "Convergence Analysis of a Multiobjective Artificial Immune System Algorithm," in *Proc. of the 3rd Int. Conf. on Art. Immune Syst. (ICARIS)*, 2004, pp. 226–235.

[23] C. Ankenbrandt, "An Extension to the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms," in *Foundations of Genetic Algorithms*. Rawlins G. Ed., Morgan Kaufmann, 1991, pp. 53–68.

[24] H. Wang, D. Blough, and L. Alkalaj, "Analysis and Experimental Evaluation of Comparison-Based System-Level Diagnosis of Multiprocessor Systems," in *Proc. 24th Int. Symp. on Fault-Tolerant Comput.*, 1994, pp. 55–64.

[25] E. Ammann and M. D. Cin, "Efficient Algorithms for Comparison-Based Self-Diagnosis," in *Self-Diagnosis and Fault-Tolerance*, Werkhefte der Universität Tübingen, 4 At-tempo-Verlag, Tübingen, 1981, pp. 1–18.

[26] M. Elhadef and B. Ayeb, "Efficient Comparison-Based Fault Diagnosis of Multiprocessor Systems Using an Evolutionary Approach," in *proc. of the 15th Int. Parallel and Distributed Processing Symp.*, CA, USA, April 2001.

[27] ——, "An Evolutionary Algorithm for Generalized Comparison-Based Self-Diagnosis of Multiprocessor Systems," *Applied Artif. Intell.*, vol. 16, pp. 73–95, Jan. 2002.

[28] K. Abrougui and M. Elhadef, "Parallel Self-Diagnosis of Large Multiprocessor Systems Under the Generalized Comparison Model," in *Proc. of the 11th Int. Conf. on Parallel and Dist. Syst.*, Japan, Jul. 2005, pp. 78–84.

Mourad Elhadef received the B.Sc. and M.Sc. degrees in Computer science from the Institut Supérieur de Gestion, Tunis, Tunisia, and the Ph.D. degree in computer science from the university of Sherbrooke, Québec, Canada. He is currently an assistant professor at the School of Information Technology and Engineering, University of Ottawa, Ontario, Canada. His research interests include fault tolerance and fault diagnosis in distributed, wireless and ad-hoc networks, fault tolerance of optical networks, artificial intelligence, swarm intelligence, artificial immune systems and genetic algorithms.

Shantanu Das is currently a Ph.D. candidate at the University of Ottawa, Canada. He received his M.Tech and BIT degrees in computer science from the Indian Statistical Institute and the University of Delhi, India, in 2004 and 2002, respectively. His current research interests include distributed computing, fault tolerance in distributed systems and network security. **Amiya**

Nayak is an associate professor in the School of Information Technology and Engineering at University of Ottawa since 2002. He received his B.Math. degree from University of Waterloo in 1981, and Ph.D. in computer engineering from Carleton University in 1991. He has over 15 years of industrial experience in software engineering in avionics and telecommunication applications, working at CMC Electronics and Nortel Networks. His current research interests are in the area of fault tolerance and testing of VLSI systems, distributed systems, and mobile ad hoc networks.

Dr. Nayak has over 80 publications in refereed journals and conference proceedings. He is in the Editorial Boards of International Journal of Parallel, Emergent and Distributed Systems and International Journal of Computing and Information Science.