Node Fault Tolerance in Graphs

Frank Harary

Department of Computer Science, New Mexico State University, Las Cruces, New Mexico 88003

John P. Hayes

Department of Electrical Engineering and Computer Science, University of Michigan,
Ann Arbor, Michigan 48109

A graph G^* is a k-node fault-tolerant supergraph of a graph G, denoted k-NFT(G), if every graph obtained by removing k nodes from G^* contains G. A k-NFT(G) graph G^* is said to be optimal if it contains $n + k$ nodes, where n is the number of nodes of G and G^* has the minimum number of edges among all $(n + k)$-node k-NFT supergraphs of G. We survey prior results on the design of optimal k-NFT supergraphs of various useful forms of G; this work covers cycles and various types of trees. We also introduce the concept of exact node fault tolerance, which requires that every graph obtained by removing k nodes from G^* be isomorphic to G, and explore its basic properties. We conclude with a discussion of some unsolved and partially solved problems. © *1996 John Wiley & Sons, Inc.*

1. INTRODUCTION

Fault tolerance is the ability of a system such as a computer or a communication network to operate correctly in the presence of faults. Graph theoretic models of fault tolerance in these systems often involve two graphs G and G^*, where G must be embedded in G^* even when a fault removes certain nodes or edges from G^*. The following problem is of great practical interest: Given a graph G that represents the structure of some useful system, construct a graph G^* that has a prescribed level of fault tolerance with respect to G and also satisfies some optimality constraint, such as having the minimum number of edges or the maximum node degree. The supergraph G^* may be constructed from G by adding extra or "spare" nodes or edges to G; the problem is how to do so efficiently. We examine the design of optimal node-fault-tolerant graphs in which the faults of interest remove a prescribed number of nodes from G^* [8]. A companion paper [5] addresses edge-fault-tolerant graphs, for which the faults of interest only remove edges from G^*.

We begin by formalizing the concept of node fault tolerance. Let G be a graph with n nodes and q edges. An $(n + k)$-node graph G^* is *k-node fault-tolerant,* or k-NFT, with respect to G if every graph $G^* - S$ obtained by removing any set S of $k > 0$ nodes from G^* contains G. We will refer to G^* as a k-NFT supergraph of G or simply as a k-NFT(G). We also say $G^* \in k$-**NFT**(G), the set of all k-NFT(G) supergraphs of G. The complete graph K_{n+k} of $n + k$ nodes is trivially a k-NFT supergraph of every G that contains up to n nodes. We are concerned mainly with k-NFT graphs that satisfy the following optimality criterion: If G^* has the smallest number $|E(G^*)|$ of edges among all $(n + k)$-node supergraphs that are k-NFT with respect to G, then G^* is *optimally k-NFT* with respect to G. The number $ec(G, k) = |E(G^*) - E(G)|$ is called the *k-NFT edge cost* of G.

Figure 1(b) depicts a graph, the 6-node wheel W_6, which is 1-NFT with respect to the 5-node cycle C_5 of Figure 1(a), i.e., $W_6 \in$ 1-**NFT**(C_5). Clearly, C_5 can be embedded in the 5-node periphery of W_6, in which case the new center node s and its incident edges are regarded as *spares.* If some peripheral node f "fails," we can modify the embedding to obtain the copy of C_5 shown by the heavy lines in Figure 1(b), in which s replaces or *covers* f. Because the spare node s is adjacent to every node ap-

