EFFICIENT AND OPTIMAL FAULT-TO-SPARE ASSIGNMENTS IN DOUBLY FAULT TOLERANT ARRAYS

Nimish Shrivastava and Rami Melhem
Department of Computer Science
The University of Pittsburgh
Pittsburgh, PA 15260

Abstract

Given a doubly fault tolerant system with m faults, we present an algorithm for finding a fault-to-spare assignment for the m faults in $O(m)$ *time. This improvement over the* $O(m^{1.5})$ *bi-partite graph technique for finding an assignment is obtained by partitioning the problem into independent regions and proving that, in each region, an assignment may be found in linear time. The region approach also allows for an efficient solution of a related problem. Namely, finding a fault-to-spare assignment which minimizes the number of uncovered nodes. For two dimensional arrays augmented with one row and one column of spares, such an optimal assignment may be found in linear time.*

Introduction and Problem Formulation

Current technology allows for the efficient implementation of large arrays of processing elements which are regularly interconnected. However, as the sizes of such arrays increase, so does the importance of providing techniques for tolerating manufacturing-time defects and run-time faults. One such technique which has become popular in recent years is the augmentation of processor arrays with spares, thus allowing fault tolerance through reconfiguration [3, 5, 7, 8, 13]. Spares may be added to arrays either globally or locally [12]. In the former case, any spare may replace any faulty processor in the system, while in the latter case, restrictions are imposed regarding which spare may replace which processor. More specifically, in a local sparing system, each spare, S, may only replace faulty processors in a subset, $PS(S)$ of processors. Using the same notation, we can characterize a global sparing system to be one in which $PS(S)$ consists of all the processors in the system, for any spare S. Clearly, global sparing provides more fault tolerant capabilities than local sparing, but requires more complex hardware support.

In [1], it was shown that, for a given hardware overhead, the fault tolerance capabilities of local sparing strategies may be optimized if coverage overlap is minimized. That is, if for any two spares, S and S', the intersection of $PS(S)$ and $PS(S')$ is minimized. For doubly fault-tolerant systems, that is for systems where each faulty processor may be replaced by one of two spares, minimum coverage overlap is obtained when the processors are logically arranged as a two dimensional mesh with one added row and one added column of spares. A spare may replace a faulty processor in its row or its column. Figure 1 shows such an arrangement that we will call a "2-D augmented mesh".

In order to generalize the architecture of Figure 1, we consider a set of nodes (processors) $X = \{x_1, x_2, \ldots x_N\}$, where some nodes in X are designated as primary nodes and some as spare nodes. We assume that the system is modular in the sense that X is divided into subsets called

0-8186-2457-4/91 $1.00 © 1991 IEEE

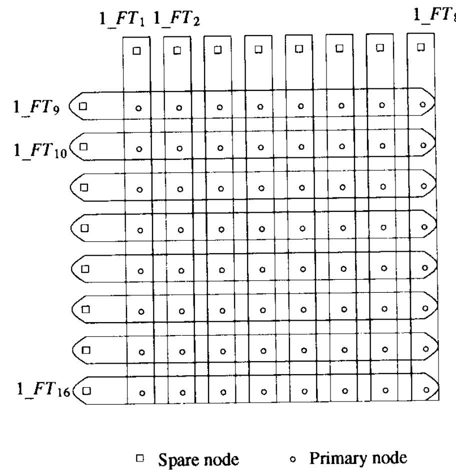


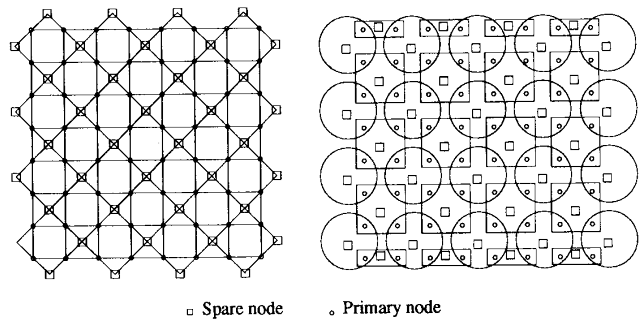
Figure 1 - A two-dimensional augmented logical mesh

1_*FT* sets, where each 1_*FT* set can handle exactly one failure. From a different perspective, a 1_*FT* set consists of some primary nodes and a spare which, if not faulty, can replace any faulty primary node in the 1_*FT* set. We consider a specific class of systems in which each node in X belongs to at most two 1_*FT* sets. Doubly fault tolerant systems fall into that class because if each primary node belongs to two 1_*FT* sets, and each spare node belongs to one 1_*FT* set, then any two faults in X can be tolerated. In the doubly fault tolerant system of Figure 1, each row of processors and each column of processors form a 1_*FT* set.

In addition to the 2-D augmented mesh shown in Figure 1, many fault tolerant arrays structures may be described by the model outlined above. For example, in the 50 percent redundancy interstitial structure introduced by Singh [14], spares are added in interstitial sites of an array (see Figure 2(a)). Each spare can cover for any one of four primary nodes and each primary node can be replaced by at the most two spares. Therefore, each 1_*FT* set consists of a spare node and at most four primary nodes as shown in Figure 2(b).

