A general algorithm for detecting faults under the comparison diagnosis model

Iain A. Stewart

School of Engineering and Computing Sciences, Durham University,
Science Labs, South Road, Durham DH1 3LE, U.K.

Abstract

We develop a widely applicable algorithm to solve the fault diagnosis problem in certain distributed-memory multiprocessor systems in which there are a limited number of faulty processors. In particular, we prove that if the underlying graph $G = (V, E)$ forming the interconnection network has connectivity no less than its diagnosability δ and can be partitioned into enough connected components of large enough size then given a syndrome of test results under the comparison diagnosis model resulting from some set of faulty nodes of size at most δ, we can find the actual set of faulty nodes with time complexity $O(\Delta N)$, where Δ is the maximal degree of any node of the graph and N is the number of nodes.

1 Introduction

There has been a considerable amount of research undertaken as to designing interconnection networks with application to parallel computing. There can be no specific family of interconnection networks that is better than all of the others, for the quality of a family of interconnection networks depends upon the properties that happen to be of most relevance to a particular application. What is more, a whole range of properties of interconnection networks have proven to be of relevance to a whole range of applications. These properties include having low degree and high connectivity, being node- or edge-symmetric, having simple and efficient routing and broadcast algorithms, and possessing embedded Hamiltonian cycles or paths and cycles of a whole variety of lengths.

Not only should interconnection networks possess desirable properties such as those above but they should be able to tolerate a (limited) number of node or edge failures (more precisely, the machines whose processors are joined according to the interconnection network should be able to tolerate processor or link failures). This expectation has provoked much research on not just the sustainability of specific properties in the presence of faults but also the detection of the actual faults. It is with this latter research direction that we are concerned in this paper.

Imagine the situation. A distributed multiprocessor system is known to possess some faulty nodes but it is not known as to which nodes are faulty. The problem is to detect the

faulty nodes; that is, to *diagnose* the set of faulty nodes. Crucial to this diagnosis is the observation that we can use the nodes of the system to do this; that is, we can undertake a self-diagnosis. As to how this is done depends upon the model adopted. A popular model is the *comparison diagnosis model* (also called the *MM model*), advocated by Malek and Maeng [18,19]. In this model, a node can send a message to any two of its neighbours who then send replies back to the node. On receipt of these two replies, the node compares them and proclaims that at least one of the two neighbours is faulty if the replies are different or that both neighbours are fault-free if the replies are identical. However, if the node itself is faulty then no reliance can be placed on this proclamation. The goal is to use these tests made by various nodes in order to deduce exactly which are the faulty nodes.

Obviously there are limits as to what can be done under this model. For example, if all nodes are faulty then there is no way that this can be detected (from any collection of tests undertaken). For a specific interconnection network, there is a bound on the number of faulty nodes that can necessarily be detected within this model. Work has been undertaken on determining these bounds, or the *diagnosabilities*, for different interconnection networks. For example, in [23] it was proven that when $n \geq 5$, the n-dimensional hypercube has diagnosability n; so, if there are at most n faulty nodes then these nodes can be found given the set of all possible test results. Also, in [14] it was proven that when $n \geq 4$, the n-dimensional crossed cube has diagnosability n. In addition, in [28] it was proven that when $n \geq 4$, the n-dimensional star graph has diagnosability $n - 1$. However, an important, generic result (subsuming the above results) was obtained in [6] where it was proven that if an interconnection network is regular of degree n, has connectivity n and has at least $2n + 3$ nodes then it has diagnosability n.

Related to determining the diagnosability of a system is the fault diagnosis problem; that is, the problem of actually determining the faulty nodes, given a set of test results (assuming that one knows the diagnosability of the network and in the knowledge that the number of faults is bounded above by the diagnosability). In [20] an algorithm was given to determine the set of faulty nodes (in the circumstances described in the previous sentence) with time complexity $O(N^5)$, where N is the number of nodes in the network. The time complexity was recently improved in [26] to $O(d\Delta^3 N)$, where d (resp. Δ) is the minimum (resp. maximum) degree of any node in the network. In the particular case of an n-dimensional hypercube, it was shown in [27] that the fault diagnosis problem can be solved with time complexity $O(n^2 2^n)$.

However, a very recent paper of Chiang and Tan [8] developed a theory of node diagnosability. Here, nodes are individually diagnosed as to whether they are faulty or healthy. It turns out that under certain circumstances the health of a node can be deduced by examining a local neighbourhood of the node. In particular, it was shown in [8] that if every node is the root of an 'extended star structure' then a lower bound on its node diagnosability can be obtained and also that this extended star structure can be utilised to solve the fault diagnosis problem with time complexity $O(\Delta N)$. The general technique was illustrated on hypercubes and stars. (We shall discuss the content of the important papers [6,8,27] in more detail presently.)

In this paper, we establish a generic result that is applicable to a wide range of families of interconnection networks prevalent in parallel computing. In particular, we prove that under certain conditions (relating to the connectivity of the network and its intrinsic structure),

2

which hold within numerous interconnection networks, we can obtain algorithms for the fault diagnosis problem. The time complexity of all our algorithms is $O(\Delta N)$, where N is the number of nodes in the network and Δ is the maximal degree of any node. Our algorithms are either the fastest such algorithm known for the particular interconnection network or match the time complexity of the fastest known algorithm. The conditions under which we can apply our techniques are significantly less demanding than the conditions under which Chiang and Tan can apply their approach and, unlike Chiang and Tan, we do not require the whole of the syndrome table to be computed; that is, we dispense with the computation of redundant data. We illustrate the efficacy of our approach by applying it to hypercubes, crossed cubes, twisted cubes, folded hypercubes, enhanced hypercubes, augmented cubes, shuffle cubes, twisted N-cubes, k-ary n-cubes, augmented k-ary n-cubes, (n, k)-stars, stars, pancake graphs and arrangement graphs.

