PROBABILISTIC ESTIMATION OF DAMAGE FROM FIRE SPREAD

Charles C. Colbourn, Louis D. Nel,
T.B. Boffey and D.F. Yates

SCS-TR-189, APRIL 1991

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

Probabilistic Estimation of Damage from Fire Spread

Charles J. Colbourn
Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario
CANADA N2L 3G1

Louis D. Nel
Computer Science
Carleton University
Ottawa, Ontario
CANADA K1S 5B6

T. B. Boffey and D. F. Yates
Statistics and Computational Mathematics
University of Liverpool
Liverpool, England
U.K.

Abstract

Efficient methods for the probabilistic assessment of damage from fire spread and other invasive hazards in segmented structures are developed. The methods exploit a basic relationship between the fire spread problem and the probability of reachability in communications networks. A novel efficiently computable upper bound is developed for reachability and for fire spread probability, using noncrossing cuts.

1 Invasive Hazards in Segmented Structures

Buildings and other segmented structures are subject to numerous hazards; among these are *invasive* hazards such as fire spread and contamination. Typically, buildings are partitioned into *volumes* (rooms, corridors, and so on); in addition to structural and utilitarian reasons, this is done to reduce damage by invasive hazards. A basic reason for limiting such damage is naturally to prevent damage to the building itself. However, it is often more important, than merely to protect the building, to keep active certain critical functions housed within the building. For example, if the building houses the control centers for a nuclear reactor, or an air traffic control center, or an emergency dispatch center, the consequences of damage by an invasive hazard depend primarily on its interruption to the *function* performed within the building, rather than the (coincidental) damage to the structure.

Naturally, if a building houses such an important function, we expect that there are *critical components* or subsystems that are replicated in the system to prevent failure through the loss of one subsystem. In general, we consider a building in which there is a critical subsystem replicated in a number of different volumes of the building. The failure of one or more of these subsystems may adversely affect performance, but typically only the failure of *all* of the replications of the critical component is deemed catastrophic.

Our concern in this paper is to estimate the probability with which an invasive hazard, such as fire, starts in a building and spreads in such a way as to disable all critical components. An extensive literature exists on the growth of fire within a compartment, flashover and spread to other compartments. Stochastic and probabilistic methods have been widely advocated as supplements, or even as replacements, for the standard tables used in building codes and insurance; see, for example, [4, 5, 8, 13, 14]. The stochastic models studied are concerned with fire spread when there is an active effort to extinguish the fire, and hence are of necessity time-dependent models. However, our concern is primarily with the spread of fire in the absence of effective countermeasures, enabling us to view the fire spread as instantaneous. We establish that this concern is a very natural complement to a main problem arising in communication networks: determining the probability that a message can be delivered to all of its intended recipients in the network. In order to develop this correspondence, we first describe the problem with invasive hazards more precisely. We use nomenclature for fire spread, but emphasize that the problem concerns invasive hazards more generally.

A *segmented structure* is a set of *volumes* $v_1, \ldots, v_n$ and an adjacency relation between neighbouring volumes; examples are buildings, ships and aeroplanes. The adjacency relationship is arbitrary, but is typically taken to be the usual geometric adjacency, such as sharing a wall in a building. Each volume has an *ignition frequency*, which is the frequency with which a fire begins in the volume; we use the *ignition probability* ip_i of volume v_i to be the probability that a fire begins within a specified time period (chosen in such a way that the probability that two independent fires ignite within the time period is negligible). The ignition probability is taken to be the probability that a "significant" fire erupts in the volume (insignificant fires are those that *a priori* have no chance of establishing themselves in the volume.) Once a significant fire is initiated in a volume, it may or may not become *established* to the point where it threatens the barriers of the volume. Each volume v_i has a *spread probability* p_i that a fire entering the volume or igniting there becomes established. An established fire in a volume renders all critical components in the volume inoperative. In addition, an established fire may *breach* the barrier into a neighbouring volume (or many barriers into neighbouring volumes). For every two neighbouring volumes v_i and v_j, the *breach probability* p_{ij} is the probability that an established fire in volume v_i enters volume v_j. It may or may not become established in volume v_j. Once a volume has been breached, the spread probability determines whether or not a fire becomes established; a subsequent breach of a second barrier into the volume does *not* alter this probability. This requirement can be accommodated by a "single fire entry assumption" that every volume is only entered once; however, in order to avoid the time-dependency this necessitates, we instead stipulate that, while a volume may be breached more than once, the spread probability depends only on whether or not a breach into the volume occurs. Naturally the determination of the probabilities used in the model is itself a difficult problem; arguably it is more complex to obtain reasonable breach and spread probabilities than to determine the extent of fire spread given these probabilities. These probabilities have, however, been studied in some depth; for spread probabilities, see for example [16], and for breach probabilities see [9] and references therein.

The building houses k *critical components* $C_1, ..., C_k$. A volume may contain many components, and a component may reside in many volumes. One example of importance is when the "components" are escape routes from the building. A component is *inoperative* if *any* volume in which the component resides has an established fire.

Under the assumption that the ignition, spread and breach probabilities are all independent, we are interested in the probability that all critical components are rendered inoperative. In general, we may also ask what the expected number of operative critical components is, or what the probability that at least t of the k components remaining operative is.

It appears that the probability that a fire spread damages critical components depends on the spread of the fire through the structure over time. In particular, to determine whether or not a fire is established in a volume, we must determine whether the volume is breached. Nevertheless, we can adopt a static view that is equivalent in terms of probabilities to the dynamic view of fire spread. A *random state* of the structure is obtained by determining:

1. for each volume, whether or not a fire ignites in the volume;

2. for each volume, whether or not a fire becomes established in the volume, *given* that a fire enters the volume; and

3. for each barrier, say (x, y), whether or not the fire breaches the barrier *given* that a fire is established in volume x.

In determining a random state, each decision can be made independently using the known ignition, spread and breach probabilities. Hence the probability of a specific state can be easily computed. It is important to note that, while we assume knowledge of the conditional probabilities precisely as part of the formulation of the problem, we do not know with what probability a specific barrier is actually breached, or a specific volume contains an established fire.

The specific fire spread probability of concern to us is the probability that a state chosen randomly with the specified probabilities permits a fire to reach all critical components. In a dynamic view, we "start a fire" and determine with what probability the fire spreads through portions of the structure; in the static view, we decide *a priori* what the outcomes will be if a fire enters a volume, or attempts to breach a barrier, and then examine what fire spread would actually occur for such a set of outcomes. While the first mirrors our physical understanding of fire spread more closely, the second is equivalent and removes the apparent time dependency. In fact, using this static view, we establish a close connection between fire spread and reachability in communications networks, a well-studied problem. In order to explore this connection, our first task is to develop a network representation for segmented structures.