NETWORKS, Vol. 27 (1996) 19–23

CCC 0028-3045/96/010019-05

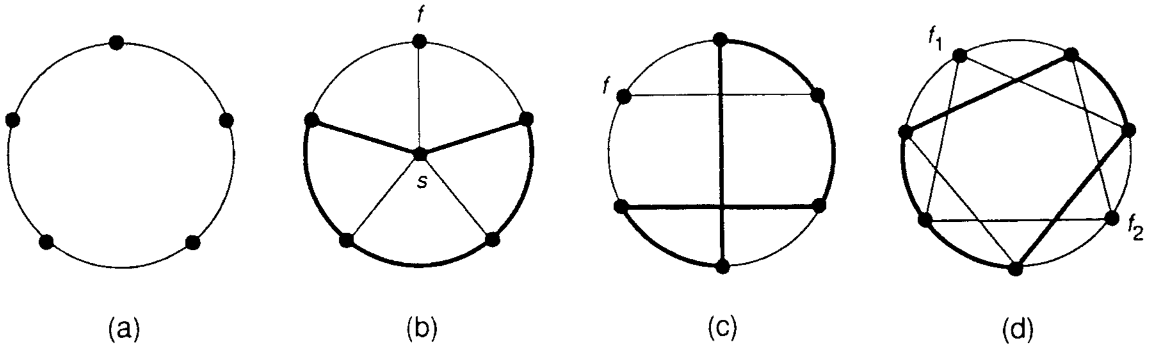


Fig. 1. (a) The cycle C_5; (b) a nonoptimal 1-NFT(C_5); (c) an optimal 1-NFT(C_5); (d) an optimal 2-NFT(C_5).

pearing in the original C_5, it is called a *global spare* node. The graph W_6 has 10 edges and is not an optimal 1-NFT(C_5) graph. It is easily proven that the 9-edge graph of Figure 1(c), the triangular prism $K_3 \times K_2$, is an optimal 1-NFT(C_5), which implies that the edge cost $ec(C_5, 1) = 4$. Unlike the wheel, this graph does not have a global spare. Figure 1(d) shows C_7^2, the square of C_7, which happens to be an optimal 2-NFT(C_5), implying that $ec(C_5, 2) \leq 7$. The heavy lines show an embedding of C_5 in the presence of the 2-node fault set $S = \{f_1, f_2\}$.

The problem of constructing an optimal k-NFT(G) was introduced in [8], where it was solved for paths, cycles, and some special types of trees. More recent work has extended these results and introduced new optimization criteria, such as minimizing the effort required for reconfiguration, i.e., deriving a new embedding of G after a fault occurs. However, many basic questions remain unresolved, such as characterizing optimal k-NFT(G) for arbitrary trees. Figure 2 gives two optimal 1-NFT supergraphs of the 5-node tree Y_5 shown in heavy lines. Observe that the two supergraphs are not isomorphic and neither has a global spare.

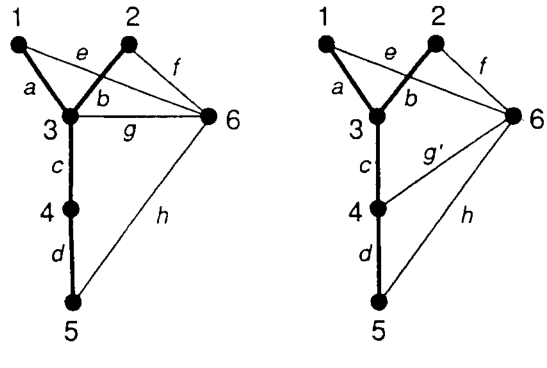


Fig. 2. Two optimal 1-NFT supergraphs of the tree Y_5 shown in heavy lines.

2. OPTIMAL NODE FAULT TOLERANCE

We now review the known results on optimal k-NFT(G) graphs. First, consider the case where G is the n-node cycle C_n. If G^* is an $(n + k)$-node k-NFT(C_n), then the degree of every node in G^* is at least $k + 2$, for, otherwise, a k-node fault F can reduce G^* to an n-node graph $G^* - F$ in which some node has degree less than two; hence, $G^* - F$ could not contain C_n. It follows that an optimal k-NFT(C_n) must contain at least $\lceil (k + 2)(n + k)/2 \rceil$ edges. The construction of graphs that meet these bounds exactly leads to the following results [8]:

Theorem 1. The two graphs of Figure 3, which are the Hamiltonian-connected graphs with $n + 1$ nodes and the smallest number of edges, are optimal 1-NFT(C_n) graphs.