We present the comparison diagnosis model in more detail in the next section before we discuss the papers [8] and [27] in Section 3. In Section 4, we present our generic algorithm before we apply it in a wide range of interconnection networks in Section 5. We compare our algorithms with those resulting from applying the techniques of Chiang and Tan in Section 6 and also present directions for further research.

2 The comparison diagnosis model

In this section, we detail the basic definitions and the essential notions and results relating to the comparison diagnosis model. In actuality, we perform fault diagnosis for a distributed-memory multiprocessor whose processors are connected via an interconnection network and according to some topology. However, throughout we adopt a graph-theoretic terminology and abstract our multiprocessor as an undirected graph G consisting of nodes V and edges E (as opposed to processors and links).

The comparison diagnosis model is as follows. Given a graph $G = (V, E)$ within which there may be faulty nodes, every node u of V tests every pair v and w of its neighbours by sending a test message to both neighbours and receiving replies. We assume that: all faults are permanent; and a faulty node always produces an incorrect response to any test message, so that two faulty nodes do not produce identical responses to any test messages. Suppose that u is a healthy node (that is, it is not faulty). If the replies from v and w are identical then the test result $s_u(v, w)$ is set at 0 (signalling that both v and w are healthy), otherwise $s_u(v, w)$ is set at 1 (signalling that at least one of v and w is faulty). However, if u is a faulty node then the test result $s_u(v, w)$ can be arbitrarily 0 or 1 with no reliance placed upon this result. The set of all test results for every node and its pairs of neighbours is called a *syndrome*. The general fault diagnosis problem is: given a graph $G = (V, E)$ and a syndrome, can we use the data therein to obtain exactly the set of faulty nodes and, if so, to find these faulty nodes?

Note that the same syndrome could arise from different sets of faulty nodes; that is, there might be more than one set of faulty nodes *consistent* with the syndrome. A graph $G = (V, E)$ is said to be δ-*diagnosible* if given a syndrome s resulting from a set of at most δ faulty nodes, there is exactly one set of faulty nodes consistent with s. The maximum number δ for which a graph $G = (V, E)$ is δ-diagnosable is the *diagnosability* of G. Sengupta and

Dahbura [20] were the first to provide structural conditions upon G for it to be δ-diagnosible. One remark we have is that the diagnosability of any graph $G = (V, E)$ is bounded above by the minimal degree of any vertex of V. So see this, suppose that u is some vertex of minimal degree in G and consider the following two sets of faulty nodes: the first fault set consists of all u's neighbours; and the second of all u's neighbours as well as u. It is not difficult to see that there is a syndrome that both of these sets of faults are consistent with.

Henceforth, the fault diagnosis problem for a graph $G = (V, E)$ is defined as follows: given a syndrome for a set of faults $F \subseteq V$ of size at most the diagnosability of G, we require an algorithm that outputs exactly the set of faults F.

3 Related work

In this section we overview work related to solving the fault diagnosis problem. Of direct relevance to our research is the work of Yang [27] and Chiang and Tan [8].

The roots of our approach to determining the faulty nodes of a graph lie in [27] where an algorithm specific to hypercubes was developed. It is worthwhile reviewing this algorithm in order to introduce our techniques and also to point out some deficiencies of the algorithm (which we shall remedy). In [27], Yang utilized various cycle decompositions of the n-dimensional hypercube Q_n. If one fixes some components of the bit-strings of length n describing the nodes of Q_n then one obtains a subgraph isomorphic to Q_m, for some $m < n$, and it has long been known that Q_m is Hamiltonian. Furthermore, the set of 2^{n-m} (Hamiltonian) cycles resulting from varying the values of the chosen fixed components are 'well connected' to one another in that the cycles are connected in the 'shape' of the hypercube Q_{n-m}, via additional matchings; that is, by a set of node-disjoint edges where every node of the two cycles involved is incident with exactly one edge. (In Fig. 1 we illustrate 4 cycles, with 'dotted' edges, joined using matchings in the shape of a cycle of length 4.) Thus, Q_n is the union of a collection of node-disjoint cycles, interconnected in the shape of a hypercube. Yang observed that if the cycles are 'long enough' and 'plentiful enough' (that is, the number of components in the bit-strings of length n chosen to be fixed is not too large and not too small) then there must be enough cycles which can be deduced to consist entirely of healthy nodes and which can then be used to determine exactly where the faulty nodes lie within the rest of the hypercube. For example, let s be some syndrome. If some cycle is such that $s_x(y, z) = 0$ for every triple (y, x, z) of consecutive nodes on the cycle and the cycle has length greater than n (that is, the diagnosability of Q_n [23] and so an upper bound on the number of faulty nodes) then necessarily all nodes on this cycle must be healthy. Also, if this cycle is 'connected' to another cycle, by an additional matching, then the healthy cycle can be used to find the faulty nodes in the potentially faulty cycle, with this process subsequently iterated. What results is an algorithm for finding the faulty nodes in an n-dimensional hypercube that has time complexity $O(n^2 2^n)$.

However, as we show momentarily, the focus in [27] on decompositions into cycles is unnecessary and also adds an additional complexity-theoretic burden on the algorithm (this burden was not considered in [27]). As regards this latter point, note that in order to apply Yang's algorithm one needs to be able to actually construct Hamiltonian cycles in hypercubes. Whilst it is stated in [27] that such cycles can 'easily be constructed recursively', there is
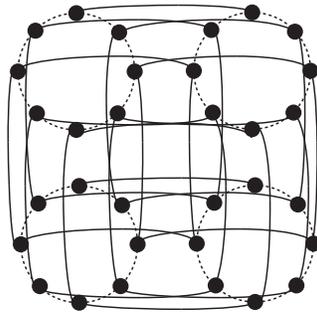
Fig. 1. Four cycles connected in the shape of a cycle.