2 Network Models

A convenient representation for a building or other segmented structure is a directed network (see, e.g., [3]). The *nodes* of the network are the volumes $\{v_1, ..., v_n\}$. Between any two neighbouring volumes v_i and v_j, there is an *arc* (v_i, v_j). Each node v_i has an associated ignition probability ip_i and spread probability p_i, while each arc (v_i, v_j) has an associated breach probability p_{ij}. In general, we do not assume that $p_{ij} = p_{ji}$, and hence we treat directed networks. Components $C_1, ..., C_k$ are associated with subsets $V_1, ..., V_k$ of the network nodes. The network model is far from unique. A physical volume may be partitioned for convenience into a number of virtual volumes among which the breach probabilities are unity, changing the network model.

While this network model is a close analogue of the physical structure, the presence of three kinds of probabilities, and subsets representing the components, is somewhat

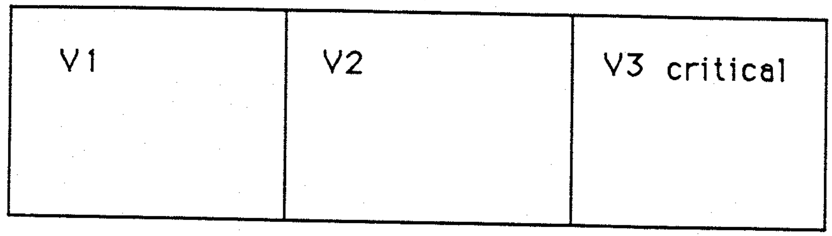

fig. 2.1a

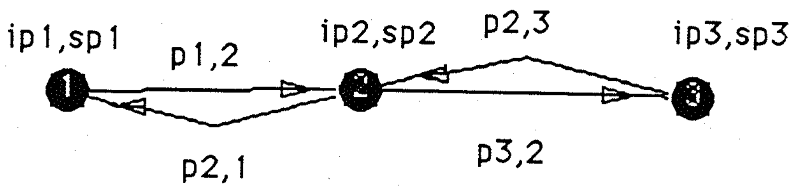

fig. 2.1b

Figure 2.1: Fig. 2.1a shows a segmented structure with one critical component V3. Fig. 2.1b shows the resulting network model.

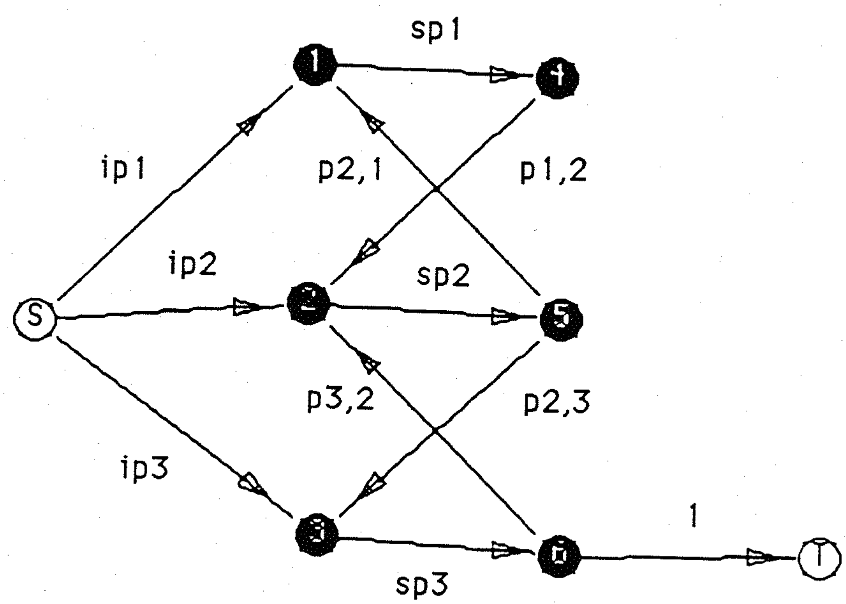

Figure 2.2. The network reliability model.

inconvenient. We therefore modify the network model in a number of steps to arrive at a model whose description is simpler.

First, we introduce a new node w_j for each component C_j; w_j has ignition probability zero, and spread probability one. For each $v_i \in C_j$, we add an arc (v_i, w_j) having breach probability one. To maintain the physical analogy, we are treating each component C_j as a volume that never initiates fires, but develops an established fire whenever one of the volumes containing the component does.

In the original problem, there are n potential sources of fires, namely the n volumes. We can introduce a single source of fire, that we call f, whose ignition probability is one (it is always on fire). Then for each volume v_i, add an arc (f, v_i) whose breach probability is the *ignition* probability of v_i. All ignition probabilities on the nodes are eliminated thereafter. If we treat the network as *always* having an established fire at node f, the probability that a fire commences next at node v_i is precisely the original ignition probability of v_i, as required.

Next we transform spread probabilities into breach probabilities as well. Split every node v_i into two nodes x_i and y_i, replacing all arcs of the form (a, v_i) by (a, x_i), and all arcs of the form (v_i, b) by (y_i, b). Then add an arc (x_i, y_i) whose breach probability is the spread probability of node v_i, and eliminate all spread probabilities.

The resulting network has only breach probabilities, but it is an easy exercise to see that the probability of fire spread to a component is the same as in the original network. Moreover, the original assumption about independence of the probabilities translates into statistical independence of all breach probabilities in this transformed network. We henceforth assume that this sequence of network transformations has been done, and call the resulting network model of the building its *structural network*. See Figures 2.1 and 2.2.

In the structural network, we need not distinguish between fires that "enter" a volume and fires that are "established" in a volume; this distinction rests only on whether the entering arc to the node represents an original breach probability or an original spread probability.

Now we turn to fire spread probabilities. A *random state* S of the structural network $G = (V, A)$ is a subset $S \subseteq A$. Arcs in A are in four types:

1. *ignition arcs* (f, x_i) carrying original ignition probabilities;
2. *spread arcs* (x_i, y_i) carrying original spread probabilities;
3. *original arcs* (y_i, x_j) carrying original breach probabilities; and
4. *sink arcs* (y_i, w_k) carrying probability 1.

A random state S arises with probability $\prod_{a \in S} p_a \prod_{a \in A \setminus S}(1 - p_a)$; hence a random state of the structural network arises with the *same* probability as the corresponding random state of the structure. Here, by a random state of the structure, we do *not* mean the extent of actual fire spread, but rather the states of each volume with respect to ignition and spread, and each barrier with respect to whether it will be breached if a fire is present in the neighbouring volume.