We do not know whether these are the only such optimal graphs and ask for the determination of all such graphs. Figure 1(c) illustrates Theorem 1 for $n = 5$. Following [8], we give a general construction for optimal 1-NFT(C_n) graphs when $k > 1$. Let C_s^t denote the power graph obtained by adding edges to the cycle C_s that join each node i to all nodes at distance t or less from i.

Theorem 2. Let $k = 2h$ when k is even and $k = 2h + 1$ when k is odd.

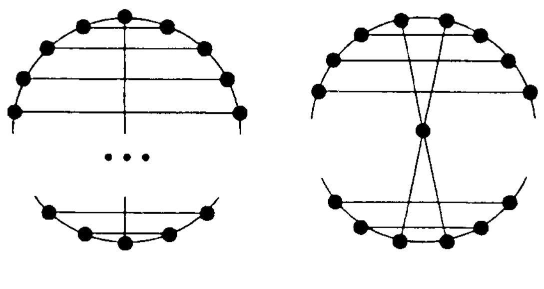
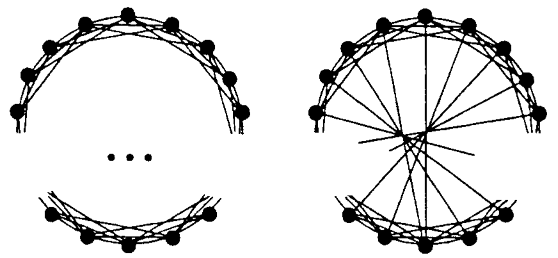


Fig. 3. Optimal 1-NFT(C_n) graphs for (a) n odd and (b) n even.



(a) (b)

Fig. 4. Optimal k-NFT(C_n) graphs, $k > 1$, for (a) n odd and (b) n even.

(a) When k is even, the power graph C_{n+k}^{h+1} shown in Figure 4(a) is an optimal k-NFT(C_n).
(b) When k is odd, the graph obtained by adding $\lfloor (n + k + 1)/2 \rfloor$ "bisector" edges to C_{n+k}^{h+1}, as shown in Figure 4(b), is an optimal k-NFT(C_n).

Figure 1(d) illustrates Theorem 2a for $n = 5$ and $k = 2$. Again, we do not know whether there are additional optimal graphs beyond those of Figures 3 and 4.

We turn next to the simplest type of tree, namely, the n-node path P_n. Now, P_n is obtained by deleting any node from C_{n+1}; therefore, C_{n+1} is an optimal 1-NFT(P_n). This property generalizes in an obvious way to the following result:

Theorem 3 [8]. For every $k \geq 1$, the two sets of graphs k-NFT$(P_n) = (k-1)$-NFT(C_{n+1}).

Hence, we can use Theorems 1 and 2 to construct optimal k-NFT supergraphs of P_n for all k. For example, the graph $C_7^2 =$ 2-NFT(C_5) of Figure 1(d) is an optimal 3-NFT(P_4).

Another simple form of tree is the n-node *star*, denoted $K_{1,n-1}$ or $K(1, n-1)$, which is a bipartite graph. Farrag and Dawson [2] characterized fully the k-NFT supergraphs of $K_{1,n-1}$. Following [2], let $K(1,1,\ldots,1,n-1)$ denote a complete q-partite graph whose nodes are partitioned into q disjoint subsets, with an edge between every pair of nodes u and v if and only if u and v belong to different subsets.

Theorem 4. [2].

(a) The complete $(k+2)$-partite graph $K(1, 1, \ldots, 1, n-1)$ is an optimal k-NFT supergraph of $K(1, n-1)$ for all even n and for odd n that satisfy $k \leq n^2 - 4n + 3$.
(b) Let n be odd and $k \geq n^2 - 4n + 3$. The complete $[(k+n)/2]$-partite graph $K(2, 2, 2, \ldots, 2)$ is an optimal k-NFT supergraph of $K(1, n-1)$ when k is odd; the $[(k+n+1)/2]$-partite graph $K(1, 2, 2, 2, \ldots, 2)$ is an optimal k-NFT supergraph of $K(1, n-1)$ when k is even.

