[FW] R. L. FRANCIS AND J. A. WHITE, *Facility Layout and Location*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[G] R. L. GRAHAM, *An efficient algorithm for determining the convex hull of a finite planar set*, Inform. Process. Lett., 1 (1972), pp. 132–133.

[HSP] S. L. HAKIMI, E. F. SCHMEICHEL AND J. G. PIERCE, *On p-centers in networks*, Transportation Sci., 12 (1978), pp. 1–15.

[HM] G. Y. HANDLER AND P. B. MIRCHANDANI, *Location on Networks Theory and Algorithms*, MIT Press, Cambridge, MA, 1979.

[KH] O. KARIV AND S. L. HAKIMI, *An algorithmic approach to network location problems, Part* I. *The* 1*-centers*, SIAM J. Appl. Math., 37 (1979), pp. 513–538.

[L] N. LEVIN, *Location problems on weighted graphs*, unpublished thesis, Tel Aviv Univ., 1977.

[M1] N. MEGIDDO, *Applying parallel computation algorithms in the design of serial algorithms*, in Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Angeles, 1981, pp. 399–408; J. Assoc. Comput. Math., to appear.

[M2] ———, *The weighted Euclidean* 1*-center problem*, Math. Oper. Res., to appear.

[M3] ———, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach., to appear.

[NC] K. P. K. NAIR AND R. CHANDRASEKARAN, *Optimal location of a single service center of certain types*, Naval Res. Logist. Quart., 18 (1971), pp. 503–510.

[PH] F. P. PREPARATA AND S. J. HONG, *Convex hulls of finite sets of points in two and three dimensions*, Comm. ACM, 20 (1977), pp. 87–93.

[RT] H. RADEMACHER AND O. TOEPLITZ, *The Enjoyment of Mathematics*, Princeton Univ. Press, Princeton, NJ, 1957.

[Sh] M. I. SHAMOS, *Geometric complexity*, in Proc. 7th Annual ACM Symposium on Theory of Computing, 1975, ACM, New York, 1975, pp. 224–233.

[ShH] M. I. SHAMOS AND D. HOEY, *Closest-point problems*, in Proc. 16th Annual IEEE Symposium on Foundations of Computer Science, 1975, IEEE Computer Society Press, Los Angeles, 1975, pp. 151–162.

[Sm] R. D. SMALLWOOD, *Minimax detection station placement*, Oper. Res., 13 (1965), pp. 636–646.

[Sy1] J. J. SYLVESTER, *A question in the geometry of situation*, Quart. J. Math., 1 (1857), p. 79.

[Sy2] ———, *On Poncelet's approximate valuation of Surd forms*, Philosophical Mag., XX, 4th Series (1860), pp. 203–222.

[Y] A. C. YAO, *A lower bound for finding convex hulls*, J. Assoc. Comput. Mach., 28 (1981), pp. 780–787.

SIAM J. COMPUT.
Vol. 12, No. 4, November 1983

0097-5397/83/1204-0013 $01.25/0

THE COMPLEXITY OF COUNTING CUTS AND OF COMPUTING THE PROBABILITY THAT A GRAPH IS CONNECTED*

J. SCOTT PROVAN† AND MICHAEL O. BALL‡

Abstract. Several enumeration and reliability problems are shown to be #P-complete, and hence, at least as hard as NP-complete problems. Included are important problems in network reliability analysis, namely, computing the probability that a graph is connected and counting the number of minimum cardinality (s, t)-cuts or directed network cuts. Also shown to be #P-complete are counting vertex covers in a bipartite graph, counting antichains in a partial order, and approximating the probability that a graph is connected and the probability that a pair of vertices is connected.

How can one use this?