more to it than this. The construction of cyclic Gray codes (for that is what an Hamiltonian cycle in a hypercube is) has been a longstanding topic of interest in computer science and discrete mathematics (see, for example, [4] for an account of the status of the problem some 30 years ago). There now exist numerous efficient algorithms for the generation of cyclic Gray codes (though note that the standard recursive algorithm is somewhat unsatisfactory in that it uses exponential space). Nevertheless, the actual computation of the Hamiltonian cycles is ignored in [27]. On the other hand, if the construction of the Hamiltonian cycles is to be done by a distributed algorithm implemented on the parallel machine whose interconnection network is the (faulty) hypercube then this becomes more problematic, for now it is not so straightforward to construct these cycles efficiently (that is, in time polynomial in the dimension of the hypercube) and when some of the processors may be faulty. The reader is referred to [21] and the references therein for the consideration of distributed algorithms to construct Hamiltonian cycles in faulty hypercubes. Of course, given that with our method we no longer need to rely on (Hamiltonian) cycle decompositions such as those utilized by Yang, this whole problem disappears with our approach and we obtain a widely-applicable algorithm (that is not restricted to just hypercubes). An additional drawback of Yang's algorithm is that when applied to Q_n its time complexity of $O(n^2 2^n)$ does not compare favourably with the algorithm due to Chiang and Tan [8], discussed in the next paragraph, which has time complexity $O(n 2^n)$.

In [8], Chiang and Tan adopt a different approach to solving the fault diagnosis problem. They develop an algorithm that is applied at each and every node in order to ascertain whether that particular node is healthy or faulty. Their ingenious algorithm only requires that syndrome tests involving nodes in a particular neighbourhood around the actual node x be studied, where this neighbourhood is an *extended star*, rooted at x, as illustrated in Fig. 2 (in this figure, only tests undertaken by the black nodes and involving only the nodes and edges of the extended star need be considered in order to deduce whether x is healthy or faulty). They show that if a graph is such that every node is the root of an extended star where there are n branches in this extended star then not only is the diagnosability of the graph at least n but there is an algorithm that solves the fault diagnosis problem that has time complexity $O(\Delta N)$, where N is the size of the input graph and Δ is the maximal degree of any node. Consequently, their algorithm is quite widely applicable. However, they make a crucial assumption that an extended star can be efficiently computed at any node and do not include the time or intellectual effort required to actually compute these extended stars in

5

their complexity analysis. They go on to illustrate methods for finding these extended stars in hypercubes and star graphs. In actuality, the additional time required to find extended stars in hypercubes and star graphs does not add to the time complexity (as it is subsumed by the 'big-O' notation) but, nevertheless, it will still be consumed in any computation. Also, it is not always clear as to how one constructs extended stars (or even whether they exist) in many other graphs prevalent as interconnection networks.
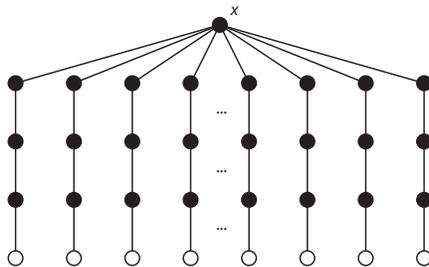

Fig. 2. An extended star rooted at x.

As regards diagnosability, it is worth comparing the results in [6] and [8]. The main result from [6] is that if a graph is regular of degree n, has connectivity n and has at least $2n + 3$ nodes then it has diagnosability n, whereas the main result (on diagnosability) from [8] is that a graph has diagnosability at least n if every node x is the root of an extended star with n branches. In fact, Chiang and Tan's result is a corollary of Chang, Lai, Tan and Hsu's for graphs that have connectivity n, are regular of degree n and for which for every node x there exists a node y such that the distance between x and y is at least 5 (as, by Menger's Theorem, there exist n node-disjoint paths joining any two nodes x and y with each of these paths of length at least 5). Both results are powerful and widely-applicable; indeed, all the graphs discussed in Section 5 satisfy the common hypothesis.

4 A general algorithm

Let $G = (V, E)$ be a connected graph of diagnosability $\delta \geq 1$ and let $F \subseteq V$ be a set of faulty nodes of size at most δ. Let s be a syndrome under the set of faulty nodes F. In what follows, if we write $s_x(y, z) = 0$ or $s_x(y, z) = 1$ then it is implicit that (x, y) and (x, z) are in E. In this section, we describe a general algorithmic procedure that solves the fault diagnosis problem for a wide variety of graphs G (that is, given a syndrome s, can be used to determine the faulty nodes of F so long as $|F|$ is no greater than the diagnosability of G). We go on to apply this procedure to a number of graph families prevalent as interconnection networks for parallel processing.

4.1 Looking for healthy components

We shall show how to modify Yang's approach so as to obtain a simpler, faster and more generally applicable algorithm to solve the fault diagnosis problem. We detail here our core algorithm.

We fix an ordering of the nodes of V. Also, fix $u_0 \in V$ and let s be some syndrome. We begin by initializing $U_0 = \{u_0\}$ and U_1 as

$$\{u_0\} \cup \{v \;\;:\;\; (u_0, v) \in E \text{ and there exists } w \in V \setminus \{v\} \text{ such}$$
$$\text{that } (u_0, w) \in E \text{ and } s_{u_0}(v, w) = 0\}.$$

For every $v \in U_1 \setminus \{u_0\}$, we set $t(v) = u_0$. We define U_i, for $i \geq 2$, iteratively as

$$U_{i-1} \cup \{v \;\;:\;\; v \notin U_{i-1} \text{ and } (u, v) \in E \text{ for some}$$
$$u \in U_{i-1} \setminus U_{i-2} \text{ with } s_u(v, t(u)) = 0\}.$$

For every $v \in U_i \setminus U_{i-1}$, where $i \geq 2$, we define $t(v)$ to be the least (w.r.t. our fixed ordering) node $u \in U_{i-1} \setminus U_{i-2}$ for which $s_u(v, t(u)) = 0$. We say that the nodes of the set $C_i = \{t(v) : v \in U_i \setminus U_{i-1}\}$ *contribute* to the construction of U_i, for $i \geq 1$. Note that no node contributes to the construction of both U_i and $U_{i'}$, where $i \neq i'$.