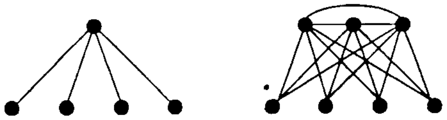
Figure 5(a) depicts the star $K(1,4)$, while Figure 5(b) shows the complete 4-partite graph $K(1, 1, 1, 4)$ which is the optimal 2-NFT $K(1, 4)$; its uniqueness is proven in [2]. Here, $n = 5$ and $k = 2$, so $n^2 - 4n + 3 = 8 \geq k$. It is readily seen that $K(1, 1, \ldots, 1, n-1)$ can be formed by joining the k-node complete graph K_k to the star $K(1, n-1)$. In other words, $K(1, 1, \ldots, 1, n-1) = K_k + K(1, n-1)$, where $+$ denotes the join operation [4, p. 24]. Consequently, $K(1, 1, 1, 4)$ of Figure 5(b) can be expressed as $K_3 + K(1, 4)$.

3. EXACT FAULT TOLERANCE

It will be clear from the preceding examples that it is by no means easy to determine whether G is a subgraph of $G^* - F$. In fact, this is a case of the well-known "subgraph isomorphism problem." We now introduce the concept of *exact* fault tolerance, for which $G^* - F$ is required to be isomorphic to G, written $G \cong G^* - F$, for all possible fault sets. We say that G^* is an *exact k-NFT supergraph* of G if every graph obtained by removing k nodes from G^* is isomorphic to G. Every exact k-NFT supergraph of G is also an optimal k-NFT supergraph of G. It is obvious that the complete graph K_{n+k} is exactly k-NFT with respect to K_n for all n and k and that the cycle C_{n+1} is an exact 1-NFT supergraph of the path P_n. Theorems 1–3 imply that no optimal k-NFT supergraph of P_n can be exact for $k > 1$ and $n > 1$.

There is a close relationship between 1-node fault tolerance and node symmetry. If G^* is node-symmetric, then all subgraphs of the form $G^* - v$ obtained by deleting a node v from G^* are isomorphic. This leads to the following result:

Theorem 5. Graph G^* is an exact 1-NFT supergraph of G if and only if G^* is node-symmetric.



(a) (b)

Fig. 5. (a) The star $K(1, 4)$ and (b) the optimal 3-NFT $K(1, 4) = K(1, 1, 1, 4)$.

The "if" part of this theorem is trivial. The "only if" part follows readily from the result of Harary and Palmer [7]. Theorem 5 implies the existence of a large class of graphs that are exactly 1-NFT. For example, the triangular prism $G^* = K_3 \times K_2 \in$ 1-**NFT**(C_5), as depicted in Figure 1(c). Because G^* is node-symmetric, it is exactly 1-NFT with respect to the 5-node 6-edge graph consisting of C_5 plus an extra edge.

It is easy to verify that no G^* is an exact 1-NFT(C_5). It is an unsolved problem to determine which graphs G can be exactly 1-NFT. Clearly, the complete graph K_n has exact k-NFT supergraphs for all k. It does not appear, however, that other exact k-NFT supergraphs exist for $k \geq 2$.

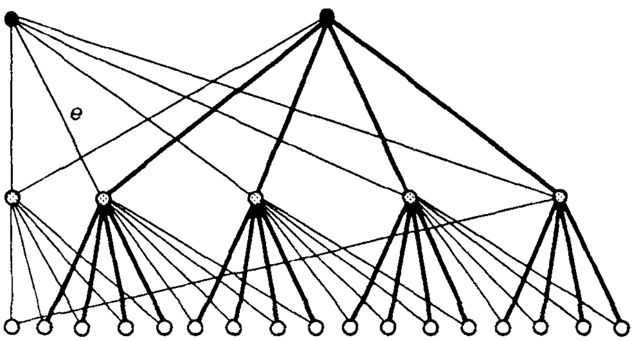


Fig. 7. A near-optimal 1-NFT supergraph $G^+[1, T_3^{(4)}]$ of the NST $T_3^{(4)}$.

4. UNSOLVED PROBLEMS

We discuss some additional concepts and results pertaining to node fault tolerance. We also present a list of open problems and some partial solutions.