Given a set of failed nodes $F = \{f_1, f_2, \ldots, f_m\}$, which is a subset of X, the fault-to-spare assignment problem is to assign each faulty primary node to a non-faulty spare. This is equivalent to assigning each node f in F to a unique 1_*FT* set. Note that if f is a faulty spare, then it is assigned to the 1_*FT* set that contains it, and thus no other node can be assigned to that 1_*FT* set. This means that no faulty node will be assigned to a faulty spare.

For each set, 1_FT_i, we define a set, π_i, which contains all the faulty nodes of 1_FT_i. We also define the superset Π_F to include all the 1_*FT* sets that contain a faulty node. That is, $\Pi_F = \{1\_FT_l \mid \pi_l \neq \varnothing\}$ is the set of all 1_*FT* sets that are affected by the failures. With this, an assignment a_q may be defined as an injection which assigns to each node $f \in F$, a 1_*FT* set, $a_q(f) \in \Pi_F$. Clearly, the assigned 1_*FT* set should contain the faulty node. That is $f \in a_q(f)$. Note that for a given F and Π_F, there may not be any assignment or there may be more than one assignment.



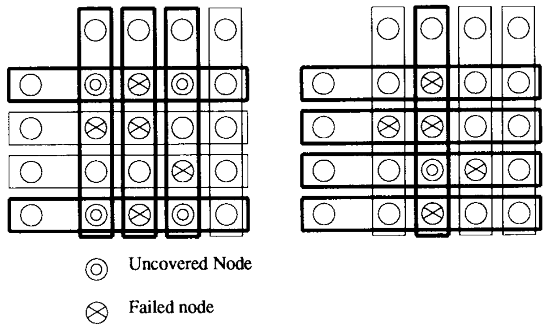
(a) The architecture (b) The 1_FT sets

Figure 2 - The Interstitial redundancy approach

In the cases where more than one assignment exists, it is desirable to find the one which optimizes some performance criteria. In [10], it is argued that real time systems alternate between strict phases and relaxed phases. When a fault occurs during a strict phase, local reconfiguration is necessary due to severe constraints on response time. Hence, during a relaxed phase, the faults in F may be assigned to 1_FT sets in a way that allows for local reconfiguration in subsequent strict phases. Local reconfiguration is guaranteed after any future fault if for each non-faulty primary node, x, in X, at least one of the two 1_FT sets containing x is available to cover for x. Here, a 1_FT set is not available to cover for x if that set is already assigned to some fault in F. If both 1_FT sets containing x are not available, then x is called ***uncovered*** in the sense that any future fault in x cannot be covered locally. In other words, if an uncovered node x becomes faulty, extensive reallocation of faults to spares may be needed to free-up a spare in one of the two 1_FT sets containing x.

For a given F, there may not be an assignment which leaves all the non-faulty primary nodes in X covered. In this case, it is shown in [10] that the expected life time of the system improves substantially if the number of uncovered nodes is minimized. We will call an assignment that minimizes the number of uncovered nodes, an "optimal assignment". For example, in Figure 3, we show (in bold) the 1_FT sets used in two possible assignment for the given 5 faults. We will use (i,j), $0 \le i,j \le 4$, to denote the node at row i and column j. In the assignment of Figure 3(a), faults (1,2) and (4,2) are assigned to the spare in their rows and faults (2,1), (2,2) and (3,3) are assigned to the spares in their columns, resulting in 4 uncovered nodes. In the optimal assignment of Figure 3(b), faults (1,2), (2,1), (3,3) and (4,2) are assigned to the spares in their rows and fault (3,3) is assigned to the spare in its column, resulting in only one uncovered node.

It has been recognized that finding a fault-to-spare assignment is equivalent to solving a bipartite matching problem [4,9]. This requires an $O(n^{1.5})$ time complexity, where n is the number of spares in the system. In the next section, we describe a solution to the assignment problem which is linear in the number of faults, and thus, linear in number of spares. This solution is based on partitioning the problem into smaller independent sub-problems. This partitioning is also used to find, in linear time, the optimal assignment for a subclass of problems that includes the 2-D augmented mesh of Figure 1. The algorithm applied in [10] to find an optimal assignment for this problem uses exhaustive search techniques.



(a) Four uncovered nodes (b) One uncovered node

Figure 3 - Minimizing the number of uncovered nodes

Region Perspective.

