Threshold Reliability of Networks with Small Failure Sets

Michael O. Ball

University of Maryland, College Park, MD

and

University of North Carolina, Chapel Hill, NC

Jane N. Hagstrom

University of Illinois, Chicago, IL

J. Scott Provan

University of North Carolina, Chapel Hill, NC

March 10, 1993

Abstract

This paper addresses two classes of reliability analysis models: a network flow model and a project scheduling model. For the network flow model we are given a capacitated source/sink graph in which arcs fail randomly. The system is defined as operating whenever the max-flow value is greater than a threshold. For the project scheduling model we are given a directed acyclic source/sink graph in which each arc has two lengths. Each arc randomly takes on one of two states. In the "operating" state it takes on its smaller length and in its "failed" state it takes on it larger length. The system is defined as operating whenever the length of the longest path is less than a threshold. We address a special case of the threshold flow problem in which all arcs have the same capacity and a special case of the project scheduling model in which the difference between the lower and higher arc lengths is constant. For these special cases we show that if the underlying systems are 1-critical, namely, all arcs are in some cutset of size two, then the threshold flow problem can be solved in polynomial time and planar project scheduling problem can be solved in polynomial time. Both solutions are obtained by reducing the problems to the problem of determining the probability that the failed arcs in a directed acyclic graph lie on a single path. We also show how the basic approach can be used to generate bounds for systems that are "almost critical".

1 Introduction

This paper examines *threshold reliability* problems associated with two popular stochastic network models: the network flow model and a project scheduling model. These problems take as input standard flow or PERT networks, with the capacities or task times being stochastic, together with a *threshold* parameter which specifies an acceptable level of system performance. The underlying system is said to *operate* if the random system performance level is within the specified acceptable level of operation. Specifically, for the flow problem, the system operates if the value of the maximum flow is greater than or equal to the threshold flow parameter, and for the project scheduling model, the system operates if the project completion time is less than or equal to the threshold completion time parameter. This paper concerns the problem of computing the probability of system operation in threshold flow and PERT systems. The threshold measures studied in this paper are simple examples of *performability* measures. Performability measures have received significant attention recently because of the recognition that, in many situations, for a system to be useful it must not only operate, but it must operate with an adequate performance level.

One novel aspect of our analysis relates to the classes of problem instances we have chosen to examine. The philosophy we espouse is that, when searching for special structure that renders a problem tractable, one should first consider classes of problems that arise in practice. Problems that arise in practice are likely to be those that would be output by (automatic or manual) system design procedures. The most typical reliable system design criterion is that the system be survivable, that is, that it be able to withstand the failure of any single component. A typical system design process would attempt to minimize cost subject to the constraint that the system be survivable. Systems output by such a process would be minimally survivable in the sense that the deletion of any single component would result in a system that was not survivable. In this paper we address reliability analysis problems for systems that are minimally survivable or close to minimally survivable. It turns out that the structure these classes of problems can be employed to give efficient algorithms for threshold reliability computations.

The paper is organized as follows. The remainder of the introduction gives basic definitions and, in the final subsection, states the main results. Sections 2 and 3 give structural and complexity results for the flow and PERT problems respectively. In particular, they relate special classes of the threshold flow and planar PERT reliability problems to *stochastic path problems* on acyclic graphs. These involve computing the probability that the failed arcs are contained in a collection of (s,t)-paths (arc-disjoint paths, in the case of flows) of cardinality equal to the differential between the threshold and optimal system performance. In Section 4 we present Algorithm 1-PATH, which computes this probability for the case where the differential is one. This algorithm solves the threshold flow problem and planar PERT problem for systems that are minimally survivable. The final section uses 1-PATH to provide bounds on threshold reliability for systems that are "almost" minimally survivable.

1.1 Threshold Flow Systems

Capacitated flow networks model transportation systems, communication networks, and power grids. In these contexts, a common reliability criterion is the ability to carry sufficient flow when failures may cause certain links in the network to be unavailable or only partially available. In this paper, we consider a two-terminal reliability problem, in which the criterion concerns the ability to carry sufficient flow between a specific origin node and a specific destination node.

The Model. A *flow graph* is defined by $G = (N, A, \boldsymbol{c}, \boldsymbol{\ell}, s, t)$ where N is a set of nodes, A is a set of directed or undirected arcs, $\boldsymbol{c}$ and $\boldsymbol{\ell}$ are non-negative integer capacity vectors defined on the arcs with $\boldsymbol{c} \geq \boldsymbol{\ell}$; and s and t are two specified terminal nodes. The inputs to the threshold flow reliability analysis problem consist of a flow graph G, a probability vector $\boldsymbol{p}$ defined on the arcs and a flow threshold, f. Each arc e operates and has capacity c_e with probability p_e and fails and has capacity ℓ_e with probability $1 - p_e$. The state of a given arc is independent of the state of any other arc. The *threshold flow reliability analysis problem* is to compute

$RF(G, \boldsymbol{p}, f) = \Pr\{G \text{ admits a flow of at least } f\}$.

Let us now examine some special cases of this problem. A very natural model with many applications arises when we assume that $\ell_e = 0$ for all e. This assumption implies that a failure renders an arc completely unusable. The results of this paper apply to this model with the further restriction that $c_e = 1$ for all e. We note that this second restriction is equivalent to the not unrealistic case of $c_e = \hat{c}$ for all e, since we could set $c_e = 1$ for all e and divide f by $\hat{c}$ to obtain an equivalent problem with all capacities equal to one. The case of $c_e = 1$ and $\ell_e = 0$ is a special case of a more general *unit failure decrement* model in which $c_e - \ell_e = 1$ for all e. Most of our results apply in this slightly more general setting. One of our motivations for considering it is the parallels that arise with the restricted project scheduling model we consider. Since we will assume the unit failure decrement model throughout the paper, we will specify a flow graph by $G = (N, A, \mathbf{c}, s, t)$ where it is understood that $\ell_e = c_e - 1$ for all e.

1.2 Project Scheduling with a Completion Time Threshold

Project networks are often used to represent the precedence constraints between tasks of a project. We will consider the activity-on-arc network, in which each task is represented by an arc and required precedence of one task before another is represented by directing the earlier task's arc into a node and directing the later task's arc out of the same node. Using the activity-on-arc representation sometimes requires the addition of so-called *dummy arcs* in order to correctly represent all precedence requirements. An alternative to this representation is the activity-on-node graph. The latter graph has the advantage that no dummy activities need to be created. We will continue with the activity-on-arc representation because it maintains easier parallels with the flow graphs of the previous section. Throughout our analysis, we will assume that the activity-on-arc representation requires no dummy tasks. At the end of Section 3, we explain how dummy tasks may be accounted for in the analysis.

If the project has a deadline d for completion, and task durations are random variables, we can consider the project schedule to work if the project can be completed in time d; otherwise we will say it has failed. The reliability of the project schedule will be the probability that the deadline can be met.

The Model. An *activity precedence graph* is defined by $G = (N, A, \boldsymbol{a}, \boldsymbol{b}, s, t)$ where N is a set of nodes, A is a set of directed arcs, containing no directed cycles; $\boldsymbol{a}$ and $\boldsymbol{b}$ are non-negative integer length vectors defined on the arcs with $\boldsymbol{a} \leq \boldsymbol{b}$; and s and t are two specified terminal nodes. The graph should be such that s is the only node with indegree 0 and t is the only node with outdegree 0. The inputs to the project scheduling reliability analysis problem are an activity precedence graph, G, a probability vector, $\boldsymbol{p}$, defined on the arcs and a project duration threshold d. The arc e "operates" and has duration a_e with probability p_e, and "fails" and has duration b_e with probability $1 - p_e$. The state of any given arc is independent of the state of any other arc. The network is considered to be working if the longest path from s to t has length no more than d; otherwise the network is failed. The *threshold project scheduling reliability analysis problem* is to compute

$RP(G, \boldsymbol{p}, d) = \Pr\{$the length of the longest path from s to t is no more than $d\}$.

In our analysis we restrict our attention to the *unit failure increment* model in which $b_e - a_e = 1$ for all e. We note that this is equivalent to the case of $b_e - a_e = \hat{d}$ for all e, since we could set $b_e = 1 + a_e$ for all e and set $d = d' + \lceil (d - d')/\hat{d} \rceil$ where d' is the project completion time when all durations are equal to a_e ($\lceil r \rceil$ = the smallest integer greater than or equal to

r). Note that the resultant problem is equivalent to the original problem and has $b_e - a_e = 1$ for all e. In a manner similar to the flow model, throughout the paper we will specify an activity precedence graph by $G = (N, A, \boldsymbol{a}, s, t)$ where it is assumed that $b_e = a_e + 1$ for all e.