Key words. complexity, #P-complete, graphs, reliability, network reliability

1. Introduction. The inherent intractability of certain counting and reliability problems has been studied by Ball [1], Rosenthal [11], and Valiant [12]. Valiant defines the notion of the #P-complete class of counting problems, and shows that problems in this class are at least as hard as NP-complete problems. He then goes on to show that several important counting and reliability problems are #P-complete, among them, counting perfect matchings in bipartite graphs and evaluating the probability that two given nodes in a probabilistic graph are connected. Three important problems are mentioned by Ball and Valiant, for which the complexity is not known, namely:

(1) evaluating the probability that a probabilistic graph is connected,

(2) approximating the probability that a probabilistic graph is connected,

(3) approximating the probability that two vertices of a probabilistic graph are connected.

In view of results by the authors in [2], the probability measure associated with problems (1) and (2) seems to have considerably more structure than that associated with (3). In [3] they also show the power of the structure in providing good upper and lower bounds for this measure. We show in this paper, however, that all three of these problems are NP-hard, in particular, #P-complete. In the process, we show that several counting problems are also #P-complete, among them: counting the number of node covers in a bipartite graph, counting antichains in a partial order, and counting minimum cardinality directed network cuts.

We now fix some terminology. Let $G = (V, E)$ be a graph with vertex set V and edge set E and let $m = |V|$ and $n = |E|$. When specified, G directed implies that the edges are taken to be ordered pairs, and G undirected implies the pairs are unordered. When not specified, G is allowed to be either. We allow loops (edges whose two end points are the same) and multiple edges (edges with the same pair of end points), although these are not strictly required for the results of this paper. Let s and t be two vertices in the graph G (directed or undirected). An (s, t)*-path* in G is any sequence $s = v_0,\ e_1,\ v_1, \cdots,\ v_{k-1},\ e_k,\ v_k = t$ of vertices $v_0,\ v_1, \cdots$ and edges $e_1, e_2, \cdots$ with $e_j = (v_{j-1}, v_j)$ for $j = 1, \cdots, k$. An (s, t)*-cut* in G is any minimal set of edges that intersects every (s, t)-path. A *network cut* (*with respect to* s) is any minimal set of edges that is an (s, t)-cut for some vertex t in G. A *spanning tree* (*rooted at* s) is a

* Received by the editors January 23, 1981, and in final revised form October 26, 1982.

† Curriculum in Operations Research and Systems Analysis, University of North Carolina, Chapel Hill, North Carolina, 27514. This work was performed while this author was an NRC/NAS postdoctoral associate at the National Bureau of Standards.

‡ College of Business and Management, University of Maryland, College Park, Maryland 20742.

minimal set of edges that contains paths from s to all other vertices in G. Note that if G is undirected, a network cut comprises any minimal set of edges whose removal disconnects G and a spanning tree is any minimal set of edges that connects all vertices; in both cases the definition is independent of the choice of s.

We now define our reliability measures. Given any real p, $0 \leqq p \leqq 1$, we impose the stochastic structure on G in which the edges of G are subject to random failure, independently and each with equal probability p. Edges that have not failed are said to be operative. We are concerned with two composite reliability measures on this stochastic model, which we will denote as functions of p. The first is the (s, t) *connectedness measure*: given vertices s and t in G,

$$\begin{aligned} f(G, s, t; p) &= \Pr\{\text{there is a path of operative edges from } s \text{ to } t\} \\ &= \Pr\{\text{the failed edges of } G \text{ do not contain an } (s, t)\text{-cut}\}. \end{aligned}$$

The second is the *connectedness measure*: given vertex s in G,

$$\begin{aligned} g(G, s; p) &= \Pr\{\text{there is a path of operative edges from } s \text{ to every other vertex in } G\} \\ &= \Pr\{\text{the failed edges of } G \text{ do not contain a network cut}\}. \end{aligned}$$

These measures are defined for both directed and undirected graphs. If G is undirected, then $f(G, s; p)$ is the probability that the operative edges in G form a connected graph on V, and is independent of the vertex s. The combinatorial significance of these reliability measures can be seen by expanding f and g:

$$f(G, s, t; p) = \sum_{j=0}^{n} f_j p^j (1-p)^{n-j},$$

$$g(G, s; p) = \sum_{j=0}^{n} g_j p^j (1-p)^{n-j},$$

where

f_j = number of sets of edges of cardinality j whose complement admits a path from s to t,
= number of sets of edges of cardinality j that do not contain an (s, t)-cut;
g_j = number of sets of edges of cardinality j whose complement admits a path from s to every vertex in G,
= number of sets of edges of cardinality j that do not contain a network cut with respect to s.