In order to find an assignment $a : F \rightarrow \Pi_F$, we may partition F into subsets, η_i, $i=1,...,z$ such that $F = \eta_1 \cup \cdots \cup \eta_z$, and accordingly $\Pi_F = R_1 \cup \cdots \cup R_z$, where $R_i = \{1\_FT_l : \pi_l \cap \eta_i \neq \emptyset\}$. That is R_i contains the 1_FT sets that are affected by the faults in η_i. The partitioning is done such that finding an assignment from η_i to R_i is independent of finding an assignment from η_j to R_j, if $i \neq j$. In order to define regions formally, we define a predicate $\delta : (1\_FT_i, 1\_FT_j) \rightarrow \{true, false\}$, such that,

$$\delta(1\_FT_i, 1\_FT_j) = \begin{cases} true & \text{if } (\pi_i \cap \pi_j) \neq \emptyset, \\ true & \text{if } \delta(1\_FT_i, 1\_FT_k) = \delta(1\_FT_j, 1\_FT_k) = true, \quad \text{for } some\ k \\ false & otherwise \end{cases}$$

For any two 1_FT sets, this predicate is defined recursively. The δ of any two sets, $1\_FT'$ and $1\_FT''$ is true if either 1) there is a fault at the intersection of the two sets or 2) if there exists a sequence of sets $1\_FT_l, 1\_FT_{l+1}, 1\_FT_{l+2}, \ldots, 1\_FT_p$, such that $\pi' \cap \pi_l \neq \emptyset$, $\pi_l \cap \pi_{l+1} \neq \emptyset, \ldots, \pi_{p-1} \cap \pi_p \neq \emptyset$, and $\pi_p \cap \pi'' \neq \emptyset$. The proof of the following lemma follows directly from the above definition of δ.

Lemma 1 : Given a set of failed nodes F,
IF Π_F is partitioned into *regions*, R_i, $i=1,...,z$, such that,
$1\_FT_l, 1\_FT_j \in R_i$ iff $\delta(1\_FT_l, 1\_FT_j)=true$,
THEN finding an assignment in each region is independent of the assignment in the other regions. □

The regions can be built on the basis that any two 1_FT sets that intersect at a failed node belong to the same region. Using this property, we start with one 1_FT set which has at least one failed node. Then, all the other 1_FT sets that intersect with the first set at failed nodes are added to the region. This procedure is propagated to the newly added 1_FT sets. When a region cannot grow any more, the next unused 1_FT set is taken as the nucleus for the next region. In the following implementation of this algorithm, each fault $f \in F$ is given a tag which initially indicates the sets that the node belongs to. That is, $tag(f)=\{i,j\}$ if f is in both

1_FT_i and 1_FT_j. If f is in only one set, 1_FT_i, then $tag(f)=\{i\}$. Each node with $|tag(f)|=2$ is visited twice to process its two 1_FT sets. A set, *visited*, is used to keep track of the nodes that have been visited once, but yet is to be visited a second time.

Algorithm 1: (Input: F, Π_F - **Output:** all the regions, R_k, $k=1,2,\cdots$)

k = 0 ; *visited* = $\emptyset$;
While $\Pi_F \neq \emptyset$ do

1) k = k + 1 ;
2) Get the next 1_FT_i from Π_F, remove it from Π_F and add it to R_k ;
3) For each node f_u in π_i, if $tag(f_u)=\{i,j\}$, then set $tag(f_u)=\{j\}$, and add f_u to *visited* ;
4) While *visited* $\neq \emptyset$ do
   - 4.1) Get and remove the next node f_u from *visited*; Assume $tag(f_u)=\{j\}$.
   - 4.2) If $1\_FT_j \in \Pi_F$, then
     - 4.2.1) Remove 1_FT_j from Π_F and add it to R_k ;
     - 4.2.2) For each node f_v in π_j, if $tag(f_v)=\{l,j\}$, then set $tag(f_v)=\{l\}$, and add f_v to *visited* ;

In step 4), each failed node is examined exactly once in the *visited* set. Thus, the algorithm runs in time $O(|F|)$. The regions resulting from the application of the above algorithm can be classified according to the ratio of the number of 1_FT sets to the number of failed nodes in the region. This is described by the following lemma.

Lemma 2: Given a region R_k

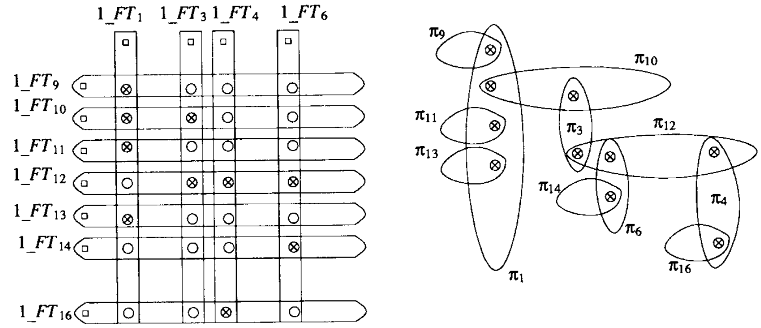
1) If $|\eta_k| > |R_k|$, then, there is no possible assignment for that region,
2) If $|\eta_k| = |R_k|$, any assignment is a one-to-one total function and uses all the 1_FT sets.
3) If $|\eta_k| < |R_k|$, then, $|R_k| = |\eta_k| + 1$. That is, there is exactly one more 1_FT set than failed nodes.