Let r be the least r such that $U_r = U_{r+1}$. The function $t : U_r \setminus \{u_0\} \rightarrow U_r$ describes a tree T with root u_0 via: if $t(v) = u$ then u is the parent of v in T. W.r.t. the tree T, the internal nodes are exactly those nodes which contribute to some U_i; in fact, the internal nodes at depth i in T (where the root has depth 0) constitute the set C_{i+1}. Note that if T does not consist of the single node u_0 then the root node u_0 has degree at least 2.

Clearly we have that if u_0 is healthy then all nodes of U_1 are healthy. In fact, a simple induction yields that if u_0 is healthy then all nodes of U_i are healthy, for every $i \geq 1$.

Alternatively, if some node u of some C_i is faulty (that is, some internal node of T) then: its parent $t(u)$ (if it exists) is faulty, as $s_{t(u)}(u, v) = 0$, for some node v (with $v = t(t(u))$, if $t(u)$ is not the root, and v equal to some child of u_0, otherwise); and all of its children v that are themselves internal nodes are faulty, as $s_v(w, t(v) = u) = 0$, for all children w of v. Consequently, if any internal node of T is faulty then so are all internal nodes of T.

The above observations can be allied to the fact that we have an upper bound on the number of faulty nodes in F. In particular, as we construct the U_i's, if ever the size of the set $C_1 \cup C_2 \cup \ldots \cup C_i$ becomes greater than δ then we know for sure that every node of T will be healthy.

The following algorithm $Set\_Builder(u_0)$, whose input is: a graph $G = (V, E)$, of diagnosability δ; a node $u_0 \in V$; and a syndrome s, implements the above discussion.

$Set\_Builder(u_0)$
```
set U_0 := {u_0}, all_healthy := false, finished := false and i := 1
while finished = false do
   build U_i
   if U_i = U_{i-1} then
      finished := true and i := i - 1
   else
      if |C_1 ∪ C_2 ∪ ... ∪ C_i| > d then
         all_healthy := true
      i := i + 1
return (all_healthy, U_i)
```

Let r be the value of the variable i on termination (and so $U_{r-1} \subset U_r = U_{r+1}$, if $r \geq 1$, and T is the tree associated with U_r). The truth of the variable *all_healthy* on termination signals that we have proven that all nodes in U_r are healthy. Of course, if *all_healthy* is *false* then the nodes of U_r might all be healthy but we cannot as yet say for sure.

Suppose that all nodes of U_r are, in fact, healthy, and that $r \geq 1$ (if $r = 0$ then $U_r = \{u_0\}$). Let N be the set of nodes adjacent to some node of U_r in G. If $x \in N$ then it is adjacent to some node $y \in U_r$ which in turn is adjacent to $z = t(y) \in U_r$, if $y \neq u_0$, or some child z of u_0 in T, if $y = u_0$. In both cases, $s_y(x, z) = 1$ as otherwise x would have been placed in some U_i and hence U_r. As y and z are healthy, we must have that x is faulty. Hence, all nodes of N are faulty. Also, either N forms an articulation set for G (that is, the removal of the nodes of N and their incident edges from G results in a disconnected graph) or $V = U_r \cup N$. We shall use this observation subsequently.

4.2 Time complexity

Consider the (sequential) time complexity of $Set\_Builder(u_0)$. Building U_1 takes $O(\Delta^2)$ time, where Δ is the maximal degree of any node in G (we assume an adjacency list representation of G). Building U_i, for $i \geq 2$, takes $O(\Delta p_{i-1})$ time, where $p_{i-1} = |U_{i-1} \setminus U_{i-2}|$. The first test in the body of the while-loop of $Set\_Builder$ takes constant time as does the second test, as counting the number of nodes in C_i can be built into the construction of U_i at no extra cost (note that we only need to count up to δ and that $\delta \leq \Delta$). Hence, if there are r iterations of the while-loop then $Set\_Builder(u_0)$ has time complexity $O(\Delta|U_r|)$. Note also that every time a node is added to U_i, either the node contributing to this addition is already adjacent to at least 2 nodes of U_i or it becomes adjacent to at least 2 nodes of U_i. Thus, if $Set\_Builder(u_0)$ terminates with *all_healthy* set at *false* then r is at most $\delta + 1$.

The discussion above immediately yields the following result which we shall apply in a variety of graphs in the next section.

Theorem 1 *Let $G = (V, E)$ be a connected graph of diagnosability $\delta \geq 1$ and connectivity $\kappa \geq \delta$, and where the maximal degree of any node in V is Δ. Suppose that the algorithm $Set\_Builder(u_0)$, for some $u_0 \in V$, takes as input a description of the graph G and a syndrome for some faulty set of nodes $F \subseteq V$ of size at most δ, and outputs the set of nodes U_r, where $|U_r| > 1$ and all nodes of U_r are healthy. The set N of nodes adjacent to a node of U_r is the set of faults F and the time taken by $Set\_Builder(u_0)$ is $O(\Delta|U_r|)$.*

Note that in order to apply the above theorem to solve the fault diagnosis problem in graphs whose connectivity is at least their diagnosability, all we need to do is to ensure that the number of internal nodes of the output tree T (whose node set is U_r) is greater than the diagosability.

5 Applications

In this section, we apply the general algorithm of the previous section to some specific families of graphs that have been studied as interconnection networks. We refer the reader to the specific references given for definitions of these graphs.