A. Minimizing Global Spares

As noted earlier, a global spare node is adjacent to every node of G^*. We ask: When does an optimal 1-NFT(G) require a global spare? Clearly, K_{n+1} = 1-NFT(K_n) is optimal and has a global spare, but cycles, paths, and most types of trees do not. As above, let a 1-NFT graph G^* contain a global spare node s. We term any edge that is incident with s any *s-edge* of G^*. The five spokes of the wheel $W_6 \in$ 1-**NFT**(C_5) in Figure 1(b) are examples of s-edges. Let G^* be k-NFT with respect to G and contain the minimum number of s-edges. The number $sec(G, k) = |E(G^*) - E(G)|$ is called the *k-NFT s-edge cost* of G and, like $ec(G, k)$, is an invariant of G. The questions then arise: What is $sec(G, k)$ for interesting classes of graphs and when is $ec(G, k) = sec(G, k)$? In the case of caterpillars, the following result has been found:

Theorem 6 [6]. For a caterpillar G, $ec(G, 1) = sec(G, 1)$.

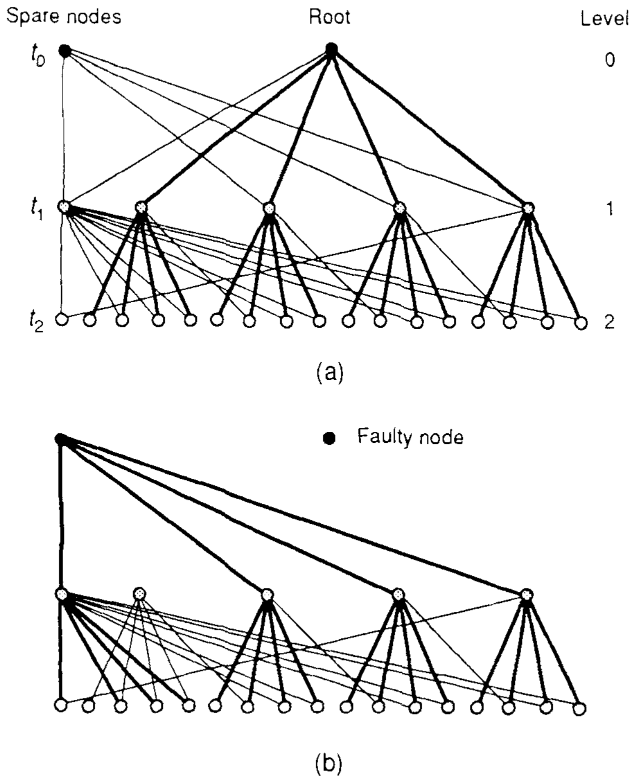


Fig. 6. An optimal 1-NFT supergraph $G^*[1, T_3^{(4)}]$ of an NST: (a) basic configuration with spares; (b) reconfiguration around a faulty root node.

B. Trees Revisited

A *full* n*-ary tree* is a rooted balanced tree, in which every *nonleaf node has* n *children. Let* $T_h^{(n)}$ *denote the full* n-ary tree of height h. The level of a node in such a tree is its distance from the root; there are $h + 1$ levels in the tree, with the leaf nodes forming level h. The path P_n is a full unary tree of height $n - 1$, while the star $K(1, n)$ is *a full* n*-ary tree of height one. The tree* $T_h^{(n)}$ *is called nonhomogeneous* if a distinct type (color) t_i is assigned to all nodes of level i; otherwise, $T_h^{(n)}$ is homogeneous. Node-fault-tolerant, nonhomogeneous trees require that each faulty node be replaced by a (spare) node of the same type. *The following problem was first posed in [8]:* Design a k-NFT supergraph $G^*[k, T_h^{(n)}]$ of a nonhomogeneous $T_h^{(n)}$, so that G^* has k spare nodes of each type and also has minimum edge count. Such a tree is referred to as an optimal k-NFT *nonhomogeneous symmetric tree (NST).*

