Self-repairing networks
&
Minimum self-repairing graphs

Arthur M. Farley & Andrzej Proskurowski
August 1993
CIS-TR-93-18

Abstract

In these two papers, we discuss the topological and operational aspects of sparse networks, in which the failure of a single site does not increase shortest-path distance between any pair of remaining sites.

Department of Computer and Information Science
University of Oregon

Self-Repairing Networks

by
Arthur M. Farley
Andrzej Proskurowski

Computer and Information Science
University of Oregon
Eugene, OR 97405 USA

Abstract

In this paper, we provide a specification for a class of communication networks that are immune to single site failures, not only maintaining the ability to transfer messages between remaining, operable sites but doing so with no additional delay (i.e., no increase in length of communication path). Our specification includes a constructive characterization of a class of minimal self-repairing graphs and an algorithmic determination of associated routing tables that can be used by a simple message transfer procedure to realize the desired immune behavior.

We call these networks self-repairing and refer to their underlying topologies as self-repairing graphs. Starting with a 4-cycle, we generate larger, minimal self-repairing graphs by operations of connecting a vertex of degree 2 to a pair of twins (i.e., vertices having identical open neighborhoods) in a given, self-repairing graph , doubling the graph, or cloning a vertex of the graph. We add new entries to routing tables as each graph operation is applied.

Introduction

A communication network provides means for the dissemination of information among a set of *sites* by transmission of *messages* over a set of *lines* interconnecting the sites. Complete specification of a communication network requires consideration of two elements: *topology* and *operation*. A network's topology corresponds to relatively static aspects of the network: the set of sites, how they are interconnected by lines, and other relevant features, such as line length, cost, and capacity, etc. A network's operational specification indicates essential features of its behavior, such as whether it is a store-and-forward or broadcast network, whether a site can communicate with more than one site concurrently, and particulars of communication protocols, including routing tables and calling sequences to be used for given information dissemination tasks.

The basic information dissemination task is that of *message transfer*, which transfers a message between one site, the *sender*, and another site, the *receiver*. In a store-and-forward network, message transfers are completed by series of calls placed between directly connected sites. In previous research, we considered network design for *broadcast* (sending a message from one site as sender to all other sites as receivers) and *gossip* (sending messages from all sites as senders to all sites as receivers) [4]. In this paper, we focus on effective network designs for the task of message transfer under conditions of single site failure.

We model the topological aspects of a network by an undirected graph G = (V,E), consisting of a set V of *vertices*, corresponding to network sites, and a set E of *edges*, each edge connecting a pair of vertices (v_1,v_2), corresponding to lines of the network. For a given vertex v, let $i_G(v)$ represent the set of edges incident to v in graph G, i.e., those for which v is a member of the edge's vertex pair. The *degree* of a vertex v in graph G, $deg_G(v)$, is equal to the number of elements in $i_G(v)$. For a given vertex v of graph G, let $n_G(v)$ denote the open neighborhood of v, being the set of vertices with which v shares edges in G. Let $n_G[v]$ denote the closed neighborhood of v, being $n_G(v) + \{v\}$.

A *path* between two vertices *x* and *y* is a sequence of edges $(v_1,v_2),\ldots,(v_{n-1},v_n)$ such that $v_1=x$ and $v_n=y$. We denote such a path as the sequence of vertices encountered on the path, i.e., $(x = v_1, \ldots, v_n = y)$. The *length* of a path is the number of edges in the path. For a given pair of vertices, u and v, in G, let the distance between them, $d_G(u,v)$ be the length of a shortest path from u to v in G. Two vertices at a distance of 1 are directly connected by an edge and are said to be *adjacent* or *neighbors*.

A graph is *connected* if there exists a path between every pair of vertices. The *vertex-connectivity of a pair of vertices* in a graph is equal to the maximum number of vertex-disjoint paths that can be found between the pair; the *vertex-connectivity of a graph* is the minimum of the vertex connectivity over all pairs of vertices in the graph. We shall say that a graph is *k-connected*, for positive integer k, with k not greater than the vertex connectivity of the graph.

Another perspective on connectivity is the effects of removing certain elements of a graph. A graph is *2-connected* iff for every vertex v of G, the graph $G'=(V-\{v\}, E-i_G(v))$ is connected. In other words, removing any vertex of G does not disconnect the graph. A *minimally 2-connected* graph is a 2-connected graph such that the removal of any edge results in a graph that is not 2-connected. A *minimum 2-connected* graph is a 2-connected graph having the fewest edges for a given number of vertices. Obviously, the set of minimum 2-connected graphs is a subset of minimally 2-connected graphs. These notions of minimal and minimum subsets of a class of graphs can be applied to other classes of graphs, as well.

Viewing a graph as a model of the toplogy of a communication network, we see that a graph that is 2-connected represents a network that can be made *immune* to single failures. The failure of a single site (vertex) would still allow all remaining sites to communicate with each other, assuming appropriate rerouting of message transfers as needed. To complete specification of an immune network, we must specify a communication protocol that completes possible message transfers under anticipated failure conditions. In this paper, we investigate a class of 2-connected graphs and associated immune networks that preserve vertex distances, and thus message transfer times, under single vertex removal, or site failure, conditions.

We say a graph G is *immune with penalty k, IP(k)*, if it is 2-connected and the maximum (additive) increase in distance between any pair of vertices, regardless of which vertex is removed, is equal to k. Cycles, i.e., minimum 2-connected graphs in which every vertex has degree 2, pay the highest penalty in maximum increased distance when a vertex is removed for any graph with n vertices. Vertices that neighbor the removed vertex are at distance 2 in the original graph. However, they are at distance n-2 in the graph with the vertex removed; as such, the distance between them is increased by n-4. From this, we can conclude the following: If G is a 2-connected graph of n vertices, then G is IP(k) for some k, $0 \le k \le n-4$.

Cycles are important aspects of 2-connected and IP(k) graphs. In a 2-connected graph, there are 2 vertex disjoint paths between every pair of vertices; thus, every pair of vertices lies on a cycle. A cycle with n vertices is called an *n-cycle*, denoted as C_n.