1.3 Coherent Binary Systems, k-Survivable Systems and k-Critical Systems

A *stochastic binary system* (SBS) represents a system that fails randomly as a function of the random failure of its components. Each component, e, in the system component set, T, operates with probability p_e and fails with probability $1 - p_e$. The structure of the system is represented by a function $\phi(S)$ defined for each $S \subseteq T$ by

$$\phi(S) = \begin{cases} 1 \text{ if when } S \text{ operates and } T - S \text{ fails, the system operates} \\ 0 \text{ if when } S \text{ operates and } T - S \text{ fails, the system fails.} \end{cases}$$

An SBS is *coherent* if $\phi(T) = 1$, $\phi(0) = 0$ and $\phi(S') \geq \phi(S)$ for any $S' \supset S$. The third property implies that the failure of any component can only have a detrimental effect on the operation of the system. The reliability analysis problem is to compute:

$$\Pr\{\ \phi(S) = 1 \text{ where } S \text{ is the set of operative components}\}$$

given some representation of $\phi()$. For any stochastic coherent binary system (SCBS), define a *pathset* as a set of components whose operation implies system operation, and a *minpath* as a minimal pathset; similarly, define a *cutset* to be a set of components whose failure implies system failure, and a *mincut* to be a minimal cutset. An SCBS is *k-survivable* if it continues to operate after the failure of any set of elements of size k or smaller, i.e. if there are no cutsets of size k or less; this definition applies for any k greater than or equal to 0. An SCBS *k-critical* if it is k-survivable and if each element is in some mincut of size k+1. 1-survivable and 1-critical systems are also referred to simply as *survivable* and *critical* systems, respectively.

In this paper we concentrate our efforts on the solution of reliability analysis problems on k-critical systems or systems that are almost k-critical. We treat most extensively critical or near-critical systems. It is our contention that when one considers design philosophies and design criteria that it becomes clear that critical or near-critical systems are found in the majority of cases encountered in practice.

1.4 Summary of Results

The acyclic graph reliability analysis problems that result from the threshold flow and project scheduling problems all involve definitions of system operation where failed arcs must be restricted to certain path subsets. Given a directed acyclic graph, we define a *k-path* as a subset of arcs that can be formed by taking the union of k paths and a *k-dpath* as a set of arcs that can be formed by taking the union of k disjoint paths. A *k-path subset* and *k-dpath subset* are sets of arcs that are the subset of some k-path or k-dpath, respectively.

Given an acyclic graph $G = (N, A)$, a non-negative integer, k and a probability vector $\boldsymbol{p}$ defined on the arcs of G so that each arc e operates with probability p_e and fails with probability $1 - p_e$, we define

$$\Psi(G, \boldsymbol{p}, k) = \Pr\{\text{the set of failed arcs form a } k\text{−path subset}\}$$
$$\Psi^d(G, \boldsymbol{p}, k) = \Pr\{\text{the set of failed arcs form a } k\text{−dpath subset}\}$$

Note that k-paths and k-dpaths define the same structures for $k = 1$ and so we define

$$\Psi(G, \boldsymbol{p}) = \Psi(G, \boldsymbol{p}, 1) = \Psi^d(G, \boldsymbol{p}, 1)$$

Section 2 shows that computing threshold flow reliability on a k−critical directed flow graph is equivalent to computing $\Psi^d(G, \boldsymbol{p}, k)$ on the graph when $c_e = 1$ for all e. When $k = 1$, this result also holds for unit failure decrement case. Section 3 shows that the project scheduling reliability analysis problem can be solved on a k-critical planar activity precedence graph by computing $\Psi(G, \boldsymbol{p}, k)$ on the dual graph. In Section 4 we present Algorithm 1-PATH, which computes $\Psi(G, \boldsymbol{p})$ in $O(n^2)$ time. Thus, it forms the basis for the two main results of the paper,

Theorem 1.1 *The threshold flow reliability analysis problem can be solved in $O(n^2)$ time for directed 1-critical systems with unit failure decrements.*

Theorem 1.2 *The threshold project scheduling reliability analysis problem can be solved in $O(n^2)$ time for planar 1-critical systems with unit failure increments.*

Section 4 also presents other results for critical systems. In particular, it shows how 1-PATH can be used to compute threshold flow reliability for undirected 1-critical graphs.

Section 5 embeds 1-PATH in a procedure for computing bounds on "almost-critical" threshold flow systems.

Both the threshold flow reliability analysis problem and the project scheduling reliability analysis problem are NP-hard, with the project scheduling problem being NP-hard even for 1-critical systems. Basic complexity results are given in Sections 2 and 3. In a sequel to this paper we will present a more detailed complexity analysis as well as solvable cases for higher values of k.

2 Analysis of Threshold Flow Systems

In this section we derive some basic properties of threshold flow systems. Let $G = (N, A, \boldsymbol{c}, s, t)$ be a flow graph, with the capacities $\boldsymbol{c}$ and unit failure decrements. Denote by f^* the max flow value for G, and let f be the given threshold flow value.

We start by deriving some basic complexity results. Note that when $c_e = 1$ and $f = 1$ we have that $RF(G, \boldsymbol{p}, f) = \Pr\{$there exists an operating path from s to t$\}$, the 2-terminal reliability value. Further, by adding a single arc $e' = (t, t')$ from t to new node t' and setting $c_{e'} = 1$ (see Figure 1), we can transform a 2-terminal reliability analysis problem defined on an arbitrary graph, G, into a threshold flow reliability analysis problem defined on a graph G', with $f = f^*$ so that $RF(G', \boldsymbol{p}, f^*) = p_{e'} RF(G, \boldsymbol{p}, 1)$. Thus, we have,

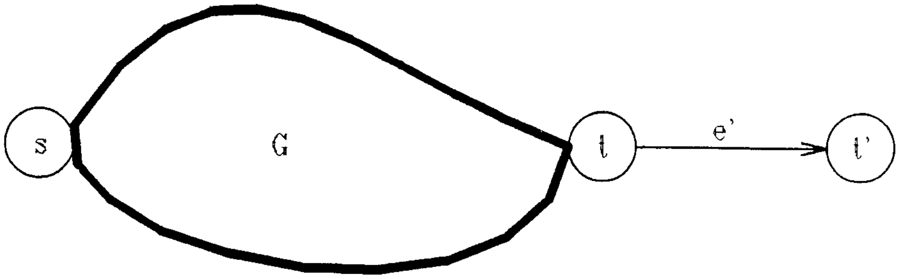

Figure 1: Transformation with Max Flow = 1

Proposition 2.1 *The threshold flow reliability analysis problem is NP-hard even when all of the following hold:*

1. $c_e = 1$ *(and* $l_e = 0$*) for all* e*,*
2. G *is directed, planar, acyclic and has node degree at most 3 or* G *is undirected, planar and has node degree at most 3,*
3. $f = f^*$ *or* $f = 1$.

Proof: The result follows from the preceding discussion and the fact that the 2-terminal reliability analysis problem is NP-hard when restricted to the class of graphs given in (2.) [8]. ▮

We remark that a more detailed analysis which would use techniques similar to those given in [5] could be used to show that problems equivalent to those mentioned in the proposition are in the class *#NP-complete.* #P-complete is an equivalence class of *counting* problems that contains problems such as counting Hamiltonian circuits, 3-matchings, or most other solutions to NP-complete problems (see [11]).

We next investigate properties of the coherent binary system underlying the threshold flow problem. An *(s,t)-cut* of G is a subset of arcs of the form

$$C = \{e = (u,v) \in A \mid u \in X \text{ and } v \in \bar{X}\}.$$

where X and $\bar{X}$ are subsets of nodes with $s \in X$ and $t \in \bar{X} = N \setminus X$. For directed graphs, the arc orientation is important so that the cut contains only those arcs that are directed from X to $\bar{X}$. The *capacity* of C, *cap*(C), is the sum of the capacities of the arcs of C. Cutsets of threshold flow systems, which we call *f-flow-cuts*, are simply subsets of arcs for which decreasing capacities to their lower value — *i.e.* decreasing the capacity in these arcs by 1 — leaves the value of the max (s,t)-flow strictly less than f. The following proposition relates f-flow-cuts and (s,t)-cuts.

Lemma 2.2 *Let G be a flow graph with unit failure decrements, f^* the maximum flow value for G, and γ^* the minimum number of arcs in an (s,t)-cut for G. Then for any subset D of arcs and any threshold flow value $f > f^* - \gamma^*$:*

(i) D is an arc-minimal f-flow-cut if and only if D is contained in every cut C with cap$(C) = f + |C \cap D| - 1$ and there is at least one such cut.

(ii) D is a minimum cardinality f-flow-cut if and only if $|D| = f^* - f + 1$ and D is contained in some (s,t)-cut C^* with cap$(C^*) = f^*$.

Proof: (i): Denote by c_D the capacity vector obtained by decreasing each of the capacities of arcs in D by one unit, and for any (s,t)-cut C denote by $cap_D(C)$ the capacity of C with respect to c_D. Note that $cap_D(C) = cap(C) - |D \cap C|$. Now D is an f-flow-cut if and only if the max flow using the capacity vector $\boldsymbol{c}_D$ has value at most $f - 1$, and from the Max-flow Min-cut Theorem this occurs if and only if there is an (s, t)-cut C^0 with $cap_D(C^0) = cap(C^0) - |C^0 \cap D| \leq f - 1$. Further, D is arc-minimal if and only if in addition to the above each proper subset D' of D has the property that $cap(C) - |C \cap D'| \geq f$ for every (s, t)-cut C. Part (i) follows from the above two observations.

(ii): First note that every f-flow-cut must have cardinality at least $f^* - f + 1$, and this cardinality can be achieved by taking any min cut C^*— of capacity f^* — and choosing any $f^* - f + 1 \leq \gamma^* \leq |C^*|$ arcs to be the f-flow-cut. Part (ii) now follows directly from part (i). ▮

In Figure 2, let all capacities be 1 and the let threshold flow value be 2. Then $f^* = 4$, and {(1,3),(2,3),(4,5),(4,8)} is a *minimal* 2-flow-cut of size 4 while {(3,5),(4,5),(4,6)} is a *minimum cardinality* 2-flow-cut of size 3.