The use of this form of the polynomial might seem slightly unnatural since coefficients are defined in terms of complements. However, it is consistent with the independence system interpretation of the reliability analysis problem used in other papers [2], [3]. Thus, the evaluation of f and g depend on the counting problems associated with (s, t)-cuts and network cuts in a way that will be shown precisely below.

We explore the computational complexity of counting and reliability problems in the manner proposed by Valiant [12]. The study of the complexity of feasibility and optimization problems has been pursued in the setting of recognition problems [5]. An important class is NP, which consists of those recognition problems accepted by a nondeterministic Turing machine of polynomial time complexity. The "hardest" problems in NP are called NP-complete; it is generally considered unlikely that polynomial algorithms exist for solving problems in this class. Valiant defines #P to be the set of integer-valued functions that can be computed by counting the number

of accepting computations of some nondeterministic Turing machine of polynor
time complexity. We extend Valiant's definition slightly to include rational and mult
valued functions that can be evaluated using functions of the above type. We say
a function f is *polynomially reducible* to a function $g (f \propto g)$ is there exists an algori
which, for any input z, evaluates $f(z)$ with a number of elementary operations
evaluations of g that is polynomial in the length of z. A function f is called #P-*comp*
if (a) f is in #P and (b) every function g in #P can be reduced to f by a polynom
time reduction. The classes #P and #P-complete provide a natural setting for study
the complexity of counting and reliability problems. We remark that the counti
problem associated with a given recognition problem is at least as hard as
recognition problem. In particular, the counting versions of NP-complete proble
are NP-hard, i.e. at least as hard as NP-complete problems. To illustrate this poi
note that a polynomial algorithm to determine the *number* of Hamiltonian circuits
a graph would immediately give a polynomial algorithm to determine if a gra
contained at least one Hamiltonian circuit. In fact, the counting versions of mo
NP-complete problems can be easily shown to be #P-complete. See [5] for a detail
treatment of NP-completeness and its relationship to #P-completeness.

With these definitions in mind we state our main result:

THEOREM. *The following functions are* #P-complete:

1. BIPARTITE VERTEX COVER
Input: *bipartite graph* $G = (V, E)$
Output: $|\{S \subseteq V$: *for each* $e = (u, w) \in E$, $u \in S$ *or* $w \in S\}|$;
2. BIPARTITE INDEPENDENT SET
Input: *bipartite graph* $G = (V, E)$
Output: $|\{S \subseteq V$: *for all* $u, w \in S$, $e = (u, w) \notin E\}|$;
3. ANTICHAIN
Input: *partial order* $(X, \leqq)$
Output: $|\{S \subseteq X$: *there are no* $x, y \in S$ *with* $x \leqq y\}|$;
4. MINIMUM CARDINALITY BIPARTITE VERTEX COVER (MAXIMUM CARDINALITY BIPARTITE INDEPENDENT SET, MAXIMUM CARDINALITY ANTICHAIN, RESPECTIVELY)
Input: *same as* 1 (2, 3, *resp.*)
Output: *the number of minimum cardinality* (*maximum cardinality, resp.*) *elements of the output set*;
5. BIPARTITE 2-SAT WITH NO NEGATIONS
Input: *Boolean expression B in the variables* $x_1, \cdots, x_k, y_1, \cdots, y_l$ *of the form* $B = (x_{i_1} \vee y_{j_1}) \wedge \cdots \wedge (x_{i_m} \vee y_{j_m})$
Output: $|\{x_1, \cdots, x_k, y_1, \cdots, y_l)$ *that satisfy* $B\}|$;
6. MINIMUM CARDINALITY (s, t)-CUT
Input: *graph* $G = (V, E)$, $s, t \in V$
Output: $|\{C \subseteq E$: *C is a minimum cardinality* (s, t)-*cut in* $G\}|$;
7. MINIMUM CARDINALITY DIRECTED NETWORK CUT
Input: *directed graph* $G = (V, E)$, $s \in V$
Output: $|\{C \subseteq E$: *C is a minimum cardinality network cut with respect to* $s\}|$;
8. CONNECTEDNESS RELIABILITY
Input: *graph* $G = (V, E)$, $s \in V$, *rational* p, $0 \leqq p \leqq 1$
Output: $g(G, s; p)$;
9. CONNECTEDNESS RELIABILITY ε-APPROXIMATION
Input: *graph* $G = (V, E)$, $s \in V$, $\varepsilon \leqq 0$, *rational* p, $0 \leqq p \leqq 1$
Output: *rational* r *with* $r - \varepsilon < g(G, s; p) < r + \varepsilon$;