Self-Repairing Graphs

We will focus on the class of IP(0) graphs, which we will call *self-repairing graphs*. Self-repairing graphs are graphs, such that the removal of any single vertex results in no increase in distance between any pair of remaining vertices. We have the following theorem, which localizes this global distance property to vertex neighborhoods.

Theorem 1. A graph G is self-repairing iff, for every vertex v and for each pair of vertices x and y in $n_G(v)$, $n_{G'}[x] \quad n_{G'}[y]$ is not empty, where $G' = (V-\{v\}, E-i_G(v))$

Proof: Let v be the vertex removed from G to form G'. For a pair of vertices, u and w, in G', such that no shortest path between them in G includes v, $dist_G(u, w) = dist_{G'}(u,w)$. However, if there exists a shortest path in G between u and w that includes v, then we must show that there exists a path in G' of the same length. The shortest path through v in G is of the form ($u = ...v_u \; v \; v_w ... = w$), where there are zero or more vertices between u and v_u and between w and v_w. By our assumption as to the non-empty intersection of neighborhoods of v_u and v_w in G', there exists a path of the form ($u = ...v_u \; v'v_w \; ... = w$) in G'. Thus, $dist_G(u, w) = dist_{G'}(u,w)$.

If our assumption as to non-empty neighborhood intersections is not true, i.e., there exists vertices u and w that are neighbors of a vertex v in G and do not share a vertex in their closed neighborhoods in G', then $dist_{G'}(u, w) > dist_G(u, w) = 2$. []

We can restate Theorem 1 as the following corollary in terms of the possible size of a cycle involving pairs of neighbors of a vertex in a self-repairing graph.

Corollary 1. G is a self-repairing graph iff, for every vertex v in G and for each pair of vertices x and y in $n_G(v)$, vertices v, x, and y are members of a 3-cycle or 4-cycle in G.

We have characterized self-repairing graphs in terms of a local property of vertex neighborhoods. Are there graphs which have this property? The complete graph on n vertices ($n>2$), in which every vertex is connected to every other vertex, is a self-repairing graph; every vertex and any two neighbors form a cycle of length 3. Complete graphs with n vertices require $O(n^2)$ edges. If we remove an edge from a complete graph having more than three vertices, the remaining graph is still self-repairing. Thus, complete graphs are not minimal self-repairing graphs; fewer edges suffice. Can we find a constructive characterization of an sparse class of self-repairing graphs, where by *sparse* we will mean minimal self-repairing graphs having only O(n) edges? In addition, can we find a calling scheme and routing tables sufficient to generate the desired, immune communication in the corresponding networks?

Sparse Self-Repairing Graphs

We consider the class of twin graphs, to be defined below. Vertices u and v in a graph G are *twins* iff $n_G(x) = n_G(y)$. For a given vertex v in graph G, $twin_G(v)$ is the set, possibly empty, of vertices that are twins of v in G. Based upon this notion of twin vertices, we define a *twin graph* as follows: (i) the 4-cycle is a twin graph; (ii) if G is a twin graph, then the graph G' constructed by connecting a new vertex by two edges to a pair of twins in G is a twin graph. Note that, when a new vertex is connected to a pair of twins in G, the pair remains twins in G'. In fact, once a pair of twin vertices have degree higher than 2, the two vertices must remain twins of each other throughout further construction of the twin graph.

Theorem 2: A twin graph G is self-repairing.

Proof: Let G be a twin graph. Each vertex of G is of degree 2 or higher. Let a vertex v of degree 2 be removed from G to form G'. In G, v is connected to twin vertices x and y, which, by definition, share equal neighborhoods in G'. Each vertex v of degree greater than 2 in G has a unique twin vertex, u. Every neighbor of vertex v in G is also a neighbor of u. Thus, if v is removed to form graph G', the intersections of neighborhoods of all pairs of vertices in $n_G(v)$ include u. Thus, G is self-repairing, by Theorem 1. []

Twin graphs with n vertices have 2n-4 edges, as the 4-cycle has 2n-4 edges and 2 new edges are added for each new vertex connected to the graph. Furthermore, twin graphs are minimal self-repairing graphs.

Theorem 3. Twin graphs are minimal self-repairing graphs.

Proof: The 4-cycle is minimal, as the removal of any edge results in a graph that is not 2-connected. In a larger twin graph, the removal of an edge that results in a vertex of degree 1 creates a graph that is not 2-connected. Thus, if an edge can be removed from a twin graph and not eliminate the self-repairing property, the edge must be between two vertices x and y, each of degree greater than 2. Without loss of generality, let x have the uniquetwin vertex x', to which y is also connected by an edge, and y have unique twin vertex y', which is connected by edges to x, x', and anotherm vertex z, where z is also connected to y. See Figure 1. This local subgraph must be part of the twin graph G from which we remove edge (x, y) to create graph G'. Is G' self-repairing? No. Consider the removal of vertex v_y from G', forming graph G". Then, $dist_{G'}(z, x) = 2$, but $dist_{G''}(z, x) \geq 4$. []

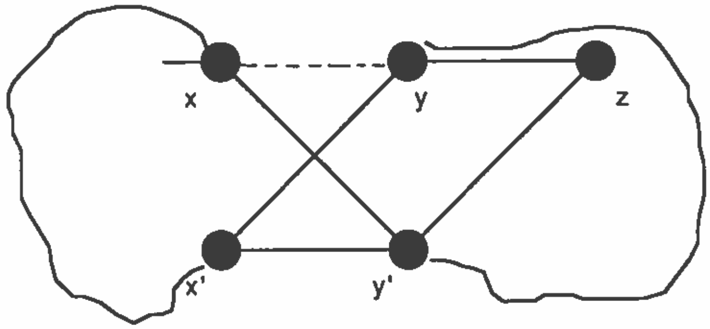


Figure 1. Neighborhood in twin graph.

Thus, by our definition of sparse graphs, twin graphs are sparse, self-repairing graphs. In a companion paper, we show that twin graphs, together with the 3-regular cube graph on 8 vertices, constitute exactly the class of minimum self-repairing graphs, i.e, those having the fewest edges over all self-repairing graphs for a given number of vertices [3]. It is interesting to note that twin graphs are not contained in the class of minimally 2-connected graphs. Due to the strong restriction on allowable distance penalty of 0, more edges are needed in minimal self-repairing graphs than in minimally 2-connected graphs. Figure ~~s the smallest twin graph that is not minimally 2-connected.