Hayes [8] presented a general method for constructing optimal 1-NFT NSTs, which is illustrated in Figure 6 for the quaternary NST $T_3^{(4)}$ of height 3. The heavy edges in the graph $G^*[1, T_3^{(4)}]$ of Figure 6(a) show the underlying NST $T_3^{(4)}$. There are three spare nodes, one per level and each of a different type, which appear at the left of the graph. The fault tolerance of $G^*[1, T_3^{(4)}]$ is illustrated by Figure 6(b) for a fault affecting the root node; the heavy lines show a fault-free subgraph isomorphic to $T_3^{(4)}$. Kwan and Toida [9] discovered how to design an optimal $G^*[k, T_h^{(n)}]$ for $n = 2$ and $k = 2$, but the general NST problem still remains open—an indication of the difficulty of finding optimal k-NFT trees.

C. Near-optimal Fault Tolerance

A number of workers have devised methods for constructing k-NFT supergraphs of trees that exhibit various desirable properties such as ease of reconfiguration, but are far from optimal. A design for a 1-NFT supergraph of $T_h^{(n)}$ that is provably near-optimal is easily described as follows [1]:

1. Place a spare node of type t_i at the left of each level i of $T_h^{(n)}$.
2. Insert an edge between each node u in level i and all the children of u's right hand neighbor.
3. Insert edges between the left- and rightmost nodes in level i and the spare node in level $i + 1$.

Figure 7 shows the 1-NFT supergraph $G^+[1, T_3^{(4)}]$ produced by applying the above procedure to $T_3^{(4)}$. On comparing the 1-NFT designs of Figures 6 and 7, we see that the second is nonoptimal with respect to edge cost. However, $G^+[1, T_3^{(4)}]$ has only one more edge per node between levels—e.g., the edge e between levels 0 and 1—so it can be said to be nearly optimal. The graph $G^+[1, T_h^{(n)}]$ has several desirable characteristics not possessed by $G^*[1, T_h^{(n)}]$. For example, its node degrees are better balanced than those of $G^*[1, T_h^{(n)}]$, in which the spare nodes have very big degrees. Reconfiguration around a faulty node in level i of $G^+[1, T_h^{(n)}]$ can be confined to levels $i - 1$, i, and $i + 1$, a desirable property in fault-tolerant computers. The supergraph $G^+[1, T_h^{(n)}]$ can also tolerate up to h faulty nodes in any positions, provided there is only one fault per level. This implies that $G^+[1, T_h^{(n)}]$'s h spare nodes can be fully utilized. Dutt and Hayes [1] showed that the near-optimal construction of Figure 7 can be extended to k-NFT designs for any k.

This research was supported by Grant N00014-90J1860 from the U.S. Office of Naval Research.

REFERENCES

[1] S. Dutt and J. P. Hayes, On designing and reconfiguring k-fault-tolerant tree architectures. *IEEE Trans. Comput.* **39** (1990) 490–503.

[2] A. A. Farrag and R. J. Dawson, Designing optimal fault-tolerant star networks. *Networks* **19** (1989) 707–716.

[3] A. A. Farrag and R. J. Dawson, Fault-tolerant extensions of complete multipartite networks. *Proceedings of the 9th International Conference on Distributed Computing* (1989) 143–150.

[4] F. Harary, *Graph Theory*. Addison-Wesley, Reading, MA (1969).

[5] F. Harary and J. P. Hayes, Edge fault tolerance in graphs. *Networks* **23** (1993) 135–142.

[6] F. Harary and M. Khurrum, One node fault tolerance for caterpillars and starlike trees. *Internet J. Computer Math.* **56** (1995) 135–143.

[7] F. Harary and E. M. Palmer, On similar points of a graph. *J. Math. Mech.* **15** (1966) 623–630.

[8] J. P. Hayes, A graph model for fault-tolerant computer systems. *IEEE Trans. Comput.* **C-25** (1976) 876–884.

[9] C. L. Kwan and S. Toida, An optimal 2-FT realization of symmetric hierarchical tree systems. *Networks* **12** (1982) 231–239.

Received January 5, 1994
Accepted May 17, 1995