10. (s, t) CONNECTEDNESS RELIABILITY ε-APPROXIMATION

Input: *graph* $G=(V, E)$, $s, t \in V$, $\varepsilon>0$, *rational* p, $0 \leqq p \leqq 1$

Output: *rational* r *with* $r-\varepsilon<f(G, s, t; p)<r+\varepsilon$;

Before going on to the proof of the theorem, we illustrate how our results fit in with previous results concerning reliability and important related counting problems. Computation of the functions f and g are considered the two most important and well-studied network reliability problems. The theorem settles the complexity of computing g exactly and the ε-approximation problem for f and g. In terms of computing or approximating f, two important quantities are the number of minimum cardinality (s, t)-cuts and the number of minimum cardinality (s, t)-paths. These correspond, respectively, to the first $f_i<\binom{n}{i}$ and the last $f_i>0$. The two corresponding quantities for g are the number of minimum cardinality network cuts and the number of minimum cardinality connected sets, i.e. spanning trees, and these correspond, respectively, to the first $g_i<\binom{n}{i}$ and the last $g_i>0$. Table 1 describes the known complexity results for all of these problems. It uses the generic term pathsets to refer to both spanning trees and (s, t)-paths and cutsets refer to both (s, t)-cuts and network cuts. Columns 1 and 2 refer to the problems of determining the number of minimum cardinality pathsets and cutsets respectively, column 3 to the problem of determining the polynomial f or g, and column 4 to the approximation problem defined in parts 9 and 10 of the theorem.

TABLE 1

	Min. card. pathset	Min. card. cutset	Rel. poly.	Rel. approx.
undirected and directed two-terminal (f)	*[3]	!TH	![12]	!TH
undirected network (g)	*[10]†	*[3]	!TH§	!TH
directed network (g)	*[10]†	!TH	!TH§¶	!TH

Either the appropriate reference is given or TH which indicates the result is contained in the theorems given in this paper; * implies polynomial; ! implies #P-complete.

† Reference [10] reduces the problem to computing the determinant of a matrix. It is (now) well known that determinants can be computed in polynomial time.

§ These results have recently been proven independently by Jerrum [9].

¶ This result has recently been proven independently by Hagstrom [6].

2. Proof of the theorem. The format for establishing a function f as #P-complete is as follows. We first establish that f is in #P by showing that, for any input z, there exists a polynomial algorithm for recognizing structures associated with the input z whose number is $f(z)$. In the context of the functions given in the theorem this is a trivial matter, since virtually all the functions count easily recognizable objects in the graph G associated with the input z. To show that f is #P-complete, we start with a known #P-complete function g, and show that there exists an algorithm which, for any z, evaluates $g(z)$ using a polynomial number of evaluations of f. In many cases this simply involves altering the input z (here the graph G) in polynomial time to a new input z' (here a new graph G') for which $g(z)=f(z')$. In some cases, however, we must evaluate f for a number of inputs $z_1, \cdots, z_m$, that number being polynomial in the size of z. We then relate the values $f(z_i)$, $i=1, \cdots, n$ to the value of g by equations of the form

$$v_i=f(z_i)=\sum_{j=1}^{k} a_{ij} b_j, \quad i=1, \cdots, k \tag{1}$$

where the a_{ij} are known and $g(z)$ is some simple function of the b_j. If we can sh that the $k \times k$ matrix of the coefficients a_{ij} for (1) is nonsingular, we can perforn evaluations of f, and then solve the linear system to obtain the values of b_j, and her the value of $g(z)$.