Proof: If the maximum number of 1_FT sets available, $|R_k|$, is less than the number of failures, $|\eta_k|$, then it is obvious that not all faults can be covered. Thus, there is no assignment. The result for $\eta_k = |R_k|$ is also obvious. The result for $|\eta_k| < |R_k|$ will be proved by induction on the number of failures.

If $|\eta_k| = 1$, then by the property of doubly fault tolerant systems, the fault in η_k may be in at most two 1_FT sets. Hence, if $|\eta_k| < |R_k|$, then $|R_k| = 2$. Next, let us assume that the Lemma is true for $|\eta_k| = m$ faults. In other words, $|R_k|$ is $m+1$. We will prove that the lemma is satisfied if we add another fault, f_{m+1}, to this region. We will refer to this augmented region as R_k^+ and to its fault set as η_k^+.

The node f_{m+1} can be in at most two 1_FT sets. However, for f_{m+1} to be in the same region as the faults in η_k, it has to share one of these sets with a fault in η_k. Thus, R_k^+ may have at most one 1_FT set that is not in R_k. That is, $|R_k^+| = |R_k| + 1 = m+2 = |\eta_k^+| + 1$. □

In Figure 4, we illustrate a region in which $|R| > |\eta|$; In Figure 4(a) we show a set of faulty nodes in a 2-D augmented mesh, and in Figure 4(b), we show the π sets for that region. Recall that a π set contains the faulty nodes in a 1_FT set.



(a) The faulty nodes (b) The π sets

Figure 4 - A region with $|R| > |\eta|$.

Finding an Assignment in a Region.

Given a region, R_k and the corresponding set of faults η_k, the assignment algorithm is based on the successive reduction of the region size. Nodes are assigned to the appropriate 1_FT sets one by one. After a node $f \in \eta_k$ is assigned to a 1_FT set from R_k, f is removed from η_k and 1_FT is removed from R_k. Two types of assignments of an individual node are defined:

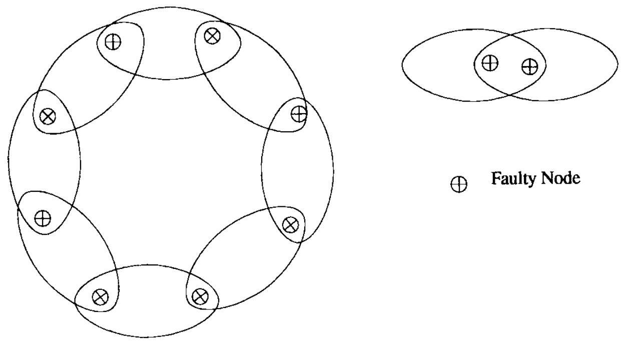
1) If a node, f, belongs to only one 1_FT set in R_k, then the assignment of f to that 1_FT set is called a **mandatory assignment**.
2) If a node, f, belongs to two sets, 1_FT_1 and 1_FT_2 in R_k and one of the two sets, say 1_FT_1, has no other faults beside f (that is $|\pi_1|=1$), then the assignment of f to 1_FT_1 is called a **greedy assignment**.

We first consider assignments for regions with $|\eta_k|=|R_k|$. If we apply the greedy assignment repetitively in such a region, then each time we assign one fault to a 1_FT set, we reduce the size of both R_k and η_k by one. This process may terminate, thus giving an assignment for the entire region, or may be blocked because every node belongs to two 1_FT sets and neither of them satisfies the requirement for a greedy assignment. This case is formally defined next.

Definition: A **locked region** is a region R_k such that for every $f \in \eta_k$, there are two sets, 1_FT and $1\_FT'$ in R_k, that contain f. That is, $f \in 1\_FT \cap 1\_FT'$. □

If may be shown that if a region R_k with $|R_k|=|\eta_k|$ is locked, then $|\pi_l|=2$ for every $1\_FT_l \in R_k$. (This may happen only if the π sets form a cycle or if $R_k=\{1\_FT, 1\_FT'\}$ and π intersects π' at exactly two nodes - see Figure 5). If faced with a locked region, the assignment may proceed by forcing an arbitrary assignment. That is, chosing any fault f from η_k and assigning f to one of the two 1_FT sets that contains f. This will also reduce the sizes of R_k and η_k by one, and will allow the assignment process to continue. In fact, it may be proved that once an assignment is forced, the process will not encounter another locked region. This observation, however, is not crucial to the assignment algorithm discussed next, and thus will not be proved here.

In order to find an assignment for a region with $|R_k| = |\eta_k|$, first, all greedy assignments are made. When no more greedy assignments are possible, either all the nodes in the region are



(a) a cycle (b) $|\pi \cap \pi'| = 2$

Figure 5 - The two types of locked regions

assigned or the region is reduced to a locked region. In the latter case, an arbitrary node is assigned to one of the 1_*FT* sets that the node belongs to. Due to this **forced assignment**, one of the sets in R_k will contain only one unassigned fault. This either allows for the resumption of the greedy assignment (in case of cycles) or reduces the region to one 1_*FT* set and one fault. In the latter case, a mandatory assignment will complete the solution. The algorithm uses a set, *greedy* which initially contains the failed nodes in η_k that satisfy: $tag(f_p)=\{i,j\}$ and either $|\pi_i|=1$ or $|\pi_j|=1$.