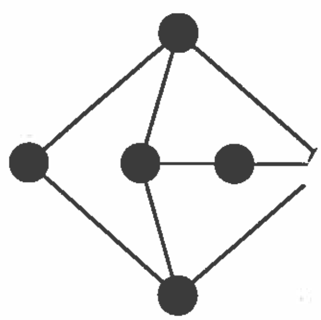


Figure 2. The smallest twin graph that is not minimally 2-connected.

Can we extend our constructive characterization of sparse, self repairing graphs? Given a graph G, we define the double of G, *double(G)*, to be the graph composed of two copies of G (called *components*), with *bridging edges* connecting corresponding(isomorphic) pairs of vertices in a one-to-one fashion. Figure 3 shows that the cube graph is a doubled 4-cycle.

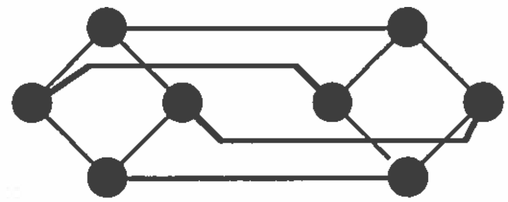


Figure 3. Cube Graph as double(C_4).

Theorem 4. If G is a self-repairing graph, then double(G) is a self-repairing graph.

Proof: By Corollary 1, we must show that every pair of neighbors of a given vertex v is in a 4-cycle that includes v. If both neighbors of v are in the same component as v in double(G), then, by our definition that each component of double(G) is a copy of the self-repairing graph G, the required condition is met. When the neighbors of v are in differing components, one neighbor u is within the same component as v and the other neighbor v' corresponds to v in the other component (connected to v by a bridging edge). By definition, vertex u has a neighbor u' corresponding to u in the same component as v'. Since v is a neighbor of u in one component, their corresponding vertices (v' and u') are neighbors in the other component. As such, vertices v, u, u', and v' form a 4-cycle. Thus, double(G) is self-repairing. []

By the above theorem, the class of graphs known as hypercubes, obtained by iterated doubling of a 4-cycle, are self-repairing graphs. The doubling operation preserves the property of being minimal self-repairing, as well.

Theorem 5. If G is a minimal self-repairing graph, then double(G) is minimal self-repairing.

Proof: By the above theorem, we know the self-repairing property is preserved. We only need to show that every edge in double(G) is required to maintain this property. There are two cases to consider: (i) the edge removed is between two vertices of a single component, and (ii) the edge removed is a bridging edge between vertices of different components. Consider case (i). By our assumption as the the minimal property of G, elimination of such an edge means that the distance between at least one pair of vertices in the affected component must increase when a vertex is removed. The original distance between these vertices can not be maintained by using a path involving bridging edges, as going between components adds at least 2 to the original distance in a single component. Thus, all edges within each component are needed.

Looking at case (ii), consider eliminating the bridging edge between a vertex v and its corresponding vertex v' in the other component. Now consider the removal of a neighbor u of v in v's component and the impact this will have on the distance between v and u', the vertex corresponding to u in the other component. In the graph before removal of u, the distance is 2; without the edge between v and v', the distance after removal of u must be at least 3. Thus, the graph without edge (v, v') is not self-repairing. This establishes our theorem. []

We can maintain the property of self-repairing by the operation of vertex-cloning. A graph G' is obtained from graph G by the operation of *vertex cloning* , i.e., adding vertex v' to graph G such that the open neighborhood of v' in G' is made equal to the open neighborhood of some vertex v in G, i.e., $n_{G'}(v') = n_G(v)$ (= $n_{G'}(v)$). Vertex v' is a *clone* of v in G'.

Theorem 6. If graph G' is formed from a self-repairing graph G by adding vertex v' as a clone of vertex v in G, then G' is self-repairing.

Proof: As v and v' are twins in G', all pairs of their (common) neighbors are elements of 4-cycles through these two twins. Likewise, each pair of neighbors of those neighbors is in a 4-cycle, either because both were in such a cycle in G already, or, if the pair is v' and another vertex u, we can use the same vertex of G that forms a 4-cycle for the pair v and u in G. Other neighborhoods of G are not affected. Thus, by Corollary 1, G' is self-repairing. []

Note that vertex cloning creates at least one new twin pair in a self-repairing graph, allowing addition of degree 2 vertices between these twins while preserving the self-repairing property (by Theorem 1). This observation suggests that vertex cloning does not guarantee minimality of the resultant self-repairing graph in all cases. In fact, we have the following.

Theorem 7: If we clone vertex v by v' in a self-repairing graph G, where vertex v is of degree greater than 2 and has neighbors x and y that are twins in G, then the resulting graph G' is not minimal self-repairing.

Proof: We need only connect vertex v' to the pair of twins x and y in the neighborhood of v. If we do this, vertex v' and its neighbors x and y form a 4-cycle as x and y are twins and share a common neighbor. Similarly, x (y) together with v' and any other neighbor of x (y) form a 4-cycle through y (x). Thus, v' need only be connected to x and y. Cloning v in this case introduces unneeded edges into graph G'. []

This means cloning vertices of degree greater than 2 in a twin graph will destroy minimality.

Routing Tables for Self-Repairing Networks

We have shown that twin graphs and doubled graphs are self-repairing. To complete the design of the corresponding, self-repairing communication networks, we must determine calling schemes and routing tables that allow message transfers to be completed in the same number of successful calls (i.e., over paths of the same length), given single site failures.