Valiant, in [12], has made use of a special class of matrices to produce the desir nonsingular systems discussed above. A *Vandermonde matrix* is an $(n+1)\times(n+$ matrix of the form

$$\Delta=\begin{pmatrix} 1 & \mu_0 & \mu_0^2 & \cdots & \mu_0^n \\ 1 & \mu_1 & \mu_1^2 & \cdots & \mu_1^n \\ \vdots & & & & \\ 1 & \mu_n & \mu_n^2 & \cdots & \mu_n^n \end{pmatrix}$$

(or its transpose), where $\mu_0, \cdots, \mu_n$ are arbitrary real numbers. A well-known fac about these matrices (see, for example [7, § 5.1]) is $\det \Delta=\prod_{i>j}(\mu_i-\mu_j)$. We hav immediately from the previous discussion:

LEMMA. *Suppose we have* v_i *and* b_i, $i=1, \cdots, n+1$, *related by the equation*

$$v_i=\sum_{j=1}^{n+1} a_{ij} b_j, \quad i=1, \cdots, n+1.$$

Further, suppose that the matrix of coefficient (a_{ij}) *is Vandermonde, with parameters* $\mu_0, \cdots, \mu_n$ *which are distinct. Then, given values for* $v_1, \cdots, v_{n+1}$, *we can obtain the values* $b_1, \cdots, b_{n+1}$ *in time polynomial in* n.

We will make repeated use of this lemma throughout the proof of the theorem.

It is easy to see that the problems 1–10 of the theorem are in #P. To show that they are #P-complete, we establish a sequence of reductions, starting with the following counting problem.

CARDINALITY VERTEX COVER

Input: *graph* $G=(V, E)$, *integer* k

Output: $|\{S \subseteq V$: *S is a vertex cover for G and* $|S|=k\}|$.

This problem is known to be #P-complete (see [5, p. 169]). We also define one intermediate problem for purposes of the proof, namely,

0. VERTEX COVER

Input: *graph* $G=(V, E)$

Output: $|\{S \subseteq V$: *for each* $e=(u, v) \in E$, $u \in S$ *or* $v \in S\}|$.

We now give the reductions.

0. CARDINALITY VERTEX COVER $\propto$ VERTEX COVER. Given $G=$ (V, E), for $l=1, \cdots, m=|V|$, construct graph $G'(l)$ with vertex set $V'(l)=\{v_i': v \in V,$ $i=1, \cdots, l\}$ and edge set $E'(l)=\{(u_i', v_j'): (u, v) \in E, i=1, \cdots, l, j=1, \cdots, l\}$. This construction is illustrated in Fig. 1. Now every cover C' of $G'(l)$ has the property that if $(u, v) \in E$ then $\{u_1', \cdots, u_l'\} \subseteq C'$ or $\{v_1', \cdots, v_l'\} \subseteq C'$. Therefore, for each cover C of G there corresponds a class $\Omega(C)$ of covers of $G'(l)$ with elements of the form $\bigcup_{v \in V} S_v'$, where $S_v'=\{v_1', \cdots, v_l'\}$ if $v \in C$ and $S_v' \subsetneqq\{v_1', \cdots, v_l'\}$ if $v \notin C$. The class $\Omega(C)$ consists of $(2^l-1)^{m-|C|}$ covers, and the classes $\{\Omega(C): C$ a cover of $G\}$ partition