Consider state $S \subseteq A$. If for some critical component $w \in V$, S contains an f,w-path, a fire *will* travel from f to w thereby rendering the component at w inoperative. On the other hand, if S has no f,w-path, *no* fire can reach w *notwithstanding* the fact that S may contain an arc into w. Call a state *T-reachable* if for every $t \in T$, the state contains an f,t-path. Then if $\mathcal{T}$ is the set of all T-reachable states, the fire spread probability $FS(G,T)$ is $\Sigma_{S \in \mathcal{T}} p(S)$.

To recapture the dynamic view of fire spread, we observe that a minimal state that is T-reachable is a directed tree rooted at f, in which every sink (node with no exiting arc) is in T; we call such a minimal T-reachable state a *T-fire tree.* $FS(G,T)$ accounts precisely for all states of the structural network containing at least one T-fire tree. Naturally, $FS(G,T)$ can be computed in a "dynamic" fashion by determining the probability that fire spread occurs along each of these fire trees; this requires that the probability of following a particular fire tree be corrected to account for the probability that other fire trees are *not* followed. More precisely, if a network has s T-fire trees $F_1, ..., F_s$, $FS(G,T)$ is the probability that at least one of these fire trees operates, which can be expressed by inclusion-exclusion as follows. Let $\mathcal{F}_i$ be the collection of all $\binom{s}{i}$ unions of T-fire trees. Then $FS(G,T) = \sum_{i=1}^{s}(-1)^{i-1} \sum_{F \in \mathcal{F}_i} \prod_{a \in F} p_a$.

3 Reachability

This formulation of fire spread probability bears a remarkable resemblance to a well-studied problem arising in communication networks. A *communications network* is a set of nodes representing network sites (communication processors), and a set of arcs representing undirectional communication links between nodes. A typical operation is for a processor to initiate a message and to attempt to communicate the message to a set of intended recipient nodes. Intermediate nodes can be used to forward messages not intended for them, and all nodes are assumed to be completely reliable. However, arcs fail randomly; each arc is assumed to operate independently with known probability. In general, the T-reachability, or s,T-connectedness, for an initiator node s and recipient nodes in T is the probability that a message originating at s reaches every node of T when arcs operate independently with the specified probabilities.

An analogy with the fire spread problem is immediate, with message spread through the network in place of fire spread. The connection is more than superficial: interpreting the structural network of a building as a communications network, with breach probabilities as link operation probabilities, critical components as message recipients, and the source f of fire as the initiator of the message, the fire spread probability is *precisely* the T-reachability.

Although we have established a direct correspondence between the two problems, some important differences remain. In a communications network, our primary objective in design is to *maximize* the probability that a message reaches its recipients, while in structural design our objective is to *minimize* the potential disruption by fire damage; hence communication networks typically have probabilities close to one for link operation probabilities, while any stable structure should have breach probabilities as near zero as possible. Similarly, in network analysis, for communications networks we are concerned with establishing strong lower bounds on the reachability in order to ensure adequate performance; for structural networks, we are instead concerned with establishing strong upper bounds, to ensure acceptably low risk of disruption via fire damage. For these reasons, although reachability in communications networks has been studied extensively, the fire spread problem poses different concerns.

Using results for s, T-connectedness as a prototype, we develop a practical method for assessing risk from fire spread. We refer the interested reader to [6] for a detailed survey on the s, T-connectedness problem, but recall the most relevant results as needed.

We first establish that the fire spread problem is computationally difficult. Let $G = (V, E)$ be an *undirected* network in which each edge $e \in E$ has the same probability p of being operational. $K \subseteq V$ is a set of *target nodes* that are required to participate in the communication. The *(k-terminal) reliability* $Rel(G, K; p)$ is the probability that, when each edge operates independently with probability p, there exist paths of operating edges between every two nodes of G. Provan [12] established that computing $Rel(G, K; p)$ is #P-complete even if G is planar and $|K| = 2$ (see [7] for undefined terms in computational complexity). Vertigan [17] established that computing $Rel(G, K; p)$ is #P-complete even if G is planar and $K = V$. We describe a simple transformation from the k-terminal reliability problem to the fire spread problem. First, we form a structure whose volumes correspond to the nodes of G, with two volumes of the structure adjacent if and only if the corresponding nodes are adjacent in G. Such a structure can always be constructed for a planar graph G by taking the volumes of the structure to be the faces of the planar dual of G; in fact, the structure so produced is a single storey structure, although admittedly its volumes may have irregular shapes and sizes!

In any undirected network with target nodes in K, the k-terminal reliability is identical to the probability that for any one node $w \in K$, there exist operating paths from w to each other node in K; that is, reliability for K and reachability for w to $K \setminus \{w\}$ are the same. In addition, the replacement of an undirected edge $\{x, y\}$ having probability p by two arcs (x, y) and (y, x) each having probability p leaves the reachability unchanged [6, 11], and hence we replace each edge by two arcs in antiparallel.

Now we determine probabilities to assign to the structure corresponding to the dual of G. We assign all spread probabilities to be 1 (since nodes do not fail). We assign all breach probabilities to be p. Every barrier $\{v_i, v_j\}$ corresponds to an edge of G; the breach probabilities $p_{ij} = p_{ji} = p$ correspond to a pair of arcs in antiparallel that is equivalent to the undirected edge. Next choose any $s \in K$ to serve as a "root"; assign it ignition probability one; assign all other nodes ignition probability zero. Finally, place a critical component at each of the nodes of K, except possibly s (the presence of a critical component at s is irrelevant in terms of fire spread probability). The probability of reachability from s to $K \setminus \{s\}$ is the fire spread probability in the structure; it is precisely $Rel(G, K; p)$.

Hence we have proved:

Lemma 3.1 *Computing fire spread probability is #P-complete for single storey structures with a single fire source, all breach probabilities equal, and all spread probabilities unity,* even *if there is only one critical component, or if every volume contains a critical component.* □

One might hope that the problem is easier if all breach probabilities are unity. However, AboElFotoh and Colbourn [1] established that when every *node* has operation probability p, and every *edge* is perfectly reliable, determining the probability that a planar graph G contains an s,t-path for fixed nodes s and t is #P-complete. By a transformation similar to the above, we obtain:

Lemma 3.2 *Computing fire spread probability is #P-complete for single storey structures with a single fire source, a single critical component, all spread probabilities equal, and all breach probabilities unity.* □

These two lemmas establish that any exact algorithm to compute fire spread probabilities apparently requires an amount of time that is exponential in the size of the structure. In fact, they say something much stronger: that even under severe constraints on the type of structure, and selection of probabilities, the problem remains just as difficult (up to a polynomial). While these complexity results are essentially negative, they indicate that in order to analyze fire spread *efficiently* (in polynomial time), we must resort to approximations or bounds.