The algorithm and bounds we give in this paper concern r-critical or near r-critical systems. We say that a flow graph G is *threshold flow r/f-survivable* (*threshold flow r/f-critical*) if it is r-survivable (r-critical) with respect to $RF(G, \boldsymbol{p}, f)$. The flow system in Figure 2 is thus r/f-survivable for $4 - f \geq r$, and r/f-critical for $4 - f = r \leq 3$. In general, we get from Lemma 2.2 that as long as $f > f^* - \gamma^*$ a threshold system is r/f-survivable if and only if $f^* - f \geq r$. It turns out that for the threshold flow problem r/f-critical systems have a particularly simple structure, which depends almost exclusively on the flow graph itself rather than the values of f or r. We call a directed flow graph G a *k-flow graph* if the following conditions hold:

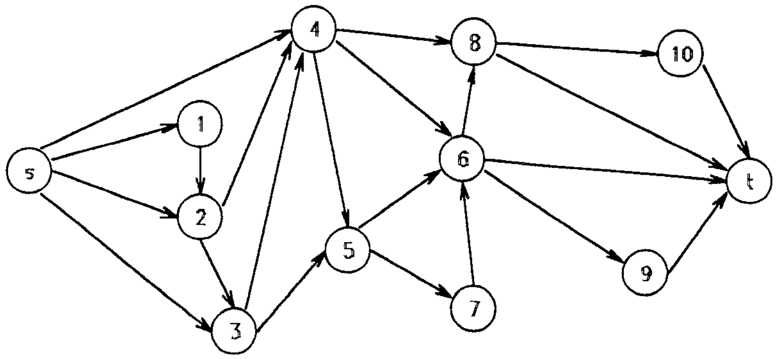

Figure 2: A 4-Flow Graph

1. G is acyclic;

2. s has total capacity k on its outgoing arcs and has no incoming arcs;

3. t has total capacity k on its incoming arcs and has no outgoing arcs;

4. Every node other than s and t has the same total capacity on its sets of outgoing and incoming arcs.

The flow graph described in Figure 2 is in fact a directed 4-flow graph. We say that an undirected flow graph is a k-flow graph if the arcs can be directed in such a way that the resulting directed graph is a directed k-flow graph. Note that one can trivially test whether a directed flow graph is a k-flow graph. To determine whether an undirected graph is a k-flow graph, first compute a max (s, t)-flow. If the flow has value k, is acyclic and saturates all the undirected arcs then the undirected graph is a k-flow graph.

It is clear that a directed or undirected flow graph has a unique acyclic max flow if and only if it is a k-flow graph. What is more, there is a close connection between k-flow graphs and criticality of the associated flow system.

Proposition 2.3 *Let $G = (N, A, \mathbf{c}, s, t)$ be a flow graph with unit failure decrements and f^*, γ^* as in Lemma 2.2. Then the following are equivalent*

(i) G is threshold flow r/f-critical for some r, f with $f^* - f = r < \gamma^*$;

(ii) G is threshold flow r/f-critical for every *r, f with $f^* - f = r < \gamma^*$;*

(iii) G is an f^*-flow graph.

Proof: $(i) \Rightarrow (iii)$: Since G is r-survivable with respect to threshold f and $r < \gamma^*$, then there must be a flow of at least $f + r$ in G. Now in order for G to be r-critical with respect to f, it must be that every arc e of G must be in an f-flow-cut of (minimum) cardinality $r + 1$, which by Lemma 2.2 implies that e must be in an (s, t)-cut of capacity $f + r = f^*$. But since all arcs are now in some minimum capacity cut, then the maximum (s, t)-flow must be unique and saturate all of the arcs of the graph. This immediately implies that G satisfies the conditions of being an f^*-flow graph.
$(iii) \Rightarrow (ii)$: Choose r and f satisfying $f^* - f = r < \gamma^*$. Since G is an $(f + r)$-flow graph, then G has a unique flow of value $f + r$ that saturates every arc. This means (a) that G is r-survivable, and (b) that every arc of G is in a cut of capacity $f + r$. Lemma 2.2 implies that every arc of G is in an f-flow-cut of cardinality r, and (ii) follows.
$(ii) \Rightarrow (i)$: This is immediate.

∎

What Proposition 2.3 implies is that the structure of an r-critical threshold flow system is essentially independent of either r or the threshold flow value f, depending only on the flow graph itself. This means that we can henceforth refer to critical threshold flow systems by simply giving a k-flow graph, with the precise degree of criticality specified when a particular r or f is given.

The property of criticality in the threshold flow reliability analysis problem immediately gives the reliability when $f = f^*$, since in a critical (in this case 0-critical) threshold flow system *all* of the arcs must operate in order that the system operate. Thus

$$RF(G, \boldsymbol{p}, f^*) = \prod_{e \in A} p_e.$$

We prove in a subsequent paper that the problem of computing $RF(G, \boldsymbol{p}, f^* - 2)$ in critical systems is NP-hard. This paper concentrates specifically on the computation of $RF(G, \boldsymbol{p}, f^* - 1)$.

In the remainder of this section we will restrict our attention to directed flow graphs. In Subsection 4.2, we will show that the results for 1-critical systems can be extended to undirected flow graphs.

For the next result we need to make the further assumption that $c_e = 1$ for all arcs. To begin with, this means that in the definition of k-flow subgraph we can replace "capacity" by "cardinality"; that is, k-flow graphs are characterized entirely by the underlying graph structure rather than by particular capacity values. Second, since $f^* = \gamma^*$, then the assumption $f^* - f < \gamma^*$ can simply be replaced by $f > 0$ in all of the results. The following lemma gives properties that are very useful in the development of analysis algorithms.

Lemma 2.4 *Let G be a directed k-flow graph with unit capacities and unit failure decrements. Then*

(*i*) *For any 0-1 (s,t)-flow in G of value f, the set of flow-bearing arcs induces an f*-flow subgraph *of G. Thus the* pathsets *for the f-threshold flow system on G are precisely those collections of arcs* containing *an f-flow subgraph.*

(*ii*) *For any 0-1 (s,t)-flow in G of value f, the set of* non-*flow-bearing arcs induces a $(k-f)$*-flow subgraph *of G. Thus the* non-cutsets *for the f-threshold flow system on G are precisely those collections of arcs* contained in *a $(k-f)$-flow subgraph.*

Proof: (*i*): The first statement follows from the definition of f-flow subgraph and the fact that flows are all 0-1, and the second follows since flows can always be assumed to be 0-1 and the set of flow-bearing arcs must be contained in the set of operating arcs.
(*ii*) The first statement follows since both the flows and the capacities are 0-1, and the second follows since flows can always be assumed to be 0-1 and the set of non-flow-bearing arcs must contain the set of failed arcs. ▮

Lemma 2.4 leads to a simple reliability formula. Let $G = (N, A, \mathbf{c}, s, t)$ be a k-flow graph with unit capacities and unit failure decrements. For any set S of failed arcs and $\bar{S} = A \setminus S$, we have

$$\begin{aligned} RF(G, \boldsymbol{p}, f) &= \Pr\{(N, \bar{S}) \text{ contains an } f\text{-flow subgraph}\} && (1) \\ &= \Pr\{(N, S) \text{ is contained in a } (k-f)\text{-flow subgraph}\} && (2) \end{aligned}$$

$$= \Pr\{S \text{ is a } (k-f)\text{-dpath subset}\} \qquad (3)$$

If we define $AF_r = \{S \subset A : S \text{ is a } r\text{-dpath subset}\}$, then we have,

Proposition 2.5 *For any directed k-flow graph G with unit capacities and unit failure decrements,*

$$RF(G, \boldsymbol{p}, f) = \Psi^d(G, \boldsymbol{p}, k-f) = \sum_{S \in AF_{k-f}} \prod_{e \in S} (1-p_e) \prod_{e \in \bar{S}} p_e \qquad (4)$$

Of course, Equation 4 does not lead directly to an efficient algorithm to compute $RF(G, \boldsymbol{p}, f)$, since $|AF_{k-f}|$ can grow exponentially in the size of G even when $|k-f| = 1$. Section 4 shows one of the situations where Equation 4 can be used to compute $RF(G, \boldsymbol{p}, f)$ efficiently.

We end this section by noting that one case of Proposition 2.5 can be generalized to the case of general capacities.

Proposition 2.6 *Let $G = (N, A, \mathbf{c}, s, t)$ be a k-flow graph with general integer capacities* **c** *and unit failure decrements. Then*

$$RF(G, \boldsymbol{p}, k-1) = \Psi(G, \boldsymbol{p}) \qquad (5)$$

Proof: Let S be the set of failed arcs, and let $\boldsymbol{c}_S$ be the capacities obtained by decreasing the capacity of each of the arcs in S by one unit. Then S is a non-$(k-1)$-flow cut for G if and only if there exists a flow $\boldsymbol{\xi} = \{\xi_e;\ e \in A\}$ of value $(k-1)$ with respect to c_S, which we can further assume to be integer. Since the network $G_{\boldsymbol{\xi}} = (N, A, \boldsymbol{\xi}, s, t)$ is an $(k-1)$-flow-graph and G itself is a k-flow-graph, then it follows that the flow graph $\bar{G}_{\boldsymbol{\xi}} = (N, A, \boldsymbol{c} - \boldsymbol{\xi}, s, t)$ must be a 1-flow-graph for which S is contained in the set of nonzero capacity arcs in $\bar{G}_{\boldsymbol{\xi}}$. The proposition follows. ▮

We note that Proposition 2.6 can be extended to threshold values less than $k-1$, although the characterization is now dependent upon the capacity values as well as the structure of the graph, and is not particularly instructive with regard to this paper.