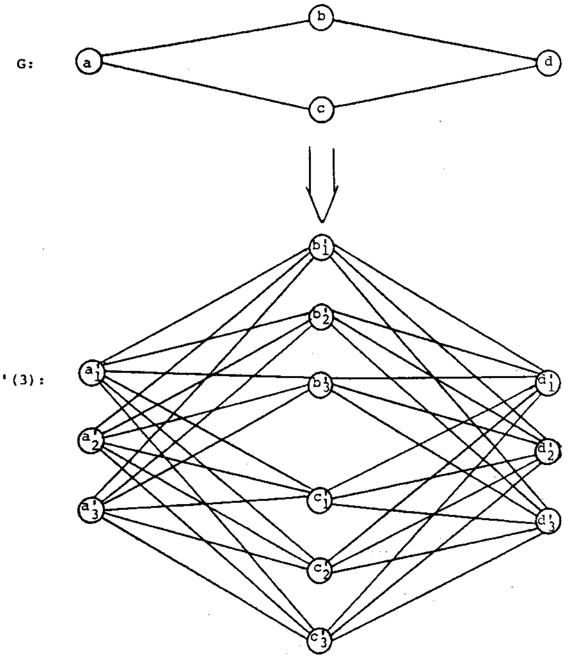

FIG. 1. *Example of transformation used in reduction* 0.

the covers of $G'(l)$. The number of covers of $G'(l)$ is therefore

$$\Gamma(l) = \sum_{i=0}^{m-1} A_i(2^l - 1)^i, \tag{2}$$

where A_i is the number of covers of G of cardinality $m-i$, $i = 0, \cdots, m-1$. Now the $m \times m$ matrix $B = (b_{jl})$ with entries $b_{jl} = (2^l - 1)^{j-1}$ $j = 1, \cdots, m$, $l = 1, \cdots, m$, is Vandermonde with $\mu_l = 2^l - 1$ distinct for $l = 1, \cdots, m$. Therefore, by the lemma we can solve (2) to obtain each A_i, and hence solve the cardinality vertex cover problem.

1. VERTEX COVER $\propto$ BIPARTITE VERTEX COVER. Given $G = (V, E)$, for $l = 0, \cdots, N = \binom{m+2}{2} - 1$ construct bipartite graph $G'(l)$ by replacing each edge (u, v) in G by the subgraph shown in Fig. 2. (Note that when $l = 0$, the graph $\Gamma(l)$ has no edges at all.) This subgraph has the property that the number of vertex covers containing neither u nor v is 2^l, the number of covers containing a particular one of u or v is 3^l and the number of covers containing both u and v is 5^l. Thus, the number of covers of $G'(l)$ is

$$\Gamma'(l) = \sum_{\substack{i+j+k=n \\ i,j,k \geqq 0}} A_{ijk}(2^l)^i(3^l)^j(5^l)^k = \sum_{\substack{i+j+k=n \\ i,j,k \geqq 0}} A_{ijk}(2^i3^j5^k)^l, \tag{3}$$

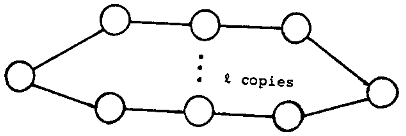

FIG. 2

where A_{ijk} is the number of sets S of vertices in G for which i edges of G have neit
vertex in S, j edges have exactly one vertex in S, and k edges have both vertices
S. The $N \times N$ matrix $B = (b_{ql})$ defined

$$b_{ql} = (2^{i_q}3^{j_q}5^{k_q})^l, \qquad q = 1, \cdots, N, \qquad l = 0, \cdots, N-1,$$

where (i_q, j_q, k_q) are all triples summing to n, is Vandermonde. Further, $\mu_q = 2^{i_q}3^{j_q}5^{k_q}$
$2^{i_r}3^{j_r}5^{k_r} = \mu_r$ if and only if $i_q = i_r$, $j_q = j_r$, and $k_q = k_r$. Therefore, the μ_q are distinct a
by the lemma we can solve (3) to obtain each A_{ijk}, for $i+j+k = n$, $i \geqq 0$, $j \geqq 0$, $k \geqq$
In particular, we can obtain

$$A = \sum_{\substack{j+k=n \\ j,k \geqq 0}} A_{0jk},$$

which is the number of sets of vertices of G for which no edge of G is uncovere
that is, the number of covers of G.