5.1 Hypercubes and their variants

Let Q_n be an n-dimensional hypercube, where $n \geq 7$. It is known that the diagnosability of Q_n is n [23]. Let F be a set of at most n faulty nodes in Q_n. Let m be the minimal integer such that $m > \log_2(n)$. Fixing the first $n - m$ components of the nodes of a hypercube at some tuple in $\mathbf{v} \in \{0,1\}^{n-m}$ results in a copy of Q_m, denoted $Q_m(\mathbf{v})$, within Q_n (we often denote the nodes of some 'dimensional' graph in bold type). Since $2^m > n$, $Q_m(\mathbf{v})$ has more than n nodes. Also, as $n \geq 7$, there are $2^{n-m} > n$ node-disjoint copies of Q_m within Q_n. Consequently, at least one of these copies contains no faulty nodes; call this copy $Q_m(\mathbf{w})$. If we start the algorithm $Set\_Builder$ at the node $u_0 = (\mathbf{w}, 0, 0, \ldots, 0)$ of Q_n then the resulting set of nodes U_r consists entirely of healthy nodes (as it must contain all 2^m nodes of $Q_m(\mathbf{w})$). The set of nodes N adjacent to U_r consists entirely of faulty nodes and is either an articulation set of Q_n or it consists of all faulty nodes in Q_n. However, as any articulation set of Q_n contains at least n nodes [24], we must have that N contains exactly the faulty nodes in Q_n.

The above presupposes that we can find the subgraph $Q_m(\mathbf{w})$ of Q_n within which there are no faulty nodes. In our search for this subgraph $Q_m(\mathbf{w})$, we need to ensure that we do not waste too much time exploring other subgraphs that may not be suitable. Given an arbitrary graph G, we denote by $Set\_Builder(u_0, H)$ the algorithm $Set\_Builder$ applied to G at the node u_0 of the subgraph H of G but where the adjacency relation is restricted to the subgraph H; that is, $Set\_Builder(u_0, H)$ starts from u_0 and only adds nodes of H to the sets it builds. This gives rise to the following algorithm.

$Faults\_in\_Hypercubes$
   $\mathbf{v} := 0^{n-m}$ and $\mathbf{u} := (\mathbf{v}, 0^m)$
   $(all\_healthy, U) := Set\_Builder(\mathbf{u}, Q_m(\mathbf{v}))$
   while $all\_healthy = false$ do
      $\mathbf{v} := next_{n-m}(\mathbf{v})$ and $\mathbf{u} := (\mathbf{v}, 0^m)$
      $(all\_healthy, U) := Set\_Builder(\mathbf{u}, Q_m(\mathbf{v}))$
   $(all\_healthy, U) := Set\_Builder(\mathbf{u})$
   $N :=$ set of neighbours of U
   output N

The function $next_{n-m}(\mathbf{v})$ delivers the next node after the node $\mathbf{v}$ of Q_{n-m} according to some fixed listing of all nodes. Note that there is no need for there to be edges between consecutive nodes; any listing of the nodes of Q_{n-m} will do. In fact, all we need is a list of $n + 1$ nodes starting from 0^{n-m} as executing $Set\_Builder$ starting from at least one of the $n+1$ resulting nodes will result in the provision of a set of healthy nodes as required.

As for the complexity of our algorithm $Faults\_in\_Hypercubes$, an execution of the algorithm $Set\_Builder(\mathbf{u}, Q_m(\mathbf{v}))$ takes $O(n2^m) = O(n^2)$ time. Thus, the time complexity of $Faults\_in\_Hypercubes$ is dominated by the final execution of $Set\_Builder$ and this execution takes $O(n2^n)$ time.

Theorem 2 *Let F be a set of at most n faulty nodes in an n-dimensional hypercube. There is an algorithm running in $O(n2^n)$ time that takes as input a syndrome for F and returns the actual set F of faulty nodes.*

Note that the key property of hypercubes that results in Theorem 2 is that the nodes of Q_n can be partitioned into 2 sets, depending upon the first component of the bit-string of length n naming a node, so that the induced subgraphs on these two sets of nodes are both isomorphic to Q_{n-1}. This results in an ability to partition Q_n into disjoint sets of nodes (the subgraphs $Q_m(\mathbf{v})$, above) so that a list of representative nodes in each set can be easily generated (the nodes $(\mathbf{v}, 0^m)$, above). Hypercubes are not alone in having such decompositions.

- For $n \geq 1$, the 2^n nodes of a crossed cube CQ_n can be partitioned into 2 sets, by fixing the first component in the bit-strings of length n naming the nodes at 0 and at 1, so that each set of nodes induces a copy of CQ_{n-1} [12]. Also, CQ_n is regular of degree n and has connectivity n [16]; thus, by [6], CQ_n has diagnosability n, when $n \geq 4$.

- For $n \geq 2$, the 2^n nodes of a twisted cube TQ_n can be partitioned into 2 sets, by fixing the first two components in the bit-strings of length n naming the nodes at either $(0, 0)$ or $(1, 0)$ and at either $(0, 1)$ or $(1, 1)$, so that each of these subsets induces a copy of TQ_{n-1} [15]. Also, TQ_n is regular of degree n and has connectivity n [7]; thus, by [6], TQ_n has diagnosability n, when $n \geq 4$.

- For $n \geq 1$, the folded hypercube FQ_n and the enhanced hypercube $Q_{n,m}$ contain the hypercube Q_n as a spanning subgraph, and are both regular of degree $n + 1$ and have connectivity $n + 1$ [3, 22]; thus, by [6], both FQ_n and $Q_{n,m}$ have diagnosability $n + 1$, when $n \geq 4$.

- For $n \geq 1$, the 2^n nodes of an augmented cube AQ_n can be partitioned into 2 sets, by fixing the first component in the bit-strings of length n naming the nodes at 0 and at 1, so that each of these subsets induces a copy of AQ_{n-1} [10]. Also, AQ_n is regular of degree $2n - 1$ and has connectivity $2n - 1$ [10]; thus, by [6], AQ_n has diagnosability $2n - 1$, when $n \geq 5$.