3 Analysis of Project Scheduling Systems

We next analyze the project scheduling problem. Let $G = (N, A, \boldsymbol{a}, s, t)$ be an activity precedence graph, with task completion times $\boldsymbol{a}$ and unit failure increments for all arcs. Let d^* be the minimum project duration, and $d \geq d^*$ the given threshold project completion target. Hagstrom [5] has shown that the problem of computing $RP(G, p, d)$ is NP-hard. In this section we show that the more specific problem of computing $RP(G, p, d^*+1)$ is also NP-hard, but that the problem can be reduced to a path problem in the special case where G is planar.

To characterize the coherent binary system underlying the threshold project scheduling problem, we note first the well-known fact that a given state of arc operation allows project completion by time d if and only if the *longest path* from s to t — with arc lengths set according to the arc states — has length less than or equal to d. It follows that a cutset for this system is any set of tasks D whose failure causes some path Γ of tasks to have length greater than d. Let us call such a set a *d-deadline-cut.* The following proposition characterizes minimal and minimum cardinality d-deadline cuts. We use the notation throughout that for any (s,t)-path Γ, $L(\Gamma)$ will be the length of Γ using the normal duration times $\boldsymbol{a}$.

Proposition 3.1 *Let G be an activity precedence graph with unit failure increments, d^* the minimum project duration, and l^* the minimum number of arcs in an (s,t)-path. Then for any subset D of arcs and any threshold project completion target d, $d^* \leq d < d^* + l^*$:*

(*i*) *D is a minimal d-deadline-cut if and only if D is contained in every path Γ with $L(\Gamma) = d - |D \cap \Gamma| + 1$ and there is at least one such path.*

(*ii*) *D is a minimum cardinality d-deadline-cut if and only if $|D| = d - d^* + 1$ and D is contained in an (s,t)-path Γ^* for which $L(\Gamma^*) = d^*$.*

Proof: (*i*): For any path Γ, let $L_D(\Gamma)$ denote the length of Γ after increasing any arcs in D by one unit. Note that $L_D(\Gamma) = L(\Gamma) + |D \cap \Gamma|$. Now D is a minimal d-deadline-cut if and only if there exists an (s,t)-path Γ^0 with $L_D(\Gamma^0) = L(\Gamma^0) + |D \cap \Gamma^0| \geq d+1$. Further, D is arc-minimal if and only if in addition each proper subset D' of D has the property that $L(\Gamma) + |D \cap \Gamma| \leq d$ for every (s,t)-path Γ. Part (*i*) follows from the above two observations.
(*ii*): Simply note that a d-deadline-cut must have cardinality at least $d - d^* + 1$, and that this can be achieved by taking any (s,t)-path Γ^* — of length d^* — and choosing any $d - d^* + 1 \leq l^* \leq |\Gamma^*|$ arcs to be the d-deadline cut. Part (*ii*) now follows from part (*i*). ▮

In Figure 3, each arc e is labeled with its normal task duration a_e. Let $d = 8$ be the given threshold project completion target. Then $\{(s,b), (b,d), (d,f), (f,t)\}$ is a minimal 8-deadline-cut of size 4, while $\{(s,a), (c,e), (e,t)\}$ is a minimum cardinality 8-deadline-cut of size 3.

We will say that an activity precedence graph is *deadline r/d-survivable* (*r/d-critical*) if it is r-survivable (r-critical) with respect to $RP(G, \mathrm{p}, d)$.

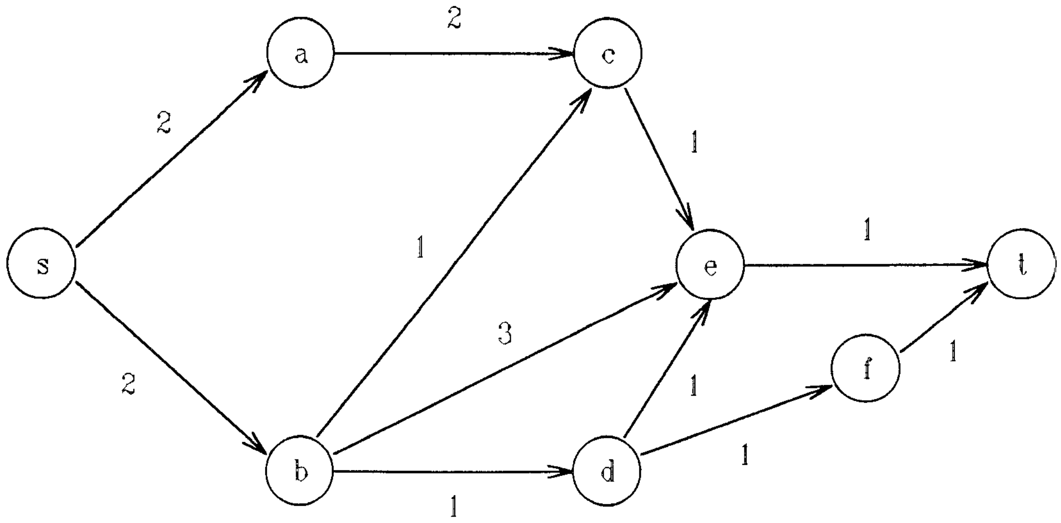

Figure 3: Activity Precedence Graph

The system given in Figure 3, for example, is r/d survivable for $d \geq 10$ or $0 \leq r < d-6$, but r/d-critical for no pair r and d. The graph in Figure 4, however, is $r/(r+19)$-critical for $r = 0, 1, 2, 3$ (and *not* $r = 4$), and r/d-survivable for $d \geq 24$ or $r \leq d - 19$. In general, we get from Proposition 3.1 that as long as $d < d^* + l^*$ a threshold project scheduling system is r/d-survivable if and only if $r \leq d - d^*$. As in the threshold flow systems, it turns out that r/d-critical activity precedence graphs have a similar simple structure which depends on the activity precedence graph itself rather than the particular value of r or d.

Proposition 3.2 *Let $G = (N, A, \boldsymbol{a}, s, t)$ be an activity precedence graph with unit failure decrements . Then the following are equivalent*

(i) G is deadline r/d-critical for some r, d with $d - d^* = r < l^*$;

(ii) G is deadline r/d-critical for every *r, d with $d - d^* = r < l^*$;*

(iii) every arc of G is in an (s,t)-path of length d^*;

(iv) every (s,t)-path in G has the same length d^*.

Proof: $(i) \Rightarrow (iii)$: Since G is deadline r/d-survivable, and every (s,t)-path must have at least $l^* > r$ arcs then it must be that every (s,t)-path in G has length at most $d - r$. Now in order for G to be r-critical with respect to d, it must be that every arc e of G must be in a d-deadline-cut of cardinality $r + 1$, which by Lemma 3.1 implies that every arc must be in an (s,t)-path of length $d - r = d^*$.
$(iii) \Rightarrow (iv)$: Suppose there exists an (s,t)-path Γ having $L(\Gamma) < d^*$. Label each node v in G by the length d_v of the longest path from s to v. Since $L(\Gamma) < d^*$, there must be a first arc $e = (u,v)$ on Γ with $a_e < d_v - d_u$. By the labeling, the longest path from s to u is of length d_u and the longest path from v to t is of length $d^* - d_v$. But this means that the longest (s,t)-path containing e must have length $d_u + a_e + (d^* - d_v) < d^*$, contradicting (iii).
$(iv) \Rightarrow (ii)$: Choose r and d satisfying $d - d^* = r < l^*$. Since every (s,t)-path of G has the same length d^*, then G is r-survivable, and Proposition 3.1 insures that every arc is in a d-deadline-cut.
$(ii) \Rightarrow (i)$: This is immediate. ▮

As in Proposition 2.3, Proposition 3.2 allows us to refer to activity precedence graphs as simply being deadline-critical, without having to specify r or f. Figure 4 illustrates a deadline-critical activity precedence graph. We remark that the l^* given in Proposition 3.2 is not the best possible, since G is 3/22-critical. Proposition 3.2, together with Proposition 3.1, gives the following useful corollary. The proof is immediate.