In view of our assumption that the components are critical in nature, the most useful information is a reasonably tight upper bound; although lower bounds and Monte Carlo estimates may provide valuable information about the fire spread probability, we are most concerned with getting conservative assurance about fire safety with a modest investment of computational resources.

4 An Efficiently Computable Upper Bound

A good first approximation yielding an upper bound for fire spread probability is obtained by examining the first term in the inclusion-exclusion formula. If breach probabilities are near zero as expected, the probability of obtaining the union of two T-fire trees is small compared to the probability of obtaining any one T-fire tree. Hence we expect that if one enumerates all T-fire trees, computes the probability of each, and sums the probabilities ignoring the overcounting, one obtains a useful upper bound. For small buildings (say 25 volumes), this is quite feasible. However, the number of T-fire trees grows exponentially with the number of volumes; in fact, simply counting the T-fire trees is a #P-complete problem (by an easy transformation from counting Steiner trees in an undirected network), and hence is computationally equivalent to computing fire spread probability *exactly*.

One might instead attempt to enumerate the T-fire trees that are the most likely to support fire spread to the critical components, but again we encounter an apparently intractable problem: determining the probability of the *single most likely* T-fire tree is an NP-hard problem, since it includes the problem of finding a minimum Steiner tree in an undirected planar graph, which is NP-hard [7].

If we are to obtain bounds efficiently, these complexity results apparently close off the avenues using T-fire trees. Naturally, we could go on to list many possible approaches that are stopped by the same obstacle of requiring the solution to an NP-hard problem; see, for example, [6]. Instead, we focus on an efficient approach that appears to provide bounds of practical value for the fire spread application.

4.1 Noncrossing Cuts

The s,T-connectedness is precisely the probability that no s,t-cutset is failed for every $t \in T$. Our strategy is to consider a subset of the cutsets, and to compute *exactly* the probability that no cutset in the chosen subset is failed; this is evidently an upper bound on the s, T-connectedness. We denote an s, T-cutset as a partition (A, B) of the vertex set V, having $s \in A$ and $T \cap B \neq \emptyset$.

Two cutsets (A, B) and (A', B') are *noncrossing* if at least one of $A \cap A'$, $A \cap B'$, $B \cap A'$ or $B \cap B'$ is empty; otherwise they are *crossing*. Lomonosov [10] and Shantikumar [15] have used noncrossing cuts to bound reliability in undirected networks when $K = V$ and $|K| = 2$, respectively. Here we use noncrossing cuts for the most general case, s, T-connectedness; our method encompasses that of Shantikumar when there is a single critical component, and improves on that of Lomonosov when every volume houses a critical component. Most importantly, it has no direct dependence on the number of terminals, or critical components.

First, we find a maximal set of noncrossing cuts for the directed graph $G = (V, A)$ as follows. Assign each arc a capacity $-\ln(1 - p_a)$. Find a minimum capacity cut (A, B) with $s \in A$, $B \cap T \neq \emptyset$. Place this cut in the collection. Now form two directed graphs G_A and G_B as follows. To obtain G_A, identify all nodes of A with s; to obtain G_B, identify all nodes of B with any one of the terminals t in B. We call a cut of G_A (G_B) *nontrivial* if it has not already been selected at a previous iteration in forming the current graph. Every cut of G is either a nontrivial cut of G_A, a nontrivial cut of G_B, a cut already selected such as the cut (A, B), or a cut that crosses (A, B). Then recursively find a nontrivial cut of G_A and of G_B, each of minimum capacity; repeat the process as long as the reduced graphs have nontrivial cuts. This greedy method yields a set of $|V| - 1 + |T| - 1$ pairwise noncrossing cuts (this is easily proved by induction on $|V|$ and $|T|$). Since minimum capacity cutsets are chosen at each stage, typically very failure prone cuts are chosen. It is not as obvious how to prevent the algorithm from selecting the same cutset twice. We employ a simple heuristic. When a cutset is found, and new graphs G_A and G_B are formed, one of the arcs of the cutset is contracted in both G_A and G_B, making it impossible to find the cutset at a later stage. The contraction of this *sacrifice edge* results in the absorption of a node. If the node so absorbed is an active terminal, we obtain an additional cutset by taking all arcs entering the active terminal being absorbed, as one of the noncrossing cuts. This absorption occurs exactly $|T| - 1$ times during the course of the algorithm.

Now consider a set $\mathcal{C}$ of ℓ pairwise noncrossing cuts $(A_1, B_1), \dots, (A_\ell, B_\ell)$. Form a directed graph H on $\ell + 1$ vertices $c_0, \dots, c_\ell$; associate with c_0 the "cutset" partition $(A_0, B_0) = (\emptyset, V)$. Place an arc labelled (A_j, B_j) from c_i to c_j whenever $B_j \subset B_i$, and there is no cut (A_k, B_k) with $B_j \subset B_k \subset B_i$. It is easy to verify that H is a directed tree, whose unique source is the cutset $(\emptyset, V)$. Each sink (leaf) of H represents a cutset $(V \setminus \{t\}, \{t\})$ for some $t \in T$. Of significance here is the fact that each arc of the directed tree is associated with one of ℓ the noncrossing cuts of G. If all arcs in one of these cutsets fail, the root of the tree disconnects from at least one of the leaves.

An arc a of G may lie in many of the cutsets of $\mathcal{C}$. For each arc a, let $H(a)$ be the subgraph of H induced by the arcs of H whose corresponding cutsets contain a. We claim that $H(a)$ is a directed path in H. To see this, consider $a = (x, y)$ and find the smallest B_i containing y. Arc a appears in the cut (A_j, B_j) whenever $y \in B_j$ and $x \in A_j$. Now every cut with $y \in B_j$ is a predecessor of (A_i, B_i) in the directed tree H, and hence $H(a)$ is a subgraph of the directed path from the source of H to (A_i, B_i). Now on this path find the smallest A_k containing x. On the directed path from (A_k, B_k) to (A_i, B_i), every cut contains x on the same shore as s, and contains y on the opposite shore.

Hence $H(a)$ is a directed path.

Computing the probability that no cut of $\mathcal{C}$ is failed is therefore equivalent to the following problem. Let $H = (V, E)$ be a directed tree rooted at r having leaves $v_1, \ldots, v_{|T|}$. Let $\mathcal{A}$ be a set of directed subpaths of H; each $a \in \mathcal{A}$ operates independently with probability p_a. What is the probability that the operating subpaths permit connections from r to each leaf of H? (When a subpath $a \in \mathcal{A}$ operates, all nodes on it are reachable from the first node on it.)