- For $n = 4k + 2$, with $k \geq 0$, the 2^n nodes of a shuffle-cube SQ_n can be partitioned into 16 sets, by fixing the first four components at some tuple from $\{0, 1\}^4$, so that each of these subsets induces a copy of SQ_{n-4} [17]. Also, SQ_n is regular of degree n and has connectivity n [17]; thus, by [6], SQ_n has diagnosability n, when $n \geq 4$.

- For $n \geq 2$, the 2^n nodes of a twisted N-cube TQ'_n can be partitioned into 2 sets, by fixing the first component in the bit-strings of length n naming the nodes at 0 and at 1, so that one of these sets induces a copy of Q_{n-1} and other induces a copy of TQ'_{n-1} [13]. Also, TQ'_n is regular of degree n and has connectivity n [13]; thus, by [6], TQ'_n has diagnosability n, when $n \geq 4$.

Consequently, be proceeding as we did for the hypercube, we immediately obtain the following result.

Theorem 3 *Let G be CQ_n, TQ_n, FQ_n, $Q_{n,m}$, AQ_n, SQ_n or TQ'_n and let F be a set of at most δ faulty nodes, where δ is the diagnosability of G. There is an algorithm running in $O(n2^n)$ time that takes as input a syndrome for F and returns the actual set F of faulty nodes.*

5.2 Other interconnection networks

Let $k \geq 3$ and $n \geq 2$, and let Q_n^k be a k-ary n-cube. However, suppose further that $(k, n) \notin \{(3, 2), (3, 3), (3, 4), (4, 2), (4, 3), (5, 2)\}$. By [6], the diagnosability of Q_n^k is $2n$. Let F be a set of at most $2n$ faulty nodes in Q_n^k. Let m be the minimal integer such that $k^m > 2n$; that is, such that $m > \log_k(2n)$. Fixing the first $n - m$ components of a k-ary n-cube at some tuple in $\mathbf{v} \in \{0, 1, \ldots, k - 1\}^{n-m}$ results in a copy of Q_m^k, denoted $Q_m^k(\mathbf{v})$, within Q_n^k. Since $k^m > 2n$, $Q_m^k(\mathbf{v})$ has more than $2n$ nodes. Also, because of our restrictions on k and n, there are $k^{n-m} > 2n$ node-disjoint copies of Q_m^k within Q_n^k. Consequently, at least one of these copies contains no faulty nodes; call this copy $Q_m^k(\mathbf{w})$. If we start the algorithm $Set\_Builder$ at the node $u_0 = (\mathbf{w}, 0, 0, \ldots, 0)$ of Q_n^k then the resulting set of nodes U_r consists entirely of healthy nodes (as it must contain all k^m nodes of $Q_m^k(\mathbf{w})$). The set of nodes N adjacent to U_r consists entirely of faulty nodes and is either an articulation set of Q_n^k or it consists of all faulty nodes in Q_n^k. However, as any articulation set of Q_n^k contains at least $2n$ nodes [5], we must have that N contains exactly the faulty nodes in Q_n^k. We can clearly construct an algorithm just as we did for hypercubes and obtain the following result.

Theorem 4 *Let F be a set of at most $2n$ faulty nodes in a k-ary n-cube. There is an algorithm running in $O(nk^n)$ time that takes a syndrome for F and returns the actual set F of faulty nodes.*

A graph called the augmented k-ary n-cube $AQ_{n,k}$ was recently defined in [25] and is an extension of the k-ary n-cube in a manner analogous to the extension of an n-dimensional hypercube to an n-dimensional augmented cube. It was proven in [25] that $AQ_{n,k}$ is regular of degree $4n - 2$ and has connectivity $4n - 2$. Thus, by [6], so long as $(n, k) \neq (2, 3)$, $AQ_{n,k}$ has diagnosability $4n - 2$. As $AQ_{n,k}$ contains a k-ary n-cube as a spanning subgraph, an immediate corollary of the above discussion is that there is an algorithm running in $O(nk^n)$ time that takes a syndrome for a set F of at most $4n - 2$ faulty nodes in $AQ_{n,k}$ and returns the actual set F of faulty nodes.

Let $n \geq 2$ and $2 \leq k \leq n - 1$, and let $S_{n,k}$ be the (n, k)-star graph. The (n, k)-star graph is regular of degree $n - 1$ and has connectivity $n - 1$ [9]. The (n, k)-star graph has $\frac{n!}{(n-k)!}$ nodes, and so if $(k, n) \neq (2, 3)$ then $S_{n,k}$ has diagnosability $n - 1$ [6]. Let F be a set of at most $n - 1$ faulty nodes in $S_{n,k}$. By fixing the kth component of the bit-strings naming the nodes of $S_{n,k}$, we obtain that $S_{n,k}$ is partitioned into n copies of $S_{n-1,k-1}$, each of which contains more than $n - 1$ nodes (as $k \neq 1$). Thus, at least one of these copies must consist entirely of healthy nodes. Hence, we can apply $Set\_Builder$ to obtain our set of faulty nodes, just as we have done above.

Theorem 5 *Let F be a set of at most $n - 1$ faulty nodes in an (n, k)-star $S_{n,k}$, where $2 \leq k \leq n - 1$. There is an algorithm running in $O(\frac{n!n}{(n-k)!})$ time that takes a syndrome for F and returns the actual set F of faulty nodes.*

Of course, $S_{n,n-1}$ is isomorphic to the star graph S_n [1] and so Theorem 5 applies to star graphs too. Note that if $k = 1$ then $S_{n,1}$ is a clique on n nodes.

Let $n \geq 2$ and let P_n be the n-dimensional pancake graph. The n-dimensional pancake graph is regular of degree $n - 1$ and has connectivity $n - 1$ [2]. The n-dimensional pancake

graph has $n!$ nodes, and so if $n \geq 4$ then P_n has diagnosability $n-1$ [6]. Let F be a set of at most $n-1$ faulty nodes in P_n. By fixing the kth component of the bit-strings naming the nodes of $S_{n,k}$, we obtain that $S_{n,k}$ is partitioned into n copies of $S_{n-1,k-1}$, each of which contains more than $n-1$ nodes (as $k \neq 1$). Thus, at least one of these copies must consist entirely of healthy nodes. Hence, we can apply $Set\_Builder$ to obtain our set of faulty nodes, just as we have done above.