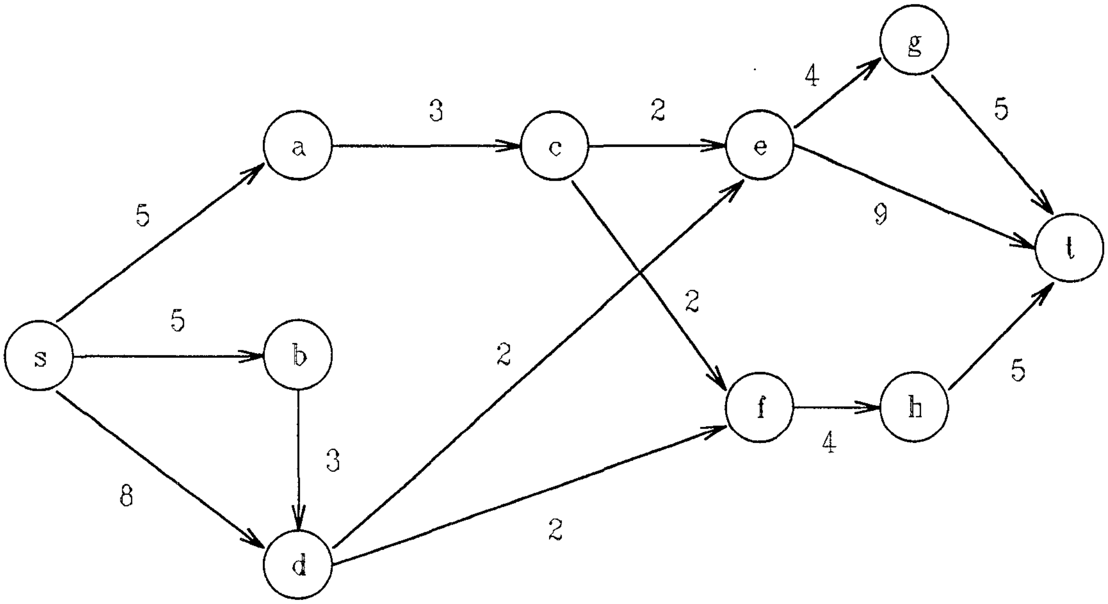

Figure 4: A Deadline-critical Activity Precedence Graph

Corollary 3.3 *Let G be a deadline-critical graph with d^* and l^* as in Proposition 3.1. Then for any threshold project completion target d, $d^* \leq d < d^* + l^*$,*

all minimal d-deadline-cuts have the same cardinality $d - d^* + 1$.

We next characterize the complements of the pathsets for project scheduling systems, i.e. the "non-cutsets". They are most easily described in terms of special kinds of cuts. Let C be an (s,t)-cut and $(X, \bar{X})$ the associated node partition. Then, C is *uniformly directed* if there are no arcs in A directed from $\bar{X}$ to X. In Figure 4, the (s,t)-cut with node partition $(\{s,a,b,c,d\}, \{e,f,g,h,t\})$ is uniformly directed, while the (s,t)-cut with node partition $(\{s,a,b,d,f\}, \{c,e,g,h,t\})$ is not.

Proposition 3.4 *Let G be a deadline-critical activity precedence graph with d^* and l^* as in Proposition 3.2, and let d, $d^* \leq d < d^* + l^*$, be the threshold project completion target. Then for any subset D of arcs, D is a non-cutset* *if and only if there exists a set of $d - d^*$ uniformly directed (s,t)-cuts whose union contains D.*

Proof: From Corollary 3.3 and Proposition 3.1, we get that D is a d-deadline-cut if and only if G has an (s,t)-path containing $d - d^* + 1$ arcs of D. Since no (s,t)-path can cross a uniformly directed (s,t)-cut in more than one arc, then if D is a d-deadline-cut it requires at least $d - d^* + 1$ uniformly directed cuts for their union to contain D.

Conversely, suppose that D is a non-cutset, so that no (s,t)-path contains more than $d - d^*$ arcs of D. Define the length vector $\boldsymbol{a}'$ having $a'_e = 1$ for all arcs in D and $a'_e = 0$ otherwise. Label each node with the length of the longest path from s to that node, using these lengths. Since D is a non-cutset, then no node of G can have label more than $d - d^*$. Now, associate with each label j the set of arcs directed from nodes with label less than j into nodes labeled j or greater. This is a uniformly directed cut since every arc directed out of a node with label j must be directed into a node with a label of at least j. Furthermore, for each j, there exists a path containing j arcs of D joining s to a node labeled j. Finally, any arc of D directed into a node labeled j must belong to the j'th uniformly directed cut, and so the collection of uniformly directed cuts so defined contains every arc of D. Since there are at most $d - d^*$ of these, the proposition follows. ▮

As in the threshold flow problem, the property of criticality in the threshold project scheduling reliability analysis problem immediately gives the reliability when $d = d^*$, since again,

$$RP(G, \boldsymbol{p}, d^*) = \prod_{e \in A} p_e.$$

The next result shows that for nonplanar graphs, this is in some sense the best possible efficient case.

Proposition 3.5 *The computation of $RP(G, \mathbf{p}, d)$ is NP-hard, even when G is deadline-critical and $d = d^* + 1 = 1$.*

Proof: Provan and Ball [9] have shown that the following problem is NP-hard:

ANTICHAIN

Given: acyclic graph $P = (V, E)$

Find: the number of *antichains* (sets of noncomparable arcs) of P

Define G to be the activity precedence graph defined on P by setting $a_{ij} = 0$ for every arc of P. Clearly G is deadline-critical, with $d^* = 0$. We prove that the number of antichains in P is equal to $2^n RP(G, \frac{1}{2}, 1)$ where $\frac{1}{2}$ is the vector of 1/2's and n is the number of arcs in P. It follows that ANTICHAIN can be reduced to a threshold scheduling reliability analysis problem with the restrictions given above.

We first make the important observation that $2^n RP(G, \frac{1}{2}, 1)$ is the number of pathsets of the threshold system G. From Proposition 3.4 we know that a set of arcs is a pathset for $RP(G, \frac{1}{2}, 1)$ if and only if its complement is contained in $d = 1$ uniformly directed cut. Computing $RP(G, \frac{1}{2}, 1)$ is therefore equivalent to counting the number of arcsets that are subsets of uniformly directed (s, t)-cuts of G. But since the collection of subsets of uniformly directed cuts of G are precisely the antichains of P, the propositions follows. ▮

As in the case of threshold flows a problem equivalent to the one treated in the proposition can be shown to be #P-complete.

The final result of this section relates $RP(G, \boldsymbol{p}, d)$ to the function $\Psi(G, \boldsymbol{p})$ given in Section 1.4, in the case where the activity precedence graph is *planar.*

Theorem 3.6 *Let G be a planar deadline-critical activity-precedence graph with unit failure increments, let d^* and l^* be as in Proposition 3.2 and let G^* be the (s,t)-dual graph to G. Then, for any project completion target d, $d^* \leq d < d^* + l^*$,*

$$RP(G, \mathrm{p}, d) = \Psi(G^*, \boldsymbol{p}, d - d^*) \qquad (6)$$

Proof: If we construct the planar (s,t)-dual graph G^* to G ([6] pp.33–34) with $\bar{s}$ and $\bar{t}$ being the corresponding dual source and sink, then the $(\bar{s}, \bar{t})$-paths in G^* correspond precisely to the uniformly directed (s,t)-cuts of G, and so from Proposition 3.4 we get that the non-cutsets in G correspond to arcs in G^* which are contained in the union of $d - d^*$ $(\bar{s}, \bar{t})$-paths in G^*. The theorem follows. ▮

Two things should be noted here. First, the paths in Theorem 3.6, unlike those for Theorem 2.5, do not have to be disjoint. Thus the k-critical cases of these two problems when $k > 1$ will have quite different solution techniques, while the 1-critical cases covered in this paper can be solved by the same algorithm. Second, note that Theorem 3.6 holds for general duration times $\boldsymbol{a}$, while Theorem 2.5 requires unit capacities. Theorem 2.5, on the other hand, does not require planarity of G.

We end the section by addressing the inclusion of dummy activities into the project scheduling problem. If we continue with the activity-on-arc representation, precedence relations may require the inclusion of dummy arcs, which are then assigned a deterministic task length of 0. The number of dummy arcs necessary is no more than the number of precedence pairings originally specified for the project. In our discussion so far, we assumed that all tasks could potentially fail. The discussion can be extended to cover the possibility that there are dummy tasks in the project if we restrict our specification of a deadline-cut D by requiring in addition that D contain no dummy arcs. Equation 6 continues to hold by simply putting $p_e = 1$ for all dummy arcs e.

4 Results for 1-Critical Systems

4.1 Algorithm for Analyzing 1-Critical Systems

In this section we present the algorithm 1-PATH, which computes the probability $\Psi(G, \boldsymbol{p})$ that the failed arcs in an acyclic graph $G = (N, A)$ form a 1-path subset. Sections 2 and 3 show that such an algorithm will compute both $RF(G, \boldsymbol{p}, k-1)$ for k-flow graphs with unit failure decrements and $RP(G, \boldsymbol{p}, d^* + 1)$ for planar deadline-critical activity precedence graphs with unit failure increments. The algorithm given here is based on the fact that the vertices and arcs of any acyclic graph G form a *partial order* where, for elements $a, b \in E \cup N$, we say that *a precedes b* ($a \preceq b$) whenever there is a directed path on which b follows a. It follows that the operating states for $\Psi(G, \boldsymbol{p})$ are exactly those states for which the failed arcs form a *chain* in this partial order, which means that any pair of failed arcs is comparable. Thus the problem is reduced to that of computing the probability that the failed arcs of G form a chain.

The algorithm assumes that the acyclic graph G input is a source-sink acyclic graph $G = (N, A, s, t)$ with s the unique *initial* node of G — i.e. node having in-degree 0 — and t the unique *terminal* node of G — i.e. node having out-degree 0. We note that the flow and dual activity precedence graphs given in Sections 2 and 3 have this property. On the other hand, if we wish to compute $\Psi(G, \boldsymbol{p})$ on a graph that does not have this property, then G can be modified by adding a super-source s and super-sink t, connecting s to each original initial node with a perfectly reliable arc and connecting each original terminal node to t with a perfectly reliable arc. The resulting source-sink graph $G = (N, A, s, t)$ now has s and t the unique initial and terminal nodes, respectively, and there is a 1-1 correspondence between chains in the two graphs with corresponding chains having equal weights. Further, the number of added edges is at most 2×(the number of nodes in the original graph).