H together with the directed paths in $\mathcal{A}$ is a *directed path model.* Our task in obtaining an upper bound is now reduced to computing the probability of reaching all leaves in a directed path model. When H is itself a directed path, this problem can be reduced to computing reliability with imperfect nodes and perfect links on interval graphs [2]; here we generalize to the case when H is arbitrary.

4.2 Directed Path Models

Given a directed path model $(H, \mathcal{A})$, we compute the probability of reaching all leaves by a dynamic programming method. We repeatedly treat one arc at a time, pruning the tree H when all arcs ending at a vertex of H have been treated. In the pruning, we amalgamate terminals when their common subtree has been completely pruned off. An *active terminal* is either one of the leaves of the tree H, or (during the course of the algorithm) a subset of terminals in a common subtree that has been pruned.

In the dynamic program, for every vertex v of H, and for every active terminal t, we maintain the probability $Pr(v, t)$ that *using the directed paths* $a \in \mathcal{A}$ *already processed*, the closest node to the root that can reach the active terminal t, *not using* paths to other active terminals, is v. Recall that t may represent many original leaves; if so, the probability maintained is that *all* leaves represented by the active terminal can be reached.

Initially, for every leaf v_i, we set $Pr(v_i, v_i) = 1$; we set $Pr(v, v_i) = 0$ for all vertices v and all leaves v_i provided $v \neq v_i$. We associate with every vertex v of H a set T_v of active terminals for v; initially this is the set of all leaves in the subtree rooted at v in H.

In updating these measures, our strategy is to choose a leaf of the current (pruned) tree H, merge all active terminals at the leaf to form a single active terminal, process all directed paths in $\mathcal{A}$ that terminate at the leaf, and finally to delete the leaf.

If there is a leaf v with two (or more) active terminals t, t', we merge the two terminals t and t' as follows. Let $r = v_0, v_1, ..., v_{\ell-1}, v_\ell = v$ be the unique root to leaf path. Let $p_i = Pr(v_i, t)$ and $p'_i = Pr(v_i, t')$. Observe that if v_i is the vertex closest to the root that can reach the active terminal t, v_i is the first vertex that can reach both t and t' provided no vertex v_j, $j < i$, can reach t' and that some vertex v_k, $i \leq k \leq \ell$, can reach active terminal t'. Therefore we replace t and t' by a single active terminal u with $Pr(v_i, u) = p_i p'_i + p_i \sum_{x=i+1}^{\ell} p'_x + p'_i \sum_{x=i+1}^{\ell} p_x$.

Once every leaf has a unique active terminal, we examine all directed paths terminating at a specified leaf v having active terminal t. Consider a directed path $v_i, ..., v_\ell = v$ having probability p_a. If the directed path operates, v_i becomes the earliest vertex that can reach active terminal t *provided* some vertex on the directed path can already reach t. On the other hand, if the directed path fails, the earliest vertex that can reach t remains unchanged. Hence we update measures by setting $Pr(v_i, t) = p_a \sum_{x=i}^{\ell} Pr(v_x, t) + (1 - p_a) Pr(v_i, t)$, and for $i + 1 \leq x \leq \ell$, $Pr(v_x, t) = (1 - p_a) Pr(v_x, t)$. Once measures are updated in this way, remove the directed path.

Once all directed paths ending at a specified leaf are processed and removed, delete the leaf from H.

When H is reduced to a single vertex, there remains only one active terminal; the probability that this active terminal can be reached is precisely the probability that all leaves of the original directed tree can be reached.

4.3 Correctness and Timing

The dynamic program maintains $O(n^2)$ measures ($n = |V|$). It performs $|T| - 1$ amalgamations, each requiring at most $O(n^2)$ time, and processes $|\mathcal{A}|$ directed paths, each requiring $O(n)$ time. In computing the probability that none of a maximal set of cuts is failed, we have $|\mathcal{A}| = |A| = O(n^2)$, and hence the total time required in the dynamic program is $O(n^3)$. It is important to note that while $O(n^2)$ paths are processed, and $O(n^2)$ measures are maintained, only $O(n)$ measures are updated for each path.

Correctness of the algorithm is easily verified by checking that the rules for merging active terminals and processing directed paths update the probabilities $Pr(v,t)$ correctly.

The application of the upper bound developed here in the fire spread problem is immediate. Most importantly, the technique developed supports arbitrary probabilities, and exploits the presence of failure-prone cuts. It also treats the multiterminal aspect of the problem, improving on the (obvious) upper bound obtained by bounding the probability of reaching a single terminal.

5 An Example

In the following figures, we illustrate the application of the cutset selection algorithm and the reduction of the cut tree directed path model using the dynmaic programming approach of section 4.2. Consider the simple graph of Figure 5.1. Node 1 is selected

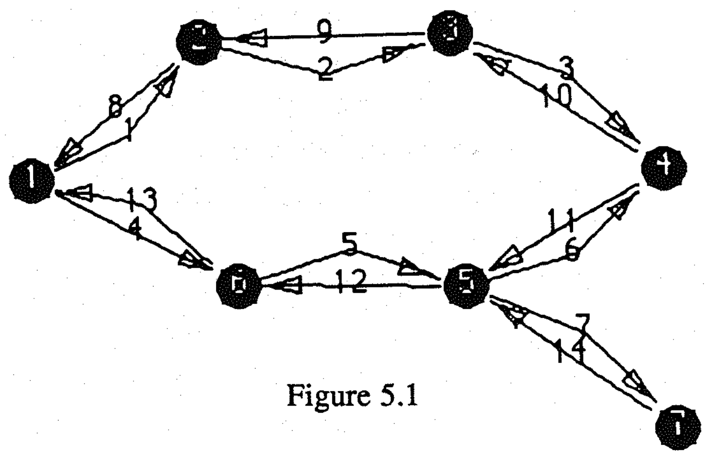

Figure 5.1

as a source, and nodes 4 and 7 as targets; edge operation probabilities are as shown in Figure 5.2. The resulting graph is an instance of the k-terminal reliability problem; we are to compute the probability that the source node S can reach both target nodes marked T, when edges operate? randomly with the stated probabilities. The

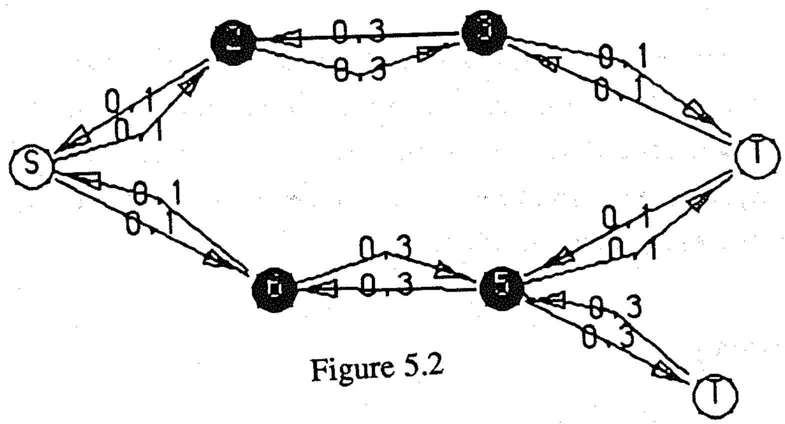

Figure 5.2

computation tree of Figure 5.3 shows the generation of consecutive noncrossing cuts. A minimum capacity cut is found; then the shore of the cut containing the source is collapsed to form a new graph, and the opposite shore is collapsed to form a second new graph. In each, cuts are then found recursively. Each node of the computation tree indicates the cutset edges selected by darkened edges in the graph icon. Left branches of the tree occur when the shore containing the shore is collapsed, and right branches when the opposite shore is collapsed.