Theorem 6 *Let F be a set of at most $n-1$ faulty nodes in a pancake graph P_n, where $n \geq 4$. There is an algorithm running in $O(n!n)$ time that takes a syndrome for F and returns the actual set F of faulty nodes.*

Let $n \geq 4$ and $2 \leq k \leq n$, and let $A_{n,k}$ be the arrangement graph [11]. The arrangement graph $A_{n,k}$ is regular of degree $k(n-k)$ and has connectivity $k(n-k)$ [11]. The arrangement graph $A_{n,k}$ has $\frac{n!}{(n-k)!}$ nodes and so has diagnosability $k(n-k)$ [6]. Let F be a set of at most $k(n-k)$ faulty nodes in $A_{n,k}$. By fixing some component of the bit-strings naming the nodes of P_n, we obtain that P_n is partitioned into n copies of P_{n-1}, each of which contains more than $n-1$ nodes. Thus, at least one of these copies must consist entirely of healthy nodes. Hence, we can apply $Set\_Builder$ to obtain our set of faulty nodes, just as we have done above.

Theorem 7 *Let F be a set of at most $n-1$ faulty nodes in an arrangement graph $A_{n,k}$, where $n \geq 4$ and $2 \leq k \leq n$. There is an algorithm running in $O(\frac{n!k(n-k)}{(n-k)!})$ time that takes a syndrome for F and returns the actual set F of faulty nodes.*

6   Conclusions and further research

As we have just demonstrated, we have developed a widely applicable technique that results in an algorithm to solve the fault diagnosis problem in a range of graphs used as interconnection networks in parallel computing. The time complexities of our algorithms match those of the algorithms due to Chiang and Tan. However, the conditions required for us to apply our algorithms are much less severe than the conditions required by Chiang and Tan. Apart from a relationship between the connectivity and the diagnosability of the graph in question, all that we need is that the input graph can be partitioned into enough sizeable connected subgraphs, whereas Chiang and Tan require not only that every node is the root of an extended star but also that this extended star can be actually computed, for every node of the graph. Indeed, as can be seen from [8], quite a bit of work still has to be done in the case of hypercubes and stars for Chiang and Tan's technique to be applied whereas, as we have demonstrated in this section, our technique can be easily applied to a range of graphs. Note that once we have satisfied ourselves that our condition holds for some graph, we are done; for unlike Chiang and Tan, we do not need to rely on any additional computational aspects of our condition in the subsequent computation of the faulty set of nodes. Note also that if the set of faulty nodes is not an articulation set then a by-product of our algorithm is that we obtain a tree spanning the set of healthy nodes of the graph which could possibly be utilised in some other context.

However, there are more improved aspects of our algorithms when compared with those of Chiang and Tan. We have hitherto assumed that all fault diagnosis algorithms are simply given the syndrome as input; that is, that the syndrome has already been obtained (presumably by every node testing pairs of its neighbours in the distributed-memory multiprocessor within which it lies). The cost of actually obtaining the syndrome has been ignored. It could well be that performing the actual tests is expensive and that we wish to minimize the number of tests performed. Any expense might be exacerbated depending upon the message-passing model adopted; for it might be that any node can only send one message at any time and thus that at least d time units are required in order for a node to send a message to each of its neighbours (with different nodes having to synchronize their messages to avoid conflicts). Alternatively, actually consulting the syndrome table could well be expensive, and so we may wish to minimize the number of test results needing to be read from the table.

We note that regardless of whether we wish to minimize the number of tests performed or the number of syndrome table look-ups, our algorithm $Set\_Builder(u_0)$ does not need to build or consult the whole of the syndrome table (in order to calculate U_r). We assume for simplicity that we are trying to minimize the number of syndrome table look-ups. The number of test results due to the node u_0 that need to be consulted is at most $\frac{\Delta(\Delta-1)}{2}$. We consult at most $\Delta - 1$ test results due to any node of $U_1 \setminus \{u_0\}$; we consult at most $\Delta - 1$ test results due to any node of $U_2 \setminus U_1$; and so on. Hence, the number of test results from the syndrome table needing to be consulted is $(\Delta - 1)(\frac{\Delta}{2} + |U_r| - 1)$, which is far less than the number of test results in the complete syndrome table. On the other hand, Chiang and Tan's algorithms need to consult all of the test results in the syndrome table. Indeed, if the whole of the syndrome table cannot be stored in memory then it is non-trivial to implement Chiang and Tan's algorithms so that the same test results do not need to be *repeatedly* consulted. The upshot is that our algorithms will consult markedly fewer test results from the syndrome table than the algorithms of Chiang and Tan as we minimise unnecessary consultations.

As a direction for further research, we suggest the following. The whole essence of finding the faults in a distributed multiprocessor system is that this should be undertaken through self-diagnosis; that is, the nodes of the system should collect the data from which the actual faulty nodes can be discovered. Hitherto, this discovery has been performed via a centralised sequential algorithm. For 'dimensional' networks such as hypercubes, crossed cubes, k-ary n-cubes, and so on, the resulting algorithm takes time exponential in the actual dimension. It would, of course, be preferable to have a time complexity that is polynomial in the actual dimension; however, in a centralized context this does not make sense given that there is generally an exponential (in the dimension) number of processors in the network. Surprisingly, given the focus on self-diagnosis, no attention has been paid to the *system itself* undertaking the computation enabling the discovery of the faulty nodes; that is, no attention has been paid to the distributed complexity of such a task. One might think that an immediate obstacle to the system itself finding the faulty nodes is that some of these nodes are themselves faulty. However, the diagnosis of faults in the context of models such as the comparison diagnosis model is that it is the nodes (that is, the main processors) where faults arise and not the relatively less complex communication links within the interconnection network nor the system which governs inter-node communication. That is, it is entirely realistic to assume that the communication network is intact and fault-free, and that the

13

communication system can perform (simple) computations in order to diagnose the faulty nodes. Indeed, when collecting the data to build a syndrome it is implicitly assumed that any processor can send a message to any of its neighbours and that the interconnection network is fault-free. It would be of practical interest to study the distributed complexity of solving the fault diagnosis problem in this setting and it is something we intend to pursue in future. Preliminary results show that a distributed implementation of our algorithm in hypercubes has a significantly improved time complexity when compared to a distributed implementation of Chiang and Tan's algorithm.