Algorithm 2: **Input:** R_k and η_k with $|R_k| = |\eta_k|$,
Output: an assignment of the nodes in η_k.

While $\eta_k \neq \emptyset$ do

1) While *greedy* $\neq \emptyset$ do
   - 1.1) Get the next node, f_p, from *greedy* ; Let $tag(f_p) = \{i,j\}$.
   - 1.2) Chose l in $\{i,j\}$ such that $|\pi_l|=1$ and let $\lambda = \{i,j\} - \{l\}$;
   - 1.3) Assign f_p to 1_FT_l and remove f_p from *greedy*, η_k, π_l and π_λ ;
   - 1.4) If $\pi_\lambda = \{f_q\}$, then
     - 1.4.1) if $tag(f_q) = \{u,\lambda\}$, then add f_q to *greedy*
     - 1.4.2) if $tag(f_q) = \{\lambda\}$, then assign f_q to 1_FT_λ, /* mandatory assignment */ and remove f_q from *greedy*, η_k and π_λ.
2) If $\eta_k \neq \emptyset$, then /* we have reached a locked region */
   - 2.1) Get a node f_p from η_k ; Let $tag(f_p) = \{i,j\}$ and $\pi_i = \{f_p f_q\}$.
   - 2.2) Assign f_p to 1_FT_i, /* forced assignment */ and remove f_p from η_k, π_j and π_i ;
   - 2.3) If $tag(f_q) = \{i,l\}$, then set $tag(f_q) = \{l\}$.
   - 2.4) If $|\eta_k| = 1$, then assign f_q to π_l and remove f_q from η_k, else, add the node in π_j to *greedy* /* $|\pi_j|=1$ */

Since the algorithm visits every node in the *greedy* set once, and each failed node in the region becomes a member of *greedy* at most once, this algorithm has $O(|\eta_k|)$ running time.

Now we examine the case where the number of 1_*FT* sets is more than the number of faults in the region. Since we have already proved that there is exactly one more 1_*FT* set than are needed in such a region, this implies that one of the 1_*FT* sets will be unused in the actual assignment. Hence, we may arbitrarily remove one 1_*FT* set from R_k, thus reducing the size of R_k such that $|R_k| = |\eta_k|$. However, removing one 1_*FT* set from R_k may either leave R_k as one region, or may partition it into two regions. In either cases, the problem of dealing with a region with $|R_k| = |\eta_k|+1$ is reduced to dealing with regions with $|R_k| = |\eta_k|$. Following is the algorithm.

Algorithm 3: **Input:** R_k and η_k, with $|R_k| = |\eta_k| + 1$,
Output: An assignment of the nodes in η_k.

1) Pick any set 1_FT_i from R_k ;
2) If $\pi_i = \{f_p\}$, then
   2.1) If $tag(f_p) = \{i, j\}$, then set $tag(f_p) = \{j\}$;
   2.2) Apply Algorithm 2 to $\{R_k\} - \{1\_FT_i\}$;
3) If $\pi_i = \{f_p, f_q\}$, then
   3.1) Form the two regions R_{k1} and R_{k2} such that $R_{k1} \cup R_{k2} \cup \{\pi_i\} = R_k$;
   3.2) If $tag(f_p)=\{i, j\}$, and $tag(f_q)=\{i, l\}$, then set $tag(f_p)=\{j\}$ and $tag(f_q)=\{l\}$;
   3.3) Apply Algorithm 2 to R_{k1} and R_{k2} ;

The algorithm runs in $O(|\eta_k|)$ time since it depends on algorithm 2. Figure 6 shows how the region of Figure 4 is partitioned into two regions after removing 1_FT_3.

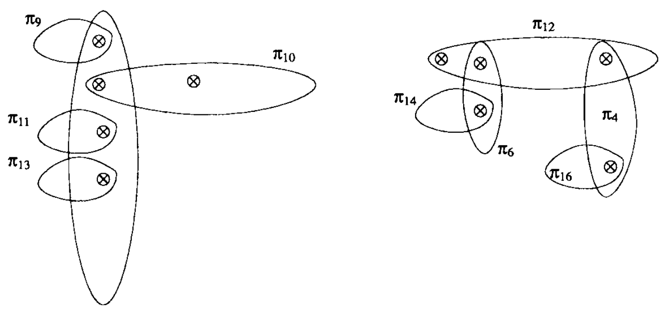


Figure 6 - partitioning the region of Figure 4 into two regions with $|R|=|\eta|$.