In order to prevent the same edge from being selected a number of times, one of the arcs of a selected cutset is contracted — this ensures that the cut cannot be selected again. The edge labels show the cutset edge that is "sacrificed" in this way. In the event that the contraction of the sacrifice edge results in the absorption of a target node (or a collapsed node containing a target node), the set of arcs entering the node is taken as one of the cuts. This occurs $|T| - 1$ times during the course of the computation.

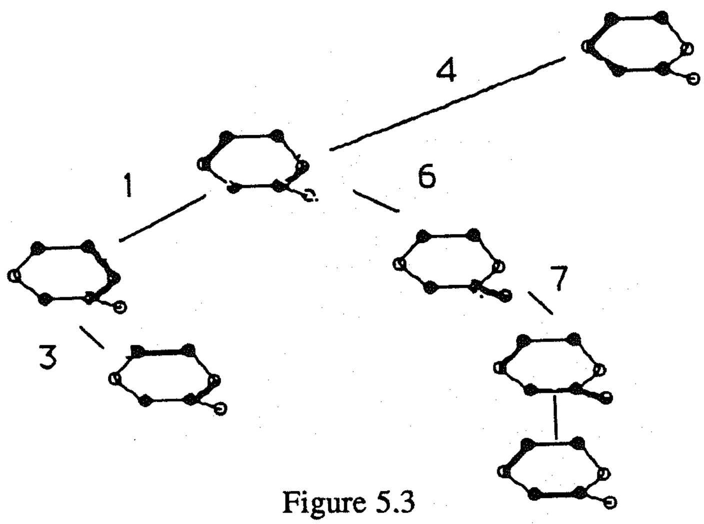

Figure 5.3

Figure 5.4 illustrates the resulting cut tree, and the directed paths of the directed path model. The tree shows that consecutive cutsets arranged as a cut tree, as described in section 4.1. (To be consistent with section 4.1, the cutset is actually associated with the *arc* of the tree terminating at the node containing the cutset.) Figure 5.4 also shows the directed paths resulting when a particular cutset arc operates. The directed paths are labelled d_i, where i is the arc label from the original graph in Figure 5.1. An upper bound on the k-terminal reliability is the probability that the root of the tree can reach all of its leaves via the *directed paths*, with each path operating with the specified probability.

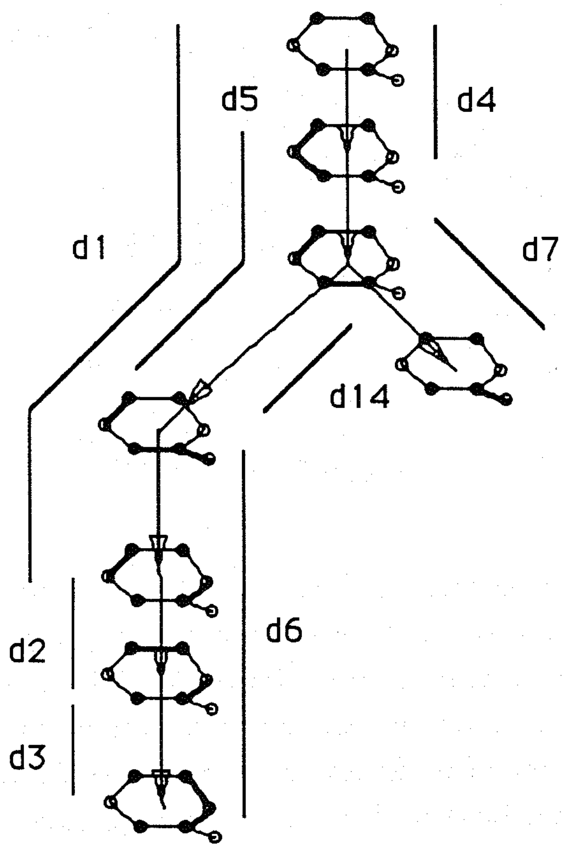

Figure 5.4

In Figure 5 (a)-(h), the dynamic programming reduction of the directed path model is shown. At each stage, all active terminals at a leaf are amalgamated, all directed paths terminating at the leaf are processed, and then the leaf is deleted. The process terminates when the tree consists of a single node. In the figures, the label of a node in the computation tree is the probability that the node is the closest node to the root that can reach the targets in its subtree. When two probabilities are stated, they give the probability of reaching the two targets separately, with the first being for the leftmost target in Figure 4, and the second for the rightmost target.

Figure 5(a) shows the initialization of this process. Target nodes commence with a probability of 1, and all other nodes with probabilities of 0. Figure 5(b) shows the effect of processing directed path d_7 with operation probability .3. The node of