References

[1] S.B. Akers, D. Harel and B. Krishnamurthy, The star graph: an attractive alternative to the n-cube, in: *Proc. of Int. Conf. on Parallel Processing* (1987) 393–400.

[2] S.B. Akers and B. Krishnamurthy, A group-theoretic model for symmetric interconnection networks, *IEEE Transactions on Computers* 38 (1989) 555–566.

[3] A. Al-Amaway and S. Latifi, Properties and performance of folded hypercubes, *IEEE Transactions on Parallel and Distributed Systems* 2 (1991) 31–42.

[4] J.R. Bitner, G. Ehrlich and E.M. Reingold, Efficient generation of the binary reflected Gray code and its applications, *Communications of the ACM* 19 (1976) 517–521.

[5] B. Bose, B. Broeg, Y. Kwon and Y. Ashir, Lee distance and topological properties of k-ary n-cubes, *IEEE Transactions on Computers* 44 (1995) 1021–1030.

[6] C.-P. Chang, P.-L. Lai, J.J.-M. Tan and L.-H. Hsu, Diagnosability of t-connected networks and product networks under the comparison diagnosis model, *IEEE Transactions on Computers* 53 (2004) 1582–1590.

[7] C.-P. Chang, J.-N. Wang and L.-H. Hsu, Topological properties of twisted cube, *Information Sciences* 113 (1999) 147–167.

[8] C.-F. Chiang and J.J.-M. Tan, Using node diagnosability to determine t-diagnosability under the comparison diagnosis model, *IEEE Transactions on Computers* 58 (2009) 251–259.

[9] W.-K. Chiang and R.-J. Chen, Topological properties of the (n, k)-star graph, *International Journal of Foundations of Computer Science* 9 (1998) 235–248.

[10] S.A. Choudum and V. Sunitha, Augmented cubes, *Networks* 40 (2002) 71–84.

[11] K. Day and A. Tripathi, Arrangement graphs: a class of generalized star graphs, *Information Processing Letters* 42 (1992) 235–241.

[12] K. Efe, P.K. Blackwell, W. Slough and T. Shiau, Topological properties of the crossed cube architecture, *Parallel Computing* 20 (1994) 1763–1775.

[13] A.-H. Esfahanian, L.M. Ni and B.E. Sagan, The twisted N-cube with application to multiprocessing, *IEEE Transactions on Computers* 40 (1991) 88–93.

[14] J. Fan, Diagnosability of crossed cubes under the comparison diagnosis model, *IEEE Transactions on Parallel and Distributed Systems* 13 (2002) 1099–1104.

[15] P.A.J. Hilbers, M.R.J. Koopman and J.L.A. van de Snepscheut, The twisted cube, *Proc. of Parallel Architectures and Languages Europe* (J.W. de Bakker, A.J. Nijman, P.C. Treleaven, eds.), Lecture Notes in Computer Science Vol. 258, Springer-Verlag (1987) 152–159.

[16] P.D. Kulasinghe, Connectivity of the crossed cube, *Information Processing Letters* 61 (1997) 221–226.

[17] T.-K. Li, J.J.M. Tan, L.-H. Hsu and T.-Y. Sung, The shuffle-cubes and their generalization, *Information Processing Letters* 77 (2001) 35–41.

[18] J. Maeng and M. Malek, A comparsion connection assignment for self-diagnosis of multiprocessor systems, *Proc. 11th Int. Symp. on Fault-Tolerant Computing* (1981) 173–175.

[19] M. Malek, A comparsion connection assignment for diagnosis of multiprocessor systems, *Proc. 7th Ann. Symp. on Computer Architecture*, ACM Press (1980) 31–35.

[20] A. Sengupta and A.T. Dahbura, On self-diagnosable multiprocessor systems: diagnosis by the comparison approach, *IEEE Transactions on Computers* 41 (1992) 1386–1396.

[21] I.A. Stewart, Distributed algorithms for building Hamiltonian cycles in k-ary n-cubes and hypercubes with faulty links, *Journal of Interconnection Networks* 8 (2007) 253–284.

[22] N. Tzeng and S. Wei, Enhanced hypercubes, *IEEE Transactions on Computers* 40 (1991) 284–294.

[23] D. Wang. Diagnosability of hypercubes and enhanced hypercubes under the comparison diagnosis model, *IEEE Transactions on Computers* 48 (1999) 1369–1374.

[24] A.Y. Wu, Embedding of tree networks into hypercubes, *Journal of Parallel and Distributed Computing* 2 (1985) 238–249.

[25] Y. Xiang and I.A. Stewart, Augmented k-ary n-cubes, *Information Sciences*, to appear.

[26] X. Yang and Y.Y. Tang, Efficient fault identification of diagnosable systems under the comparison model, *IEEE Transactions on Computers* 56 (2007) 1612–1618.

[27] X. Yang, A linear time fault diagnosis algorithm for hypercube multiprocessors under the MM* comparison model, *Proc. of 12th Asian Test Symposium*, IEEE Press (2003) 50–57.

[28] J. Zheng, S. Latifi, E. Regentova, K. Luo and X. Wu, Diagnosability of star graphs under the comparison diagnosis model, *Information Processing Letters* 93 (2005) 29–36.