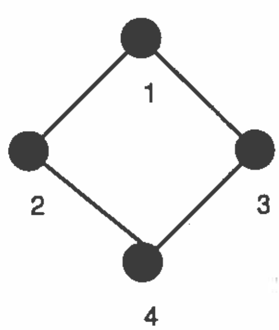
In our model of network operation, we focus on characterizing the communication process carried out to perform a particular information dissemination task. A *communication process* is the total activity performed in completing an information dissemination task. In our model, the basic activity is the *call*, being the transfer of a message between two neighboring sites (adjacent vertices). A call is initiated by a procedure *call(message, neighbor)*, which has the site transfer *message* to the specified *neighbor*; the procedure indicates success (with side effect of message transfer) or failure. Calling sequences for a given site are represented by *routing tables* that indicate neighbor(s) the site is to call for a given dissemination task. For message transfer under single site failure, we will need *preferred* and *alternate* neighbors.

Below, we define a message transfer procedure that implements the calling scheme to be followed by each site:

```
procedure transfer(self, destination, message):
    if [call(message, preferred(self, destination)) indicates failure]
        then call(message, alternate(self, destination)).
```

For each site, we need to determine values for preferred and alternate neighbors to call for every other site as eventual destination. We assign routing table entries as the associated immune graph is built. For twin graphs, we add routings with each new vertex, as follows:

For the 4-cycle with vertices numbered as shown in figure 4, we determine the routings as shown there. We use the entry "fail" for alternate calls where the preferred call is routed directly to the destination.



pref(1, 2) = 2
alt(1, 2) = fail
pref(1, 3) = 3
alt(1, 3) = fail
pref(1, 4) = 2
alt(1, 4) = 3
.
etc
.

Figure 4. Routings for the 4-cycle.

As a vertex v is added, connected to twin neighbors x and y, we assign the following, additional, routing table entries:

preferred(v, x) = x; alternate(v, x) = fail; preferred(v, y) = y; alternate(v, y) = fail;
preferred(x, v) = v; alternate(x, v) = fail; preferred(y, v) = v; alternate(y, v) = fail;
for all u, $x \neq u \neq y$:   preferred(v, u) = x; alternate(v, u) = y;
preferred(u, v) = preferred(u, x);
alternate(u,v) =
if [alternate(u, x) = fail] then y else alternate(u, x).

If alternate(u, x) = fail, then u is a neighbor of twin pair x and y. Preferred and alternate calls always are assigned vertices from a neighboring twin pair. Lemma 8 will assist us in verifying that our protocol realizes the desired, self-repairing communication behavior.

Lemma 8: Each vertex in a twin graph G is equidistant from every other pair of twins in G.

Proof : By induction on number of vertices in G. We see our hypothesis is true for the 4-cycle. Suppose it is true for all twin graphs having k or fewer vertices (k>4) and consider twin graph G with k+1 vertices. G must have at least one vertex v of degree 2, adjacent to twin pair x and y. Removing v, we have a twin graph G' for which our hypothesis holds by inductive assumption. Consider another twin pair, s and t, of G'. These vertices are equidistant from x with distance d, and similarly from y at distance d'. As a twin pair, x and y are equidistant from any other vertex in G'; thus, d=d'. In G, vertex v is at distance 1 both x and y and thus equidistant from any other twin pair in G. As any twin of v in G is also only directly connected to x and y, all vertices in G are also equidistant from any such new twin pairs. Connecting v to x and y does not change the distance between x and y; so, addition of v provides no "shortcuts", ensuring that distances between vertices already in G' do not change in G with the addition of v. []

Theorem 9. The routing tables for a twin graph make the corresponding network self-repairing.

Proof: By induction on number of vertices, we see our routing tables direct preferred and alternate calls toward a destination through a pair of twins that are each one step closer to the destination. With Lemma 8, we see this is sufficient to realize the desired, self-repairing communication behavior. []

Next we consider routing table entries determined at application of the doubling operation. Assuming we are doubling graph G for which we have determined routing table

Conclusion

In this paper, we have provided a complete specification of a class of communication networks that are not only immune to single site failures, but maintain the ability to communicate between reminaing, operable sites with no additional delay (i.e., no increase in length of communication path). Our specification includes a constructive characterization of a subclass of minimal self-repairing graphs and algorithmic determination of associated routing tables used by a simple message transfer procedure to realize the desired immune behavior. Our work on distance penalties for message transfer contrasts with recent work on finding graphs that have spanning subgraphs approximating distances in the original graphs [5].

Open research questions for future work include the constructive specification of IP(k) networks, wherein bounded delay is acceptable, i.e., k>0. As a first step, by generalizing Corollary 1, we have that a graph G is an IP(k) graph iff, every vertex v of G and every pair of neighbors of v, x and y, form part of a cycle of length less than or equal to k+4. One can also generalize the problem along the dimension of number of failures, finding self-repairing graphs that are immune to m failures for m>1, while incurring no communication delay. In particular, one might investigate the case where multiple failures are isolated (i.e., not involving neighboring elements of the network) [1]. Finally, different communication tasks could be considered, such as networks whose broadcast behavior is immune to failures [2].

References

[1] Farley, A.M., "Networks immune to isolated failures", *Networks*, 11, p255-268 (1981).

[2] Farley, A.M. and Proskurowski, A., "Reliable minimum-time broadcast networks", *Proc.of Eighteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing, Congressus Numerantium*, 59, p37-48 (1987).

[3] Farley, A.M. and Proskurowski, A., "Minimum-size self-repairing graphs", submitted.

[4] S.M. Hedetniemi, S.T. Hedetniemi, and A.L.Liestman, "A survey of research on gossiping and broadcasting", *Networks*, 18, p319-359, (1988).

[5] Peleg, D. and Schaffer, A.A., "Graph spanners", *J. Graph Theory*, 13, p99-116, (1989).

entries already described above. We will use the routings in G when assigning routings for the doubled graph. We will refer to vertices v', v" and d', d" in double(G), which are understood to be copies of vertices v and d in G. We assign routing table entries, as follows:

```
/* First assign calls within a component */
    preferred(v', d') = preferred(v, d)'; preferred(v", d") = preferred(v, d)";
    alternate(v', d') = alternate(v, d)'; alternate(v", d") = alternate(v, d)";
/* Then assign calls between components */
    if (v = d)  then            /* v' and d" at distance 1 */
                preferred(v', d") = d"; preferred(v", d') = d';
                alternate(v', d") = fail; alternate(v", d') = fail
    if [preferred(v, d) = d] then          /* v' and d" at distance 2 */
                preferred(v', d") = d'; preferred(v", d') = d";
                alternate(v',d") = v"; alternate(v", d') = v';
    otherwise,  preferred(v', d") = preferred(v', d');
                preferred(v", d') = preferred(v", d");
                alternate(v', d") = alternate(v', d');
                alternate(v", d') = alternate(v", d").
```

Theorem 10: The routing tables for the doubled graph double(G) make the corresponding network self-repairing.