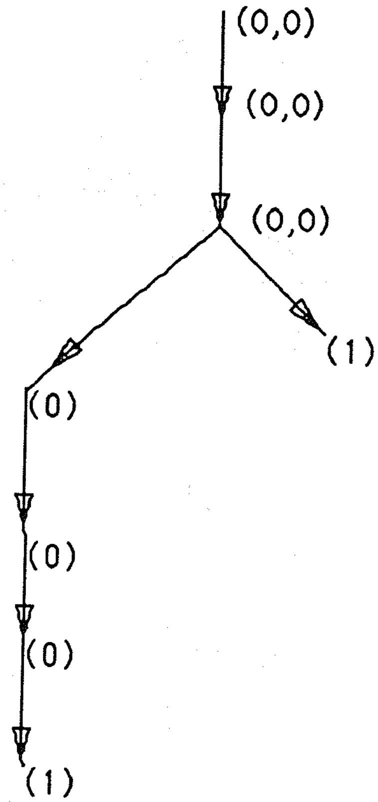

Figure 5a

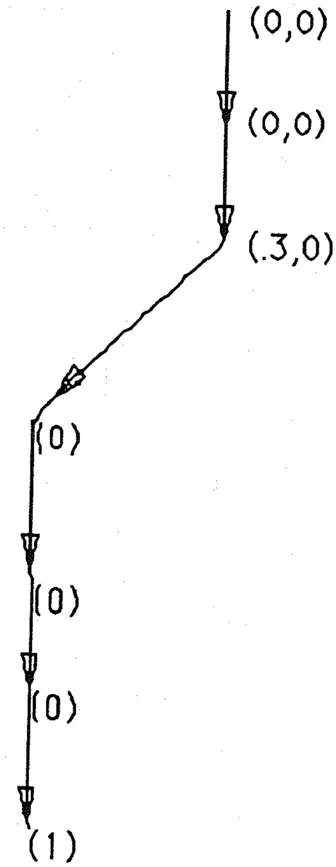

Figure 5b

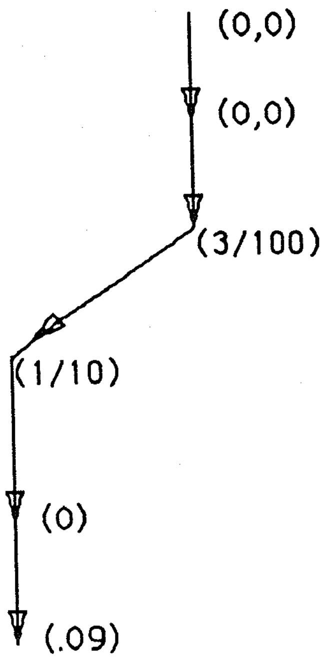

Figure 5c

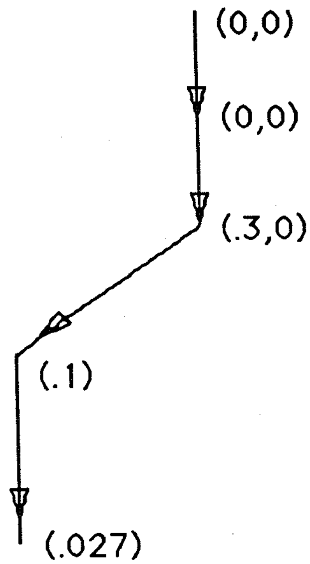

Figure 5d

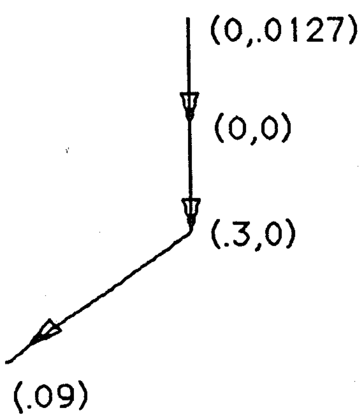

Figure 5e

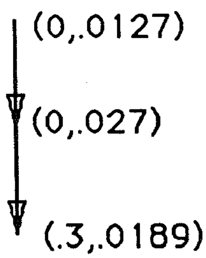

Figure 5f

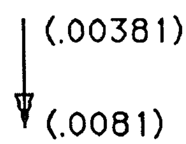

Figure 5g

(.00462)

Figure 5h

d_7 closest to the root now has probability .3 of reaching this target. Continuing the reductions as shown in Figure 5 (c)-(h) yields a result of .00462.

The actual k-terminal reliability is .0010 (to four places) for this simple example.

6 Concluding Remarks

The fire spread problem is a natural analogue of the connectedness problem in communications networks, and hence it comes as no surprise that methods applicable to one are also applicable to the other. However, fire spread assessment concerns a more detailed model than is often used for communications networks, and the range of link probabilities is less constrained. Nevertheless, using ideas from bounds obtained from noncrossing cuts, and a dynamic programming strategy, we have given an easily implementable $O(n^3)$ algorithm for computing an upper bound on the fire spread probability of a structure with n volumes.

We remark that the bound developed here, when applied to an undirected network in which every vertex is a terminal, improves on Lomonosov's upper bound [10], which is competitive with the best available efficiently computable bounds on all-terminal reliability [6].

In closing, it is important to reiterate that using the correspondence developed here with network reliability models, one accounts for fire spread in the absence of effective countermeasures. A truer model of fire spread must account for the agents attempting to control and extinguish the fire.

Acknowledgements

Research of Colbourn and Nel is supported by NSERC Canada. We also wish to express our gratitude to members of the Safety and Reliability Directorate of the UK AEA for many helpful discussions.

References

[1] H. M. F. AboElFotoh and C. J. Colbourn. Computing 2–terminal reliability for radio broadcast networks. *IEEE Trans. Reliability*, R–38:538–555, 1989.

[2] H. M. F. AboElFotoh and C. J. Colbourn. Efficient algorithms for computing the reliability of permutation and interval graphs. *Networks*, to appear.

[3] T. B. Boffey. *Graph Theory in Operations Research.* Macmillan, 1982.

[4] B. M. Cohn. Formulating acceptable levels of fire risk. In *Fire Risk Assessment*, pages 28–37. Amer. Soc. Testing and Materials, 1982.

[5] B. M. Cohn. A fire loss modeling code for industrial facilities. In *Mathematical Modeling of Fires*, pages 58–67. Amer. Soc. Testing and Materials, 1988.

[6] C. J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, 1987.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–Completeness.* Freeman, 1979.

[8] D. Gross. Aspects of stochastic modeling for firesafety. *Fire Technology*, 19:103–114, 1982.

[9] W. C. T. Ling and R. B. Williamson. Using fire tests for quantitative risk analysis. In *Fire Risk Assessment*, pages 38–58. Amer. Soc. Testing and Materials, 1982.

[10] M. V. Lomonosov. Bernoulli scheme with closure. *Problems of Information Transmission*, 10:73–81, 1974.

[11] H. Nakazawa. Equivalence of a nonoriented line and a pair of oriented lines in a network. *IEEE Trans. Reliability*, R28:364–367, 1979.

[12] J. S. Provan. The complexity of reliability computations on planar and acyclic graphs. *SIAM J. Computing*, 15:694–702, 1986.

[13] G. Ramachandran. Stochastic modeling of fire growth. In *Fire Safety: Science and Engineering*, pages 122–144. Amer. Soc. Testing and Materials, 1985.

[14] G. Ramachandran. Probabilistic approach to fire risk evaluation. *Fire Technology*, 24:204–226, 1988.

[15] J. G. Shantikumar. Bounding network reliability using consecutive minimal cutsets. *IEEE Trans. Reliability*, R–37:45–49, 1988.

[16] H. Takeda. Transient model of early stages in compartment fires. In *Mathematical Modeling of Fires*, pages 21–34. Amer. Soc. Testing and Materials, 1988.