Hence, given a set of faults F, an assignment may be found by applying the following steps:

1) Build Π_F from F; Apply Algorithm 1 to Π_F to form regions;
2) For each region R_k
   a) if $|R_k| < |\eta_k|$ then declare "no assignment" and exit;
   b) if $|R_k| = |\eta_k|$ then apply Algorithm 2;
   c) if $|R_k| > |\eta_k|$ then apply Algorithm 3;

Since each step runs in linear time with respect to the number of failures, the assignment algorithm also runs in $O(|F|)$ time.

Optimal Assignment

In this section, we consider the problem of finding an assignment which minimizes the number of uncovered nodes as defined in Section 1. Assuming that Π_F is partitioned into regions, then by Lemma 1, the assignment in different regions may be found independently. In general, however, finding the optimal assignment in each region does not lead to an optimal assignment for the entire problem.

The general problem of finding an optimal assignment in a doubly fault tolerant system may be shown to be NP-hard by reducing it to the problem of finding a truth assignment which maximizes the number of true statements in a 2-SAT problem. We will not go into the details of that proof in this paper. We will, however, show that for a specific instance of doubly fault tolerant systems, the optimal assignment may be found in linear time. This instance incorporates the 2-dimensional augmented system (Figure 1) and is formally described next.

Definition: A doubly fault tolerant system is called 2-partitionable if its 1_FT sets may be categorized into two types, A and B such that:

> Any two sets $1\_FT_i^A$ and $1\_FT_j^A$ of type A have an empty intersection, and any two sets $1\_FT_i^B$ and $1\_FT_j^B$ of type B have an empty intersection. □

We will use a superscript to denote the type of a 1_FT set. Both systems shown in Figures 1 and 2 are 2-partitionable. In Figure 1, the vertical sets may be classified as type A and the horizontal sets as type B. In Figure 2(b), the sets enclosed in circles may be classified as type A and those enclosed in squares as type B.

Definition: A 2-partitionable system exhibits the regular area property if each 1_FT set of a certain type intersects with all the 1_FT sets of the other type. □

Note that the system of Figure 1 exhibits the Regular Area Property while the system of Figure 2 does not. This property will allow us to establish Lemma 3 which identifies optimal assignments.

Given a set of faults, F, in a 2-partitionable system, the corresponding set Π_F may be partitioned into the two disjoint sets $\Pi_F^A = \{1\_FT_i^A \mid \pi_i^A \neq \emptyset\}$ and $\Pi_F^B = \{1\_FT_j^B \mid \pi_j^B \neq \emptyset\}$. With this, an assignment, a_r, assigns each fault to either an 1_FT^A set from Π_F^A, or to a 1_FT^B set from Π_F^B. Let α_r and β_r be the subsets of Π_F^A and Π_F^B, respectively, used in the assignment a_r. Note that $|\alpha_r| + |\beta_r|$ equals the total number of faults.

The critical area, A_r, of an assignment a_r is defined as the set of nodes at the intersection of the 1_FT^A sets belonging to α_r and the 1_FT^B sets belonging to β_r. Hence, the number of nodes in the critical area, A_r, is given by $|\alpha_r| \times |\beta_r|$. A non-faulty node, ν, that belongs to A_r is uncovered if it lies at the intersection of a 1_FT^A set and a 1_FT^B set that are used to cover for faults. Thus no 1_FT set is available to cover for a possible future failure of ν. Note that if a non-faulty node, μ, is not in A_r, then at most one of the two 1_FT sets containing μ is used in a_r, and thus the second is available to cover for a possible future failure of μ.

It is our goal to find the assignment that minimizes the number of uncovered node. In order to formalize this notion, we define the following:

$\Phi(A_r)$ = number of failed nodes in A_r

$\hat{\Phi}(A_r) = (|\alpha_r| \times |\beta_r|) - \Phi(A_r)$ = number of uncovered nodes in A_r.

For example, the numbers of nodes in the critical areas for the two assignments in Figures 3(a) and 3(b) are 6 and 4, respectively. For Figure 3(a) $\Phi(A)=2$, and $\hat{\Phi}(A)=4$, and for Figure 3(b), $\Phi(A)=3$, and $\hat{\Phi}(A)=1$.

It is possible to find the assignment a_p which minimizes $\hat{\Phi}(A_p)$ by decreasing the size of A_p and increasing $\Phi(A_p)$. The following lemma provides a means for chosing α_p and β_p such that $\hat{\Phi}(A_p)$ is minimized.

Lemma 3: Let F be a fault set in a system that satisfies the Regular Area Property. If $\Pi_F^B \geq \Pi_F^A$, and a given assignment a_p satisfies the following:

1) $\beta_p = \Pi_F^B$
2) $\Phi(A_p) \geq \Phi(A_q)$ for any other assignment a_q with $\beta_q = \Pi_F^B$,

then a_p is an optimal assignment in the sense that it minimizes $\hat{\Phi}(A_p)$ over all assignments.