Proof: The routings in double(G) follow the previously determined, self-repairing routings when senders and receivers are in the same component. For those in different components, it follows the routing within the originator's component until reaching a site at distance 2 from the destination. That site then first attempts to call the site corresponding to the ultimate destination in its own component; if that fails, the site calls its corresponding vertex in the other component, which is also at distance of 1 from the destination. One of these must succeed. As such, the routing maintains the immune, self-repairing property as required. []

Finally, we consider routing table entries to be assigned at application of the vertex-cloning operation. Assuming we form a new self-repairing graph G' by adding vertex v' cloning vertex v of self-repairing graph G, for which we have routing table entries. Routings to other destinations from v' are assigned to follow the self-repairing routings already determined from v in G. Routings from other sites to v' follow the self-repairing routings to v until reaching the neighborhood of v, from which v' is called directly. Message transfers between v' and v utilize their common neighborhoods to realize the desired immune behavior.

Minimum Self-Repairing Graphs

Arthur M. Farley and Andrzej Proskurowski*
Computer and Information Science Department
University of Oregon
Eugene, Oregon 97403, USA

August 17, 1993

Abstract

A graph is self-repairing if it is 2-connected and such that the removal of any single vertex results in no increase in distance between any pair of remaining vertices of the graph. We completely characterize the class of minimum self-repairing graphs (which have the minimum number of edges for a given number of vertices) and establish certain algorithmic properties of such graphs.

1 Preliminaries

In this paper, we investigate the class of graphs which we will call *self-repairing*. These are graphs from which the removal of a single vertex results in no increase in distance between any pair of remaining vertices. Specifically, we show that a minimum number of edges in a self-repairing graph with n vertices is $2n-4$ and determine the class of all such minimum size self-repairing graphs.

By a *graph* we will understand a simple undirected graph $G = \langle V, E \rangle$, consisting of a set V of vertices and a set E of edges, each edge connecting

*Research supported in part by National Science Foundation grants NSF-CCR-9213439 and NSF-INT-9214108.

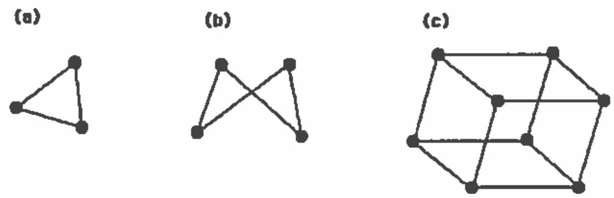


Figure 1: (a) The three-cycle, (b) the four-cycle (also $K_{2,2}$), and (c) the cube.

a pair of vertices (v_1, v_2). For a given vertex v, let $i_G(v)$ represent the set of edges incident to v in graph G, *i.e.*, those for which v is a member of the edge pair. The degree of a vertex v in graph G is equal to the number of elements in $i_G(v)$. For a given vertex v of graph G, let $N_G(v)$ denote the *open neighborhood* of v, being the set of vertices with which v shares incident edges in G. Let $N_G[v]$ denote the closed neighborhood of v, being $N_G(v) \cup \{v\}$.

A *path* between two vertices x and y is a sequence of edges $(v_1, v_2), \ldots, (v_{n-1}, v_n)$ such that $v_1 = x$ and $v_n = y$ and for $1 \leq i < j \leq n$, $v_i \neq v_j$. When this constraint is violated for $i = 1$ and $j = n$ (*i.e.*, $x = y$), such a sequence of edges defines a *cycle*. The *length* of a path is the number of edges in the path. For a given pair of vertices in G, u and v, let the distance between them, $dist_G(u, v)$, be the length of a shortest path from u to v in G. A graph is *connected* if there exists a path between every pair of vertices. A graph is *2-connected* if for every vertex v of G, the graph $G' = \langle V - v, E - i_G(v)\rangle$ is connected. In other words, removing any vertex of G does not disconnect the graph. As such, self-repairing graphs are a subset of the 2-connected graphs.

In the following, we will refer by name to the *three-cycle* (cycle of three vertices), the *four-cycle* (cycle of four vertices), the *cube graph* (the 8 vertex, 12 edge skeleton of the cube) and the complete bipartite graph $K_{2,k}$ having $k + 2$ vertices, 2 of which are adjacent to k other vertices, and no other adjacencies (see Figure 1).

We first establish a property of the neighborhoods of vertices in a self-repairing graph.

Theorem 1: A graph G is self-repairing if and only if, for every vertex v and for each pair of vertices u and w in $N_G(v)$, $N_{G'}[u] \cap N_{G'}[w] \neq \emptyset$, where $G' = \langle V - \{v\}, E - i(v)\rangle$.

Proof: Let v be the vertex removed from G to form G'. For a pair of vertices, x and y in G', such that no shortest path between them in G includes v, $dist_G(x,y) = dist_{G'}(x,y)$. However, if there exists a shortest path in G between x and y that includes v, then we must show that there exists a path in G' of the same length. The shortest path through v in G contains edges (u,v) and (v,w), for some vertices u and w in $N_G(v)$, where there are zero or more vertices between u and x and between y and w. By our assumption as to the non-empty intersection of neighborhoods of u and w in G', there exist edges (u,v') and (v',w) in G' (u and w are not adjacent because the (x,y) path has the shortest length). Thus, $dist_G(x,y) = dist_{G'}(x,y)$.

If our assumption as to non-empty intersections is not true, *i.e.*, there exist vertices u and w that are neighbors of a vertex v in G that do not share a vertex in their closed neighborhoods in G' after the removal of v, then $dist_{G'}(u,w) > dist_G(u,w) = 2$. ■

We can restate Theorem 1 as the following corollary in terms of the maximum possible length of a minimum cycle involving a pair of neighbors of any vertex in a self-repairing graph.