We next define, for any node v of G, the auxiliary graph $G^v = (N^v, A^v, s, v)$ to be the subgraph of arcs and vertices which precede v in the partial order. From our assumption, we have immediately that $G = G^t$, and hence $\Psi(G, \boldsymbol{p}) = \Psi(G^t, \boldsymbol{p})$. Now for auxiliary graph G^v, define $\Psi^o(G^v, \boldsymbol{p})$ to be the probability that the set of failed arcs in G forms a 1-path subset *whose highest arc points into v.* Observe that for each state contributing to $\Psi(G, \boldsymbol{p})$, there is a *unique* highest arc in the chain of failed arcs, and hence a unique highest node v into which this arc points. It follows that the states contributing to $\Psi(G, \boldsymbol{p})$ can be partitioned into events according to the highest node into which a failed arc points. Specifically, we have recursive equation

$$\Psi(G^v,\boldsymbol{p}) \;=\; \sum_{u\in N^v} Pr\left\{\begin{array}{c}\text{the set of failed arcs in } G^v \text{ forms a 1-path}\\ \text{subset whose highest arc points into } u \textit{ and}\\ \text{all arcs of } G^v \text{ not preceding } u \text{ operate}\end{array}\right\}$$

$$= \sum_{u\preceq v} \Psi^o(G^u,\mathrm{p}) \prod_{\substack{e\,\in\,E\\ e\,\preceq\,v,\;\; e\,\npreceq\,u}} p_e. \qquad (7)$$

On the other hand, for each state contributing to $\Psi^o(G^v,\boldsymbol{p})$, there is a unique arc $e = (u,v)$ in the chain of failed arcs which points into v, and so these states can be partitioned into events according to which arc is the final arc of the chain. Specifically, we have second recursive equation

$$\Psi^o(G^v,\boldsymbol{p}) \;=\; \sum_{u\in N^v:(u,v)\in E} Pr\left\{\begin{array}{c}\text{the set of failed arcs in } G^v \text{ below } u\\ \text{forms a 1-path subset } \textit{and}\\ \text{arc } (u,v) \text{ fails } \textit{and}\\ \text{all other arcs of } G^v \text{ not preceding } u \text{ operate}\end{array}\right\}$$

$$= \sum_{(u,v)\in E} \Psi(G^u,\mathrm{p}) \;\; (1-p_{(u,v)}) \prod_{\substack{(u,v)\,\neq\,e\,\in\,E\\ e\,\preceq\,v,\,e\,\npreceq\,u}} p_e. \qquad (8)$$

Finally, we have the trivial initial equation

$$\Psi(G^s,\boldsymbol{p}) \;=\; 1. \qquad (9)$$

Now $\Psi(G^v,\boldsymbol{p})$ can be computed by solving equations (8) followed by (7), in order of nondecreasing v with respect to $\preceq$. This requires evaluating $n-1$

formulae of each type, each formula of which has $O(n)$ terms, where $n = |N|$. Each term, in turn, involves computing products of the form

$$\alpha(u, v) \quad = \quad \prod_{\substack{e \in E \\ e \preceq v,\ e \npreceq u}} p_e$$

for $u \preceq v \in N^v$. To obtain $O(n^2)$ complexity, we need to show that these products can be computed in constant time after a preprocessing step. We first define, for any $v \in N^v$,

$$\beta(v) = \prod_{e \preceq v} p_e.$$

It follows that for any pair of vertices $u \preceq v$,

$$\alpha(u, v) \quad = \quad \frac{\beta(v)}{\beta(u)}. \tag{10}$$

The term $\beta(v)$ can be easily computed by setting

$$\gamma(v) = \prod_{e=(u,v) \in A} p_e \tag{11}$$

and then

$$\beta(v) = \prod_{u \preceq v} \gamma(u). \tag{12}$$

The equations for computing $\beta(v)$ and $\gamma(v)$ involve at most n arithmetic operations, and moreover, after computing the n values for $\beta()$ and $\gamma()$, we have that $\alpha(u, v)$ can be obtained using one arithmetic operation, as required. These equations can in fact be computed in the process of evaluating $\Psi(G, \mathbf{p})$. We thus obtain the algorithm 1-PATH, which computes the probability that

INPUT: Source-sink acyclic graph $G = (N, A, s, t)$, probability vector $\boldsymbol{p}$

OUTPUT: $\Psi(G, \boldsymbol{p})$

ALGORITHM:

```
for $v \in N$ in nondecreasing order (with respect to $\preceq$) do
    Step 1: Compute $\gamma(v)$ using (11)
    Step 2: Compute $\beta(v)$ using (12)
    Step 3: if $v \neq s$ then compute $\Psi^o(G^v, \boldsymbol{p})$ using (8), where again the
        appropriate value of $\alpha(u, v)$ is found using (10);
    Step 4: Compute $\Psi(G^v, \boldsymbol{p})$ using (7) or (9), where the appropriate
        value of $\alpha(u, v)$ is found using (10);
end do
output $\Psi(G^t, \mathbf{p})$.
```

Figure 5: 1-PATH Algorithm

all failed arcs in an acyclic graph, G lie on a single path. The algorithm is given Figure 5.

We now summarize the analysis given in this section with,

Theorem 4.1 *Algorithm 1-PATH correctly computes $\Psi(G, \boldsymbol{p})$ in $O(n^2)$ steps.*

Proof: The correctness of the algorithm follows from the formula derivations given earlier in this section. The construction of the nondecreasing order on N performed at the beginning of 1-PATH can be accomplished in $O(n \log n)$ time through classic sorting algorithms. The index sets for each of the formulae can be found in $O(n)$ time. It follows that the number of computational steps involved in computing any of the formulae (1)-(6) is $O(n)$ and the result follows. ▮

We can now apply this analysis to demonstrate the solvability of the threshold flow and project management problems introduced in the first section.

Proof and Theorems 1.1 and 1.2: From Proposition 2.6 and Proposition 3.2 we have that, for 1-critical systems, the threshold flow problem with unit failure decrements and the planar project management problem with unit failure increments both reduce to the problem of computing $\Psi(G, \boldsymbol{p})$. Theorem 4.1 now implies the results.

4.2 Extension to 1-Critical Threshold Flow Problems Defined on Undirected Graphs

An interesting and useful property of many network reliability problems is that undirected problems can be transformed into equivalent directed problems. In the flow graph setting, this transformation consists of replacing each arc in an undirected graph by its asymmetric pair of directed arcs, where each directed arc inherits the failure probability and capacity of the undirected arc from which it was derived. Hagstrom [4] showed that this transformation is reliability preserving in the stochastic flow graph setting. This fact provides a means for transforming the threshold flow reliability analysis problem defined on an undirected $1/f$-critical flow graph with unit failure decrements into a directed threshold flow problem. However, the resulting directed graph is clearly not $1/f$-critical since it is not acyclic. It turns out that 1-PATH can be used to compute $RF(G, \mathbf{p}, f)$ by applying it to a particular acyclic subgraph. The algorithm is described in Figure 6

The validity of the algorithm is given by the following Proposition.

Proposition 4.2 *All arcs deleted in Step 3 are irrelevant with respect to a unit failure decrement threshold flow system with threshold* $f = f^* - 1 = k - 1$.

Before proving this proposition we define a certain network flow construct, a *flow decrementing path.* The *flow augmenting path* is a fundamental concept in network flow theory. It has the property that if a flow of value f is not maximum then there exists a flow augmenting path that produces a flow of value $f + 1$ from the flow of value f. Similarly, for any flow of value, $f > 0$, there exists a flow decrementing path that produces a flow of value $f - 1$. Moreover, for any flow of value f that is not optimal there exists a flow of value $f + 1$ and a flow decrementing path that converts the flow of value $f + 1$ to the flow of value f. A flow decrementing path is a path from the sink node t to the source node s. If x is the flow vector, then the directed

INPUT: An undirected k-flow graph, G, with unit failure decrements and a probability vector $\boldsymbol{p}$ defined on the arcs of G.

OUTPUT: $RF(G, \mathrm{p}, k-1)$

ALGORITHM:

Step 1: Replace each undirected arc with its asymmetric pair of directed arcs, where each directed arc inherits the failure probability and capacity of the undirected arc from which it was derived.

Step 2: Find an acyclic max flow (note that the max flow must have value k).

Step 3: Delete all arcs which carry no flow (note that the resultant directed graph is a k-flow graph since we started with an acyclic max flow).

Step 4: Apply the 1-critical directed graph algorithm to the network constructed in Step 3.

Figure 6: Undirected Threshold Flow Algorithm

arc (i,j) may be on a flow decrementing path if $x_{ji} > 0$ or if $x_{ij} < c_{ij}$. We can now proceed with the proof.