Proof: Let $I = |\Pi_F^A|$, $J = |\Pi_F^B|$ and $f = |F|$. Moreover, let $J = f - k$ for some $k \geq 0$, and thus $f - k \geq I$. We will prove the lemma by contradiction. First, consider the assignment a_p. Given that $|\beta_p| = J = f - k$ and that $|\alpha_p| + |\beta_p| = f$, the total number of nodes in the critical area of that assignment is

$$|A_p| = |\alpha_p| \times |\beta_p| = k\,(f-k)$$

To compute the number of faulty nodes in A_p observe that every faulty node in A_p should be in some 1_FT^A set that belongs to α_p. In other words, $\Phi(A_p)$ is equal to the total number of faults in all the 1_FT^A sets belonging to α_p. Given that $|\Pi_F^A| = I \leq f-k$ and that each 1_FT^A set in Π_F^A contains at least one of the f faults, then there is a subset τ of Π_F^A such that the total number of faults in the 1_FT^A sets belonging to τ is at least $2k$. This implies that

$$\Phi(A_p) \geq 2k$$

because, otherwise, we can construct an assignment a_q in which $\beta_q = \Pi_F^B$, $\alpha_q = \tau$ and $\Phi(A_q) \geq 2k$, thus contradicting the hypothesis of the lemma. From the definitions of $\hat{\Phi}(A_p)$ we then get:

$$\hat{\Phi}(A_p) = |A_p| - \Phi(A_p) \leq k\,(f-k-2)$$

Now, consider any other assignment a_r. If $\beta_r = \Pi_F^B$, then from the choice of a_p, $\Phi(A_p) \geq \Phi(A_r)$, and thus $\hat{\Phi}(A_p) \leq \hat{\Phi}(A_r)$. Hence, we assume that $\beta_r \subset \Pi_F^B$. For such an assignment, $\beta_r = f-k-u$ and $\alpha_r = k+u$ for some $0 < u \leq I-k$. Thus the total number of nodes in the critical area of the assignment is

$$|A_r| = (f-k-u)\,(k+u)$$

If a set $1\_FT_i^A$ and a set $1\_FT_j^B$ intersect at a faulty node, then a_r should assign this fault to one of these two sets. Hence, either $1\_FT_i^A$ is in α_r or $1\_FT_j^B$ is in β_r. This implies that if $1\_FT_i^A$ is in Π_F^A but not in α_r and $1\_FT_j^B$ is in Π_F^B but not in β_r, then $1\_FT_i^A$ and $1\_FT_j^B$ intersect at a non-faulty node. However, each 1_FT^A set that is in Π_F^A should contain a faulty node, hence, the 1_FT^A sets in $\Pi_F^A - \alpha_r$ contain at least $|\Pi_F^A - \alpha_r| = I-k-u$ faulty nodes. Similarly, the 1_FT^B sets in $\Pi_F^B - \beta_r$ contain at least $|\Pi_F^B - \beta_r| = u$ faulty nodes. This means that at least $(I-k-u)+u = I-k$ faults are not in the critical area, A_r, of a_r. Thus, at most $f-(I-k)$ faults are in A_r. That is,

$$\Phi(A_r) \leq f - I + k$$

and thus, the number of non faulty nodes in A_r is bounded by:

$$\hat{\Phi}(A_r) \geq |A_r| - (f-l+k) = (f-k-u)(k+u) - (f-l+k)$$
$$\geq (f-k-u)(k+u) - f + u = k\ (f-k-2) + (f-2k-u)\ (u-1)$$

Given that $u \leq l-k \leq (f-k)-k = f-2k$, we have

$$\hat{\Phi}(A_r) \geq \hat{\Phi}(A_p)$$

which proves the lemma. □

Lemma 3 provides the means of chosing the 1_FT sets in a global sense. However, its real value is providing a means for optimizing the global assignment by optimizing the assignment in individual regions, as described next.

Lemma 4: For a system that exhibits the Regular Area Property, an assignment minimizes the number of uncovered nodes in the system if it minimizes the number of uncovered nodes in each region.

Proof: According to Lemma 3, all 1_FT sets of one type, say A, are going to be used in the optimal assignment. Thus, the 1_FT sets which are not going to be used have to be of the other type, B. Hence, the area of the optimal assignment, say a_p, is fixed, namely $\alpha_p\ \beta_p$. By the definition of regions, if a set 1_FT^A from α_p and a set 1_FT^B from β_p are from two different regions, then they intersect at a non faulty node (otherwise the two regions would form a single region). Hence, any faulty node in $\Phi(A_p)$ should lie at the intersection of two 1_FT sets from the same region. Consequently, by maximizing the number of faulty nodes at the intersection of 1_FT sets in each region, the total number of non-faulty (uncovered) nodes in A_p is minimized. □

We can combine the result of Lemmas 3-4 with Algorithms 1-3 to design the algorithm for an optimal assignment of failed nodes. Note that in Algorithm 3, there is one step (step 1) in which a choice of NOT using a 1_FT set in a given region, R_k, has to be made. The type of this 1_FT set may be determined by comparing $|\Pi_F^A|$ with $|\Pi_F^B|$ as described in Lemma 3. Moreover, in order to minimize the number of non-faulty nodes at the intersection of the 1_FT sets in R_k, the set which is not used should have the minimum number of faults. With this choice, the sets that *are* used are the ones with the maximum number of faults. For example, in Figure 4, The minimum number of uncovered nodes is obtained when the set 1_FT_3 is not used in the assignment as shown in Figure 6.