Corollary 1: G is a self-repairing graph if and only if, for each vertex v in G and for each pair of vertices x and y in $N_G(v)$, vertices v, x, and y are members of a cycle of length at most 4 in G, *i.e.*, a three- or a four-cycle.

2 Main result

We want to characterize self-repairing graphs with minimum number of edges. We first define the class of twin graphs that are self-repairing .

We call two vertices x and y in a graph G *twins* if and only if $N_G(x) = N_G(y)$, where the neighbor set is non-empty. Based upon this notion of twin vertices, we define *twin graphs* recursively, as follows: (i) the four-cycle is a twin graph; (ii) if G is a twin graph, then the graph G' constructed by connecting a new vertex by two edges to a pair of twins in G is a twin graph.

Note that, when a new vertex is connected to twins x and y, vertices x and y remain twins in G'. In fact, once a pair of twin vertices have degree higher than 2, the two vertices remain unique twins of each other in any twin graph constructed from this smaller graph.

Figure 2: (a) The level graph of a twin graph wrt. a degree 2 vertex, and (b) the corresponding free tree.

Fact 1: In a twin graph, vertices of degree greater than 2 occur in uniquely defined pairs of twin vertices. ∎

Theorem 2: A twin graph is self-repairing.

Proof: Let G be a twin graph. Each vertex has degree 2 or higher. Let a vertex v of degree 2 be removed from G to form G'. In G, v is connected to twin vertices x and y, which, by definition, share a non-empty neighborhood of vertices in G'.

Now, consider v having degree greater than 2. Such a vertex v has a unique twin vertex w (Fact 1). Every neighbor of vertex v in G is thus also a neighbor of w. Thus, if v is removed to form graph G', the intersections of neighborhoods in G' of all pairs of vertices in $N_G(v)$ include w.

Thus, G is self-repairing, by Theorem 1. ∎

Twin graphs with n vertices have $2n - 4$ edges since the four-cycle has $2n - 4$ edges and 2 new edges are added for each new vertex connected to the graph. We will show that, together with the cube graph, they constitute exactly the class of minimum size self-repairing graphs with n vertices.

In our proofs of the tight lower bound on the number of edges in a self-repairing graph, we will use the notion of *level graph* of a given graph G with respect to a fixed vertex x of G. Vertices of graph G are arranged in levels, depending on their distance from the vertex x. Given a vertex x in G, we assign a *level* to each vertex by defining *level*$(x) = 0$ and *level*$(y) = i > 0$ to every vertex y at distance i from x (see Figure 2(a)).

Lemma 1: In a level graph of a self-repairing graph, any vertex at level $i > 1$ is adjacent to at least two vertices at level $i - 1$.

Proof: Consider a vertex v at level $i > 1$. It must have a neighbor x at level $i - 1$, which in turn has a neighbor y at level $i - 2$. Since v, x and y are in a four-cycle, v must have another neighbor at level $i - 1$. ■

Together with the fact that self-repairing graphs are 2-connected, Lemma 1 gives us the following lemma.

Lemma 2: In a level graph of a self-repairing graph, each level $i > 0$, except for the maximum level, contains at least two vertices. ■

We want to prove that there are no self-repairing graphs of n vertices with fewer than $2n - 4$ edges. We will proceed by assuming the contrary and proving that such a graph cannot have vertices of degree 2. This, in turn, will be used to show that no such graphs exist.

Lemma 3: A self-repairing graph G with n vertices and fewer than $2n - 4$ edges has no degree 2 vertices.

Proof: Assume that there is a degree 2 vertex x in G. We will consider the levels defined *wrt.* x. By Lemma 1, every vertex on level $i > 1$ is adjacent to at least 2 vertices on level $i - 1$, which totals at least $2(n - 3)$ edges. The vertex x and its neighbors induce at least 2 edges. Thus G has at least $2n - 6 + 2 = 2n - 4$ edges, a contradiction. ■

Not having any degree 2 vertices, a self-repairing graph G with n vertices and fewer than $2n - 4$ edges must have some degree 3 vertices. We will show that this cannot happen. Thus, there are no such graphs.

Lemma 4: In a level graph of a self-repairing graph G with n vertices and fewer than $2n - 4$ edges, where levels are defined *wrt.* a degree 3 vertex, every vertex at level $i > 1$ has exactly two neighbors at level $i - 1$ and no neighbors at level i.

Proof: By Lemma 3, such a graph G does not have a degree 2 vertex and thus must have a degree 3 vertex x. We will consider the levels defined *wrt.* x. Lemma 1 requires that there are at least two level $i - 1$ neighbors of a level $i > 1$ vertex; thus the $n - 4$ vertices of $G - N[x]$ account for at least $2n - 8$ edges. This is also the maximum number of edges incident with vertices at levels $i > 1$ if G is to have fewer than $2n-4$ edges, since at least 3 edges are induced by $N[x]$. Thus every vertex at level $i > 1$ has exactly two neighbors at level $i-1$ and no neighbors at level i. ∎

Theorem 3: The minimum number of edges in a self-repairing graph of n vertices is $2n-4$.

Proof: In a self-repairing graph G with n vertices and fewer than $2n-4$ edges, every vertex at level $i > 1$ has exactly two neighbors at level $i-1$ and no neighbors at level i (by Lemma 4). This implies that vertices on the last level defined *wrt.* x have degree 2. This contradict the result of Lemma 3, which states that there are no degree 2 vertices in G. Thus, there are no self-repairing graphs with n vertices and fewer than $2n-4$ edges. Twin graphs are self-repairing and have $2n-4$ edges. ∎

We have seen that twin graphs are self-repairing and reach the lower bound on the number of edges. We will show that, except for the cube graph, there are no other minimum self-repairing graphs. Here again, we will consider level graphs of minimum self-repairing graphs defined *wrt.* a vertex of degree 2, or 3.

Lemma 5: A minimum self-repairing graph having a degree 2 vertex is a twin graph.