Proof: (of Proposition 4.2) Let $\mathbf{x} = \{x_{ij}\}$ be the acyclic max flow vector found by the algorithm. Since G is a k-flow graph it must be that for any undirected arc, $[i,j]$, either $x_{ij} = c_{[i,j]}$ or $x_{ji} = c_{[i,j]}$. We prove the Proposition by showing that if x_{ji} is zero in the acyclic flow of size k then x_{ji} must be zero in any acyclic flow of size $k-1$. Consider any acyclic flow of size $k-1$. Since the flow of size k is unique, there exists a flow decrementing path that converts the flow of size k to the flow of size $k-1$. For any undirected arc $[i,j]$ the flow decrementing path must traverse the arc in the direction (j,i) where $x_{ij} = c_{[i,j]}$. Now in implementing the flow decrementing path we have the choice of setting $x_{ji} = 1$ or $x_{ij} = c_{[i,j]} - 1$. If we set $x_{ji} = 1$ then we obtain a cycle so to get an acyclic flow we must set $x_{ij} = c_{[i,j]} - 1$. Thus, no flows that were previously zero become non-zero, i.e. if $x_{ji} = 0$ in the flow of size k then $x_{ji} = 0$ in the flow of size $k-1$ and the result follows. ▮

We now have

Corollary 4.3 *The threshold flow reliability analysis problem can be solved in $O(n^2)$ time for undirected 1-critical systems with unit failure decrements.*

4.3 Properties of Minimal Cutsets for 1-Critical Threshold Flow Systems

In Section 3, we observed that for r-critical project scheduling systems with unit failure increments, all minimal d-deadline-cuts had the same cardinality.

We note that this property is not in general true for threshold flow systems. That is, for r-critical threshold flow systems with unit failure decrements it is not always the case that all *minimal* flow-cuts are *minimum cardinality* flow-cuts. For example, in Section 2 minimal flow-cuts of size 3 and 4 were given for a 2-critical threshold system. In this subsection we show that this property does hold for 1-critical systems.

The main result of this subsection requires the following Lemma which is also used as part of the bounding arguments in the final section.

Lemma 4.4 *Given an acyclic graph, G and an arc subset, A', then exactly one of the following holds*

a.) *A' is 1-path subset,*

b.) *there exists an $\hat{A} \subset A'$ with $|\hat{A}| = 2$ where $\hat{A}$ is contained in some uniformly directed cut.*

Proof:

1. Suppose for all $e_1, e_2 \in A', e_1 \neq e_2$, either $e_1 \preceq e_2$ or $e_2 \preceq e_1$. Since G is acyclic this property implies that we can totally order the elements of A' and find a single path that contains A' so that case a.) occurs.

2. If the conditions given in 1.) do not occur then A' contains a 2-antichain $\{e_1, e_2\}$. Let $S = \{\ v \in N \mid v \preceq e_1 \text{ or } v \preceq e_2\}$. Then, the set of arcs directed out of S is a uniformly directed (s,t)-cut which contains $\{e_1, e_2\}$ and case b.) occurs.

∎

We can now derive the cutset property.

Proposition 4.5 *For 1-critical threshold flow systems with unit failure decrements, all mincuts (minimal flow-cuts) have cardinality 2, i.e. the set of mincuts is the same as the set of minimum cardinality cutsets.*

Proof: Let G be the $f+1$-flow graph associated with the 1-critical threshold flow system. Since all arcs are saturated by the unique max-flow, the capacity of any uniformly directed cut equals $f+1$ (the value of the max flow). Now consider an A' with $|A'| > 2$. If condition a.) of the Lemma occurs then A' is a 1-path subset which implies that it is not a flow-cut. If condition b.) occurs then since uniformly directed cuts have capacity $f+1$, there is a $\hat{A}$ strictly contained in A' which is a mincut. This implies that A' is a flow-cut but it is not minimal. The result now follows since no A' with $|A'| > 2$ can be a mincut. ▮

We feel that it may be possible exploit this property in further work on this problem. Note that we did not use it directly in developing an efficient algorithm for 1-critical systems. One might be tempted to list all minimal flow-cuts, which could certainly be done in no more than $O(|A|^3)$ time by the proposition, and then to compute the reliability from this list. The problem of computing system reliability given a list of minpaths or mincuts is called the union of products problem. In [2], it is shown that the union of products problem is NP-hard even when the set of minpaths or mincuts input all have cardinality two. Thus, a direct application this approach does not obviously lead to an efficient algorithm. What the results of this paper show is that the threshold flow problem for critical systems provides a solvable special case of the union of products problem.

5 Bounds for Almost Critical Threshold Flow Systems

In this section we present a method for generating bounds on $RF(G, \boldsymbol{p}, f)$ for systems that are "near-critical". We will not give a formal definition of near-critical, however, we will assume that the system is survivable and that there exist some arcs in flow-cuts of size 2. This is equivalent to assuming that $f^* = f + 1$. The qualitative requirement that makes the system near-critical is that a "large percentage" of the arcs should be in some flow-cut of size 2. Note that we continue to assume that unit failure decrement model.

Our approach makes use of the flow reduced graph which is defined in Ball and Provan [1] (equivalent structures are described by Picard and Queyranne[7] and Gardner[3]; see also Provan and Shier[10]). After executing a max-flow algorithm the arcs can be partitioned into three sets:

critical arcs (denoted by AC): the arcs that are in all flows of size f^* (these are simply the arcs in some min-capacity cut).

back arcs (denoted by AB): the arcs that are in no flows of size f^*.

redundant arcs (denoted by AR): the arcs that are in some flow of size f^* but not all flows of size f^*.

The flow reduced graph is a directed acyclic graph. Its arc set consists of the union of the critical arcs and the back arcs, where the orientation of each back arc is reversed. The redundant arcs are collapsed into pseudo-nodes. If we let NR be the set of nodes in the flow-reduced graph and AB' be the

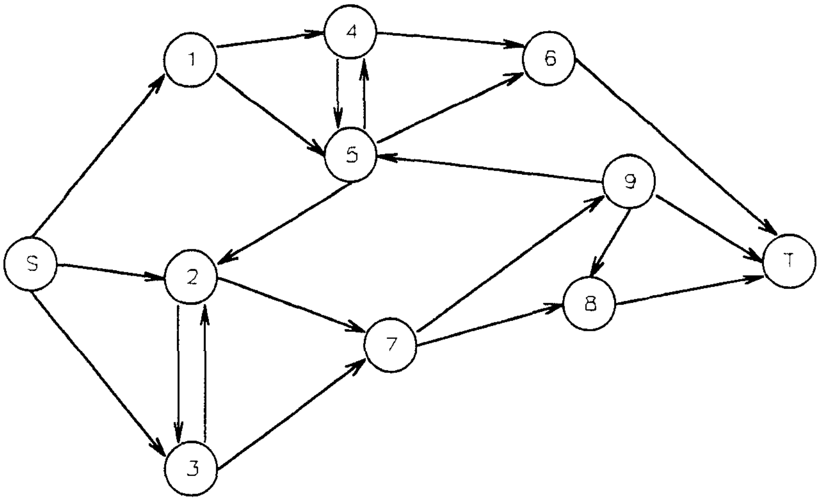

Figure 7: Flow Graph with $f^* = 3$

set of back arcs with reversed orientation then, the flow reduced graph is $(NR, AC \cup AB')$. In the graph in Figure 7,

$$\begin{aligned} AC &= \{(s,1),(s,2),(s,3),(2,7),(3,7),(7,8),(7,9),(6,t),(8,t),(9,t)\}, \\ AB &= \{(5,2),(9,5),(9,8)\} \\ AR &= \{(1,4),(1,5),(4,5),(5,4),(5,6),(4,6),(2,3),(3,2)\} \end{aligned}$$

Figure 8 illustrates the corresponding flow reduced graph. Note that redundant arcs, $\{(1,4),(1,5),(4,5),(5,4),(5,6),(4,6)\}$ have been collapsed into pseudo-node 1' and redundant arcs, $\{(2,3),(3,2)\}$ have been collapsed into pseudo-node, 2'.

An important property of the flow reduced graph (see [1]) is:

A pair of arcs is contained in a minimum capacity (s,t)-cut if and

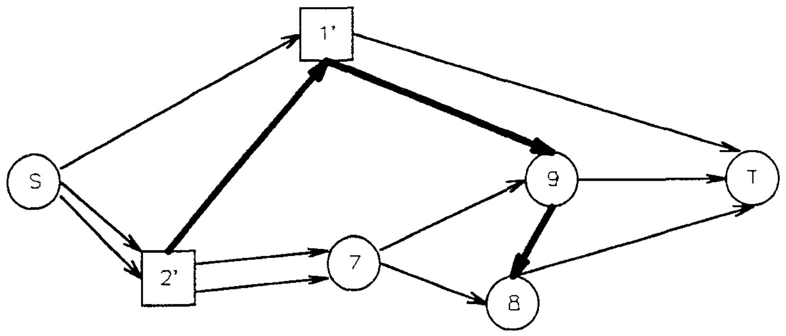

Figure 8: Flow Reduced Graph (Back Arcs Indicated in Bold)

> only if the arcs are critical and are contained in some uniformly directed cut in the flow reduced graph.

This, in turn, implies:

> A pair of arcs is a minimum cardinality flow cut, i.e. a flow cut of size two, if and only if the arcs are critical and they do not lie on a common path in the flow reduced graph.