The algorithm for optimal assignment is the same as the algorithm for any assignment except for two modifications. Two steps are added at the beginning of Algorithm 3 to determine which 1_FT set should not be used in the assignment of a given region, R_k with $|R_k| > |\eta_k|$. Specifically, step 1 of Algorithm 3 is replaced by:

1.1) If $|\Pi_F^A| > |\Pi_F^B|$ then $c = A$ else $c = B$.

1.2) Chose $1\_FT^c \in R_k$ such that $|\pi^c|$ is less than or equal to $|\pi_i^c|$ for any $1\_FT_i^c \in R_k$;

The above modifications results in an optimal assignment of all failures in doubly fault tolerant systems exhibiting the Regular Area Property.

Conclusion

The contribution of this paper is two-fold. First, it reduces the time for finding an assignment in doubly fault tolerant systems from $O(m^{1.5})$ to $O(m)$. When more than one assignment exists, the optimal assignment is specified to be the one which minimizes the number of uncovered nodes, and for a two dimensional array with an additional row and column of spares, that optimal assignment is found in linear time. This idea of choosing an assignment which enhances, in some sense, the future operation of the array may be applied to other optimization criteria such as the dilation between two nodes in the reconfigured array [2, 6], and the minimum degree of redundancy in the reconfigured system [10, 11].

Finally, we note that the work in this paper have been restricted to doubly fault tolerant systems. That is, systems in which each node may be replaced by one of two spares. Similar optimization may be defined for K-fault tolerant systems, for $K > 2$.

Acknowledgment

This work is, in part, supported under NSF grant MIP-8911303.

References

1. M. Alam and R. Melhem, "An Efficient Spare Allocation Scheme and its Application to Fault Tolerant Binary Hypercubes," *IEEE Trans. on Parallel and Distributed Systems*, vol. 2, no. 1, pp. 117-126.
2. M. Alam and R. Melhem, "Channel Multiplexing in Modular Fault Tolerant Multiprocessors," *Proc. of the International Comference on Parallel Processing*, pp. 492-496, 1991.
3. P. Banarjee, S. Kuo, and W. Fuch, "Reconfigurable Cube-Connected Cycles Architectures," *Proc. of the 16th International Symposium on Fault Tolerant Computing*, pp. 186-291, 1986.
4. K. Clarkson, "A Modification of the Greedy Algorithm for Vertex Cover," *Information Processing Letters*, vol. 16, pp. 23-25, 1983.
5. S. Dutt and J. Hayes, "An Automorphic Approach to the Design of Fault Tolerant Multiprocessors," *Proc. of the 19th International Symposium on Fault Tolerant Computing*, pp. 496-503, 1989.
6. B. Izadi and F. Ozguner, "Spare Allocation and Reconfiguration in a Fault Tolerant Hypercube with Direct Connect Capability," *Proc. of the 6th Distributed Memory Computing Conference*, pp. 711-714, 1991.
7. I. Koren and A. Singh, "Fault Tolerance in VLSI Circuits," *IEEE Comoputer*, vol. 23, no. 7, pp. 73-83.
8. S.Y. Kung, "Fabrication-time and Run-time Fault-tolerant Array Processors Using Single-track Switches," *Proceedings of the International Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 281-294, Oct. 1988.
9. S. Kuo and W. Fuchs, "Efficient Spare Allocation for Reconfigurable Arrays," *IEEE Design and Test*, vol. 4, no. 1, pp. 24-31.
10. R. Melhem, "Bi-level Reconfigurations of Fault Tolerant Arrays in Bi-modal Computational Environments," *Int. Symposium on Fault Tolerant Computing*, pp. 488-495, 1989.

(To appear in IEEE Trans. on Computers)

11. R. Melhem and J. Ramirez, ''Reconfiguration of Computational Arrays with Multiple Redundancy,'' *Proc. of the International Conference on Parallel Processing*, pp. 558-565, 1991.
12. R. Negrini, M. Sami, and R. Stefanelli, *Fault Tolerance Through Reconfiguration in VLSI and WSI Arrays*, The MIT Press.
13. M. Shean and J. Fortes, ''A Taxonomy of Reconfiguration Techniques for Fault Tolerant Processor Arrays,'' *IEEE Computer*, vol. 23, no. 1, pp. 55-69.
14. A. Singh, ''Interstitial Redundancy: An Area Efficient Fault Tolerance Scheme for Large Area VLSI Processor Arrays,'' *IEEE Trans. on Computers*, vol. 37, no. 11, pp. 1398-1410, 1988.