Proof: Assume the contrary and consider a smallest minimum self-repairing graph G with a degree 2 vertex x that is not a twin graph. In the level graph of G defined *wrt.* x, every vertex at level $i > 1$ is adjacent to exactly 2 vertices at level $i-1$ and has no neighbors at level i. By Lemma 1, the number of neighbors at level $i-1$ for each such vertex is at least 2, and x is adjacent to two edges, which accounts for the total of $2(n-3)+2 = 2n-4$ edges. As such, the two neighbors of x have identical sets of neighbors and thus are twins; x and its two incident edges can be removed to give G'. It is easy to see that G' is a minimum self-repairing graph. Also, G' has a degree 2 vertex (as discussed in the proof of Theorem 3). Thus, by our asumption, G' is a twin graph and so is G, a contradiction. ∎

Lemma 6: A minimum self-repairing repairable graph without degree 2 vertices has no three-cycle involving a degree 3 vertex.

Proof: Assume to the contrary that three vertices induce a cycle. Consider the levels defined *wrt.* a degree 3 vertex x from this cycle. The remaining $n-4$ vertices account for at least $2n-8$ edges which together with the 4 edges induced by x and its neighbors gives $2n-4$ edges. Thus, the vertices at the last level must have degree 2, a contradiction. ∎

Lemma 7: The only minimum self-repairing graph with no degree 2 vertices is the cube graph.

Proof: Such a graph G must have at least 8 degree 3 vertices. Consider the level graph of G defined *wrt.* a vertex of degree 3. Only one vertex at any of the levels $i > 1$ can have 3 neighbors on level $i-1$, or only two vertices at the same level $i > 1$ can be neighbors. By the absence of degree 2 vertices, this must apply only to the last level, k. If there are two adjacent vertices of degree 3 at level k, then they must be adjacent to four different level $k-1$ vertices (by Lemma 6). This and the absence of other edges between vertices at the same level (Lemma 4) violates the requirement that two adjacent edges in a self-repairing graph belong to a four-cycle.

Assume that the last level k consists of exactly one degree 3 vertex. By symmetry, every vertex at level i, $0 < i < k-1$, has exactly two neighbors at level $i+1$. This implies that all intermediate levels have 3 vertices with the same pattern of adjacencies with vertices on the neighboring levels. By inspection, vertices on three (or more) intermediate levels violate the self-repairing property and thus $k \leq 3$. For $k = 2$ we have $K_{2,3}$, a twin graph, and for $k = 3$ we have the cube graph. ∎

Lemmas 6 and 7 imply our main result.

Theorem 4: A minimum self-repairing graph is either a twin graph or the cube graph. ∎

3 Algorithmic properties

We have presented a constructive characterization of the class of minimum self-repairing graphs. With one exception, the class is identical with the class of twin graphs, which are thus useful as graphs underlying communication networks immune to certain element failures ([5, 6]). In this section, we also characterize twin graphs by an efficient recognition algorithm and propose a simple enumeration scheme for these graphs that is based on a bijection with the family of free trees. This bijection reflects a tree-like structure of twin graphs, which we show here to have treewidth not greater than 3, and thus belong to a well-known class of graphs with good algorithmic properties (see [1, 2]).

3.1 Recognition of twin graphs

We have completely characterized minimum self-repairing graphs by their construction process. This characterization leads to a simple and efficient recognition algorithm by means of a graph reduction system. Such a system reverses the iterative construction process discussed above.

Theorem 5: A twin graph can be recognized in time proportional to the number of vertices in the graph.

Proof: The only reduction rule is the removal (deletion) of a degree 2 vertex (and its incident edges) adjacent to twin vertices, as long as the graph has more than 4 vertices. The rule is generic in the sense that the degree of the twins is not bounded. The original graph is recognized as a twin graph, if and only if its irreducible reduct is the four-cycle. This algorithm can be implemented to execute in time proportional to the number of vertices of the input graph. Degree 2 vertices are accessible in constant time through an initial degree count and incremental maintenance of "current degree". Checking of the twin property (the equality of the neighborhoods of two vertices) requires examination of the graph's adjacency lists (which have linear size) at most twice per edge. ■

There is an additional advantage of the twin graph recognition algorithm, pertaining to its use as the topology of a self-repairing communication network ([6]). Since the recognition process reverses the iterative construction process, if the self-repairing communication protocol (routing tables) is not established with the graph's construction, it is possible to create the correct entries as a result of the recognition process.

3.2 Enumeration of twin graphs

We will construct a bijection between twin graphs and free trees. Due to this relation, a systematic enumeration of twin graphs can be performed by enumeration of free trees, see for instance [4].

The bijective relation between the set of free trees with m internal nodes and k leaves, and the set of twin graphs with k degree 2 vertices and $2m$ vertices of higher degree is established as follows.

Given a twin graph G with at least 5 vertices, the corresponding tree $T(G)$ is defined by the set of nodes and the set of edges incident to the nodes. A degree 2 vertex x in G defines a leaf of $T(G)$ that is adjacent to the node representing the pair of twin neighbors of x in G. A pair of twin vertices of degree at least 3, say x and y of G, defines an internal node of the tree $T(G)$. By Fact 1, this determines uniquely the set of nodes of $T(G)$. The node corresponding to the pair of twin vertices x and y is adjacent to every node in $T(G)$ that corresponds to a pair of twin vertices of degree at least 3, say u and v in G such that the sequence x, u, y, v constitutes a four-cycle without chords in G. If G has k degree 2 vertices and $2m$ vertices of higher degree, then $T(G)$ has k leaves and m internal nodes (*cf.* Figure 2).

Conversely, consider a tree T with at least four nodes. An internal node of T will define a pair of (twin) vertices of the corresponding twin graph $G(T)$. A leaf will define a degree 2 vertex of $G(T)$. The adjacency between two internal nodes X and U that correspond to two pairs of vertices of $G(T)$, x and y, and u and v determines the four edges in $G(T) : (x,u), (x,v), (y,u)$ and (y,v). A leaf node Z of T adjacent to an internal node X corresponds to a degree 2 vertex z of $G(T)$ adjacent to the (twin) vertices x and y corresponding to X.