This property together with Lemma 4.4 imply that the only mincuts (minimal flow cuts) consisting entirely of critical arcs have cardinality 2. Based on the preceding discussion it now follows that we can apply algorithm 1-PATH to compute:

$RF(G, \boldsymbol{p}, f^* - 1 \mid \text{all non-critical arcs operate})$.

$$= \Pr\{\text{no cutset of size 2 fails}\}$$

$$= \Psi((NR, AC \cup AB'), \boldsymbol{p}_{AC} \times \underline{1})$$

where $\boldsymbol{p}_{AC}$ is the probability vector $\boldsymbol{p}$ restricted to AC and $\underline{\mathbf{1}}$ is the vector of 1's over AB'. We also have that

$$RF(G, \boldsymbol{p}, f^* - 1 \mid \text{all redundant arcs operate and all back arcs fail})$$

$$= \Psi((NR, AC), \boldsymbol{p}_{AC})$$

In our lower bound we will use $\Psi((NR, AC \cup AB'), \boldsymbol{p}_{AC} \times \underline{1}) - \Psi((NR, AC), \boldsymbol{p}_{AC})$ which can be interpreted as the probability that the threshold flow system operates with flow threshold $f^* - 1$ and at least one operating back arc is necessary to insure system operation given that all redundant arcs operate.

Since

$$\Pr\{\text{some cutset of size 2 fails}\}$$

$$= 1 - \Psi((NR, AC \cup AB'), \boldsymbol{p}_{AC} \times \underline{1})$$

$$\leq 1 - RF(G, \boldsymbol{p}, f^* - 1)$$

we immediately have the following upper bound.

$$RF(G, \boldsymbol{p}, f^* - 1) \leq \Psi((NR, AC \cup AB'), \boldsymbol{p}_{AC} \times \underline{1})$$

The flow reduced graph can also be used to generate a lower bound on $RF(G, \boldsymbol{p}, f)$. The process of forming the flow reduced graph involves shrinking redundant arcs into pseudo-nodes. We call the set of redundant arcs associated with one such pseudo-node an R-set. The set of all R-sets is denoted by $\{RS(i)\}_{i=1}^{nr}$ Thus, the set of R-sets form a partition of the redundant arcs, i.e. $AR = \cup_{i=1}^{nr} RS(i)$ and $RS(i) \cap RS(j) = \emptyset$ for $i \neq j$.

Proposition 5.1 *Let RS be any R-set and (NS, RS) be the graph induced by RS. Let IN be the set of critical arcs directed into NS and OUT the set of critical arcs directed out of NS. Let e be any arc in RS, then there exists a flow from the tail nodes of IN through IN, $(NS, RS - e)$ and OUT to the head nodes of OUT of value $\mid IN \mid = \mid OUT \mid$.*

Proof: The result follows from the fact that critical arcs are in all max flows and for each redundant arc, e, there exists a max flow that does not contain e.▮

An immediate consequence of this proposition is,

Corollary 5.2 *If an arbitrary arc is deleted from each of the R-sets then there still exists a flow of value f^* that uses only redundant arcs and critical arcs.*

Define

$$\Gamma'(S, k) = \Pr\{k \text{ or fewer components in the set } S \text{ fail}\}$$

$$\Gamma^o(S, k) = \Pr\{\text{exactly } k \text{ components in the set } S \text{ fail}\}$$

We can now state our lower bound.

Theorem 5.3 $RF(G, \mathbf{p}, f^* - 1) =$

$Pr\{$graph supports a flow of size $f^* - 1$ and at least 2 redundant arcs in some R-set fail$\}$ +

Pr{graph supports a flow of size $f^* - 1$ and at least 1 redundant arc fails but at most 1 redundant arc in each R-set fails} +

Pr{graph supports a flow of size $f^* - 1$ and no redundant arcs fail}

$$
\begin{aligned}
&\geq \\
&\Gamma^o(AC,0)\sum_{i=1}^{nr}\left(\Gamma^o(RS(i),2)\prod_{j\neq i}\Gamma'(RS(j),1)\right) \\
&+ \\
&\Gamma'(AC,1)\left[\left(\prod_{i=1}^{nr}\Gamma'(RS(i),1)\right)-\Gamma^o(AR,0)\right] \\
&+ \\
&\Gamma^o(AR,0)[(\Psi((NR,AC\cup AB'),\boldsymbol{p}_{AC}\times\underline{1})-\Psi((NR,AC),\boldsymbol{p}_{AC})\Gamma^o(AB,0) \\
&+\Psi((NR,AC),\boldsymbol{p}_{AC})]
\end{aligned}
$$

Proof: We prove the result by showing that each of the three terms given in the bound is the probability of an event contained in the respective event whose probability sum is given before the inequality.

A quick inspection of the terms in the bound indicates that the first is the probability of an event in which exactly 2 arcs in some R-set fail, the second is the probability of an event in which at least 1 redundant arc fails but at most 1 in each R-set fails and the third is the probability of an event in which no redundant arcs fail. Thus, to prove the Theorem we must show that each of the terms is the probability that a certain set of operative states occur. From Corollary 5.2, we know that if at most 1 arc fails in each R-set then there will still exist a flow of size f^*. Now the first term is the probability that exactly 1 additional arc fails where that arc is a redundant arc. The second term is the probability that at most 1 additional arc fails where that arc is a critical arc. Thus, both cases correspond to operating states since we start with a flow of f^* and delete at most 1 additional arc which can reduce the flow by at most 1. The third term is the product of the probability that all redundant arcs operate multiplied by the sum of two probabilities. The first is the probability that all back arcs operate and the system operates, where the operation of at least one back arc is necessary for system operation. The second is the probability that sufficient critical arcs operate soas to imply system operation (when combined with the operating redundant arcs). ∎

We now provide an example of the calculation of the bound given in this Theorem using the example from Figure 7. To ease the exposition we assume that all arcs fail with the common failure probability p. If we let,

$$\begin{aligned} RS(1) &= \{(1,4),(1,5),(4,5),(5,4),(5,6),(4,6)\} \text{ and} \\ RS(2) &= \{(2,3),(3,2)\}, \end{aligned}$$

we have that,

$$|AC| = 10, |AB| = 3, |RS(1)| = 6, |RS(2)| = 2,$$

and the lower bound is,

$$\begin{aligned} & p^{10}\left[\binom{6}{2}p^4(1-p)^2(p^2+\binom{2}{1}p(1-p))+\binom{2}{2}(1-p)^2(p^6+\binom{6}{1}p(1-p)^5)\right] \\ + & \left[p^{10}+\binom{10}{1}p^9(1-p)\right] \end{aligned}$$

$$\left[\left(\binom{6}{1}p^5(1-p)+p^6\right)\left(\binom{2}{1}p(1-p)+p^2\right)-(1-p)^8\right]$$
$$+\quad p^8\left[(\Psi((NR, AC\cup AB'),\boldsymbol{p}_{AC}\times \underline{1}) - \Psi((NR,AC),\boldsymbol{p}))\,p^3+\Psi((NR,AC),\boldsymbol{p})\right]$$

We note that this lower bound "contains" the exact value of Pr{system operates and 2 or fewer arcs fail}. In addition, it includes lower bounds on higher order components.

To illustrate that it is not possible to make more "liberal" use of 1-PATH in generating bounds consider once again the graph in Figure 7 with flow threshold $f^* - 1 = 2$. $\{(s,3),(2,3),(2,7)\}$ is a mincut (minimal flow-cut) of size 3 that consists of 2 critical arcs and 1 redundant arc. This implies that it would not be possible to eliminate the second term and multiply the last term by $\prod_{i=1}^{nr}\Gamma'(RS(i),1)$ rather than by $\Gamma^o(AR,0)$. Similarly, $\{(4,6),(5,6),(7,9)\}$ is a mincut of size 3 that consists of 2 redundant arcs and 1 critical arc. This implies that it would not be possible to multiply the first term by $\Gamma'(AC,1)$ rather than $\Gamma^o(AC,0)$.

6 Acknowledgements

The work of the first author was supported by NSF grant No. CDR-8803012. The work of the second author was supported by NSF grant No. CCR-9200572.

References

[1] M.O. Ball and J.S. Provan, *Calculating Bounds on Reachability and Connectedness in Stochastic Networks*, Networks 13 (1983), 253-278.

[2] M.O. Ball and J.S. Provan, *Disjoint Products and Efficient Computation of Reliability*, Operations Research, 36 (1988), 703-715.

[3] M.L. Gardner, *Application of an Algorithm for Networks*, Congressus Numeratium, 2 (1980), 31-38.

[4] J.N. Hagstrom, *Note on Independence of Arcs in Antiparallel for Network Flow Problems*, Networks 14 (1984), 567-570.

[5] J.N. Hagstrom, *Computational Complexity of PERT Problems*, Networks, 18 (1988), 139-147.

[6] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart, and Winston, 1976.

[7] J.-C. Picard and M. Queyranne, *On the Structure of All Minimum Cuts in a Network and Applications*, Math. Programming Study 13 (1980), 8-16.

[8] J.S. Provan, *The complexity of reliability computations in planar and acyclic graphs*, SIAM J. Computing 15 (1986), 694–702.

[9] J.S. Provan and M.O. Ball, *The complexity of counting cut and of computing the probability that a graph is connected*, SIAM J. Comp. 12 (1983), 777–788.

[10] J.S. Provan and D.R. Shier, *A Paradigm for Listing (s,t)-Cuts in Graphs*, Technical Report UNC/OR TR91-3, Department of Operations Research, University of North Carolina at Chapel Hill, February, 1991.

[11] L.G. Valiant, *The complexity of enumeration and reliability problems*, SIAM J. Comp. 8 (1979), 410–421.

[12] D. Vertigan, *The computational complexity of Tutte invariants for planar graphs*, preprint, Mathematical Institute, Oxford University.