2. BIPARTITE VERTEX COVER $\propto$ BIPARTITE INDEPENDENT SET
Given $G = (V, E)$ we note that $C \subseteq V$ is a cover for G if and only if $V - C$ is a
independent set in G. The reduction follows.

3. BIPARTITE INDEPENDENT SET $\propto$ ANTICHAIN. Given bipartit
graph $G = (V, E)$ with $V = V_1 \cup V_2$ and $E \subseteq V_1 X V_2$, define partial order $(X, \leqq)$ wit
$X = V$ and order defined for $x \neq y \in X : x \leqq y$ if and only if $x \in V_1, y \in V_2$, and $(x, y) \in E$
$(X, \leqq)$ is trivially transitive and antisymmetric. Further, a set $S \subseteq X$ is an antichai
in $(X, \leqq)$ if and only if it is independent in G. The reduction follows.

4. BIPARTITE VERTEX COVER $\propto$ MINIMUM CARDINALITY
BIPARTITE VERTEX COVER (MAXIMUM CARDINALITY BIPARTITE
INDEPENDENT SET, MAXIMUM CARDINALITY ANTICHAIN, RESPEC-
TIVELY). Given bipartite graph $G = (V, E)$, construct bipartite graph $G' = (V', E')$
by adding vertices $\{v' : v \in V\}$ to V and pendant edges $M = \{(v, v') : v \in V\}$ to E. Now
since M consists of m disjoint edges that cover all vertices of G (a perfect matching),
it follows that a minimum cardinality vertex cover of G' is of cardinality m. Further-
more, there is a 1-1 correspondence between vertex covers of G and minimum
cardinality vertex covers of G' obtained by associating with cover C of G the cardinality
m cover

$$C' = \{v : v \in C\} \cup \{v' : v \notin C\}.$$

In view of the discussion in reductions 2 and 3, it follows easily that the bipartite vertex cover problem reduces to any of the three given minimum or maximum cardinality problems.

5. BIPARTITE VERTEX COVER $\propto$ BIPARTITE 2-SAT WITH NO NEGATIONS. Given bipartite graph $G = (V, E)$ with $V = V_1 \cup V_2 V_1 = \{u_1, \cdots, u_k\}$,

$V_2=\{v_1,\cdots,v_l\}$ define Boolean expression in $x_1,\cdots,x_k,y_1,\cdots,y_l$ by

$$f(x_1,\cdots,x_k,y_1,\cdots,y_l)=\bigwedge_{e=(u_i,y_j)\in E}(x_i\vee y_j).$$

Then $f(x_1,\cdots,x_k,y_1,\cdots,y_l)$ is true if and only if $\{u_i:x_i=T\}\cup\{v_j:y_j=T\}$ forms a cover of G. The reduction follows.

6. BIPARTITE INDEPENDENT SET $\propto$ MINIMUM CARDINALITY (s,t)-CUT. Given a bipartite graph $G=(V,E)$, $V=V_1\cup V_2$, $E\subseteq V_1\times V_2$, construct the graph G' with vertices $V\cup\{s,t\}$ and edges consisting of E along with sets M'_v of multiple edges of the type (s,v), $v\in V_1$ or (v,t), $v\in V_2$ with multiplicity equal to the degree of v in G. An example of this construction is given in Fig. 3. Now a minimum cardinality (s,t)-cut in G' is of cardinality $|E|$, since (a) E is an (s,t)-cut and (b) an (s,t)-flow of size $|E|$ can be obtained by directing all edges from s to t and giving each a flow of 1. It is clear that if a minimum cardinality (s,t)-cut C' of G' contains one edge of a set M'_v then it must contain every edge in M'_v. Further, if the edges (s,v) and (w,t) are in C', then (v,w) cannot be an edge, since we can obtain a cut with one less edge by replacing M'_v and M'_w with all edges in E adjacent to either v or w. Thus, the sets M'_v of C' have ends in G which are independent in G and the remaining edges in C' must be all those edges in E which do not have a vertex in common with these sets. Conversely, any set of edges of this type must be a minimum cardinality (s,t)-cut in G'. Thus, there is a one to one correspondence between minimum cardinality (s,t)-cuts in G' and independent sets in G. The reduction is now complete. Note that this reduction applies in both the directed and undirected cases.