[17] D. Vertigan. The computational complexity of Tutte invariants for planar graphs. to appear, 1990.

School of Computer Science, Carleton University
Recent Technical Reports

SCS-TR-149 **Smallscript: A User Programmable Framework Based on Smalltalk and Postscript**
Kevin Haaland and Dave Thomas, November 1988

SCS-TR-150 **A General Design Methodology for Dictionary Machines**
Frank Dehne and Nicola Santoro, February 1989

SCS-TR-151 **On Doubly Linked List ReOrganizing Heuristics**
D.T.H. Ng and B. John Oommen, February 1989

SCS-TR-152 **Implementing Data Structures on a Hypercube Multiprocessor, and Applications in Parallel Computational Geometry**
Frank Dehne and Andrew Rau-Chaplin, March 1989

SCS-TR-153 **The Use of Chi-Squared Statistics in Determining Dependence Trees**
R.S. Valiveti and B.J. Oommen, March 1989

SCS-TR-154 **Ideal List Organization for Stationary Environments**
B. John Oommen and David T.H. Ng, March 1989

SCS-TR-155 **Hot-Spot Contention in Binary Hypercube Networks**
Sivarama P. Dandamudi and Derek L. Eager, April 89

SCS-TR-156 **Some Issues in Hierarchical Interconnection Network Design**
Sivarama P. Dandamudi and Derek L. Eager, April 1989

SCS-TR-157 **Discretized Pursuit Linear Reward-Inaction Automata**
B.J. Oommen and Joseph K. Lanctot, April 1989

SCS-TR-158 (revised) **Parallel Fractional Cascading on a Hypercube Multiprocessor**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, May 1989 (Revised April 1990)

SCS-TR-159 **Epsilon-Optimal Stubborn Learning Mechanisms**
J.P.R. Christensen and B.J. Oommen, June 1989

SCS-TR-160 **Disassembling Two-Dimensional Composite Parts Via Translations**
Doron Nussbaum and Jörg-R. Sack, June 1989

SCS-TR-161 (revised) **Recognizing Sources of Random Strings**
R.S. Valiveti and B.J. Oommen, January 1990
Revised version of SCS-TR-161 "On the Data Analysis of Random Permutations and its Application to Source Recognition", published June 1989

SCS-TR-162 **An Adaptive Learning Solution to the Keyboard Optimization Problem**
B.J. Oommen, R.S. Valiveti and J. Zgierski, October 1989

SCS-TR-163 **Finding a Central Link Segment of a Simple Polygon in O(N Log N) Time**
L.G. Alexandrov, H.N. Djidjev, J.-R. Sack, October 1989

SCS-TR-164 **A Survey of Algorithms for Handling Permutation Groups**
M.D. Atkinson, January 1990

SCS-TR-165 **Key Exchange Using Chebychev Polynomials**
M.D. Atkinson and Vincenzo Acciaro, January 1990

SCS-TR-166 **Efficient Concurrency Control Protocols for B-tree Indexes**
Ekow J. Otoo, January 1990

SCS-TR-167 **A Hierarchical Stochastic Automaton Solution to the Object Partitioning Problem**
B.J. Oommen, January 1990

SCS-TR-168 **Adaptive List Organizing for Non-stationary Query Distributions. Part I: The Move-to-Front Rule**
R.S. Valiveti and B.J. Oommen, January 1990

SCS-TR-169 **Trade-Offs in Non-Reversing Diameter**
Hans L. Bodlaender, Gerard Tel and Nicola Santoro, February 1990

SCS-TR-170 **A Massively Parallel Knowledge-Base Server using a Hypercube Multiprocessor**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990

SCS-TR-171 **Parallel Processing of Quad Trees on the Hypercube (and PRAM)**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990

SCS-TR-172 **A Note on the Load Balancing Problem for Coarse Grained Hypercube Dictionary Machines**
Frank Dehne and Michel Gastaldo, May 1990

SCS-TR-173 **Self-Organizing Doubly-Linked Lists**
R.S. Valiveti and B.J. Oommen, May 1990

SCS-TR-174 **A Presortedness Metric for Ensembles of Data Sequences**
R.S. Valiveti and B.J. Oommen, May 1990

SCS-TR-175 **Separation of Graphs of Bounded Genus**
Ljudmil G. Aleksandrov and Hristo N. Djidjev, May 1990

SCS-TR-176 **Edge Separators of Planar and Outerplanar Graphs with Applications**
Krzystof Diks, Hristo N. Djidjev, Ondrej Sykora and Imrich Vrto, May 1990

SCS-TR-177 **Representing Partial Orders by Polygons and Circles in the Plane**
Jeffrey B. Sidney and Stuart J. Sidney, July 1990

SCS-TR-178 **Determining Stochastic Dependence for Normally Distributed Vectors Using the Chi-squared Metric**
R.S. Valiveti and B.J. Oommen, July 1990

SCS-TR-179 **Parallel Algorithms for Determining K-width-Connectivity in Binary Images**
Frank Dehne and Susanne E. Hambrusch, September 1990

SCS-TR-180 **A Workbench for Computational Geometry (WOCG)**
P. Epstein, A. Knight, J. May, T. Nguyen, and J.-R. Sack, September 1990

SCS-TR-181 **Adaptive Linear List Reorganization under a Generalized Query System**
R.S. Valiveti, B.J. Oommen and J.R. Zgierski, October 1990

SCS-TR-182 **Breaking Substitution Cyphers using Stochastic Automata**
B.J. Oommen and J.R. Zgierski, October 1990

SCS-TR-183 **A New Algorithm for Testing the Regularity of a Permutation Group**
V. Acciaro and M.D. Atkinson, November 1990

SCS-TR-184 **Generating Binary Trees at Random**
M.D. Atkinson and J.-R. Sack, December 1990

SCS-TR-185 **Uniform Generation of Combinatorial Objects in Parallel**
M.D. Atkinson and J.-R. Sack, January 1991

SCS-TR-186 **Reduced Constants for Simple Cycle Graph Separation**
Hristo N. Djidjev and Shankar M. Venkatesan, February 1991

SCS-TR-187 **Multisearch Techniques for Implementing Data Structures on a Mesh-Connected Computer**
Mikhail J. Atallah, Frank Dehne, Russ Miller, Andrew Rau-Chaplin, and Jyh-Jong Tsay, February 1991

SCS-TR-188 **Generating and Sorting Jordan Sequences**
Alan Knight and Jörg-Rüdiger Sack, March 1991

SCS-TR-189 **Probabilistic Estimation of Damage from Fire Spread**
Charles C. Colbourn, Louis D. Nel, T.B. Boffey and D.F. Yates