To see that $G(T)$ is a twin graph, we will construct it according to the twin graph construction rules. If T has only one internal node and k leaves, it corresponds by this definition to the twin graph $K_{2,k}$. If T has at least two internal nodes, then by considering an arbitrary pair of adjacent internal nodes X and U as representing vertices of the initial four-cycle of $G(T)$ and proceeding to construct $G(T)$ by adding vertices of $G(T)$ corresponding to nodes of T adjacent to those already regarded nodes, we get $G(T)$.

3.3 Treewidth of twin graphs

Existence of a tree describing the structure of a given twin graph indicates the possibility of an efficient algorithmic treatment of these graphs. We give an indirect proof that many optimization problems, inherently difficult for general graphs, can be solved efficiently when restricted to twin graphs. We exploit here the concept of the treewidth of a graph, $tw(G)$, or rather a related concept of a *partial k-tree*: for a given integer k, graph G is a partial k-tree

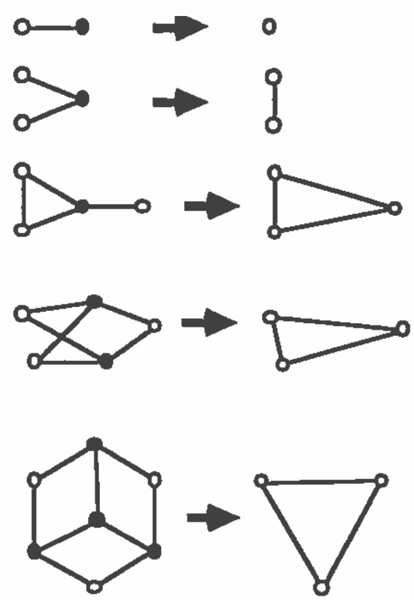


Figure 3: Reduction rules for recognition of partial 3-trees.

if and only if $tw(G) \leq k$. Such graphs have good algorithmic properties, as mentioned above, and for $k \leq 4$, there are recognition algorithms of linear time complexity. We refer the reader unfamiliar with these concepts to the relevant references ([1, 2, 3]) and only sketch the proof here.

In a graph $G = \langle V, E \rangle$, a *minimal separator* $S \in V$ is a smallest subset of vertices such that $G' = \langle V - S, E - \cup_{v \in S} i_G(v) \rangle$ is not connected. A k-tree is a connected graph such that any minimal separator induces a complete subgraph K_k. Partial k-trees are subgraphs of k-trees.

Connected partial 3-trees can be recognized by using the reduction rules in Figure 3. Applying a reduction rule consists of finding a subgraph isomorphic to the left-hand side of the rule, such that the vertices corresponding to the bold vertices of the rule have exactly the adjacencies indicated, and replacing it by a subgraph isomorphic to the right-hand side of the rule. A finite number of reductions transforms a connected partial 3-tree (and only such graphs) to the trivial graph ([3]). We will show that a twin graph can be always reduced in such a manner. The *intermediate* graphs obtained through reduction of twin graphs differ from twin graphs in that some twins will become adjacent.

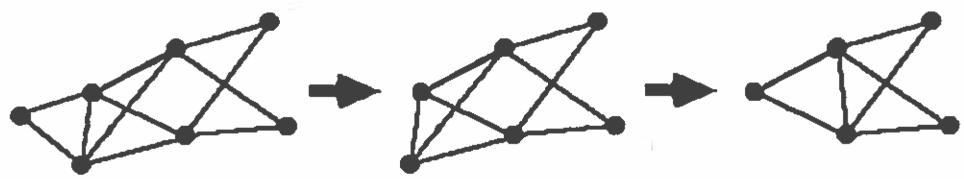


Figure 4: The result of two series reductions (first step) and a triangle reduction (second step) applied to the twin graph in Figure 1.

A degree 2 vertex of a twin graph G can be reduced according to the *series* rule (second rule of Figure 3). This reduction introduces an edge between the neighbors of the reduced vertex which are twin vertices in G.

Consider an intermediate graph G in which all degree 2 vertices adjacent to the same pair of (now adjacent) twin vertices, say u and v, have been reduced (this is illustrated in the first reduction step Figure 4). Vertices u and v have degree 3 in G and $N_G[u] = N_G[v]$. Thus, one of these vertices, say u, can be reduced according to the *triangle* rule (third rule of Figure 3). The reduction results in a graph G' with a degree 2 vertex corresponding to v adjacent to the two neighbors (in G) of u and v which are (possibly adjacent) twins in G and are now connected (second step in Figure 4). Iterating this process, a twin graph is reduced (according to the rules in Figure 3) to an edge graph (K_2). The edge graph can be reduced to the trivial graph by the first rule of Figure 3, showing that $tw(G) \leq 3$. Thus, we have established the following fact.

Theorem 6. Twin graphs have treewidth at most 3.

Actually, only twin graphs $K_{2,i}$, $i > 2$, (represented by trees of star shape) are partial 2-trees. (This follows from the property of partial 2-trees which excludes subgraphs homeomorphic to K_4.) We notice that, though most twin graphs have treewidth 3, their minimal separators are of size 2. Thus, many optimization algorithms for twin graphs, guided by the tree corresponding to the input graph, should be simpler (more efficient) than the corresponding algorithms for partial 3-trees.

References

[1] S. Arnborg, J. Lagergren, and D. Seese, Easy problems for tree-decomposable graphs, *J. Algorithms* (1991), 308-340;

[2] S. Arnborg and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, *Discrete Applied Mathematics 23* (1989), 11-24;

[3] S. Arnborg and A. Proskurowski, Characterization and recognition of partial 3-trees, *SIAM Journal of Algebraic and Discrete Methods 7*, 2(1986), 305-314;

[4] T. Beyer and S.M. Hedetniemi, Constant time generation of rooted trees, *SIAM J. Computing 9* (1980), 706-712;

[5] A.M. Farley and A. Proskurowski, Immunity to subgraph failures in communication networks, *Proceedings of the 18th South–Eastern Conference on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, Winnipeg, Congressus Numerantium 62 (1988), 49-58;

[6] A.M. Farley and A. Proskurowski, Self-repairing networks, *UO-CIS-TR-93-18.*