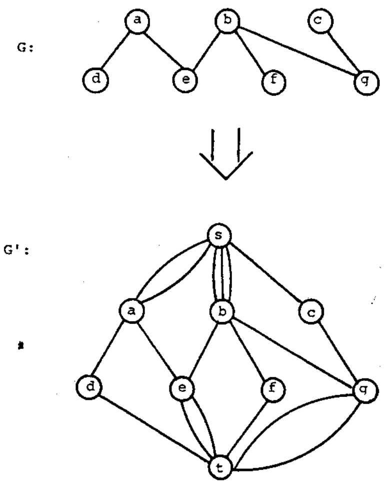

FIG. 3. *Example of transformation used in reduction* 6.

The use of multiple edges could have been avoided but we omit the argument for t sake of simplicity.

7. DIRECTED MINIMUM CARDINALITY (s,t)-CUT $\propto$ MINIMU CARDINALITY DIRECTED NETWORK CUT. Given directed graph G (V,E), $s,t\in V$, let k be the cardinality of a minimum cardinality (s,t)-cut. (It is we known that k can be calculated in polynomial time using a network flow algorithm Construct directed graph G' from G by adding multiple edges of the form (t,v) wi multiplicity $k+1$ for each $v\in V-\{s,t\}$. Fig. 4 illustrates this transformation. Now an minimum cardinality (s,t)-cut in G remains a network cut in G' since all of the added edges point out of t. Thus, the size of a minimum cardinality network cut in G' is at most k. But since removal of any set S of at most k edges from E' must leave at least one edge from t to every vertex $x\neq s$ in V, then S is a network cut in G' if and only if S is an (s,t)-cut in G. Therefore, the minimum cardinality network cuts for G' are of cardinality k, and they consist precisely of sets of edges which are (s,t)-cuts for G. This completes the reduction. As in the previous argument, the use of multiple edges could have been avoided.

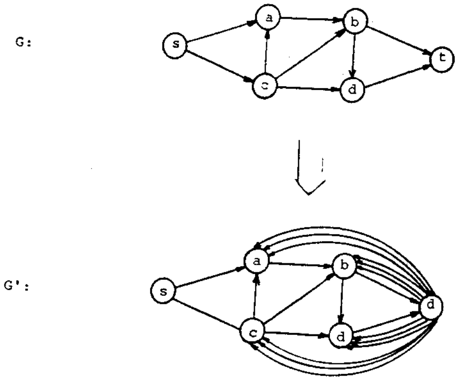

FIG. 4. *Example of transformation used in reduction* 7.

8A. MINIMUM CARDINALITY DIRECTED NETWORK $\propto$ DIRECTED CONNECTEDNESS RELIABILITY. Given $G=(V,E)$, we write, as in § 1,

$$g(G,s;p)=\sum_{j=0}^{n}g_jp^j(1-p)^{n-j}=(1-p)^n\sum_{j=0}^{n}g_j\left(\frac{p}{1-p}\right)^j, \tag{4}$$

where g_j is the number of sets of edges of cardinality j whose complement admits a path from s to every other vertex in G. Thus, $\bar{g}=\binom{n}{j}-g_j$ is the number of sets of edges of cardinality j that contain a directed network cut. Further, the matrix $B=(b_{ij})$ with $b_{ij}=(p_i/(1-p_i))^j$ for $i=0,\cdots,m$, $j=0,\cdots,n$ is Vandermonde for any choice $0<p_0\cdots<p_n<1$. Therefore, by evaluating $g(G.s;p)/(1-p)^n$ for $i=0,\cdots,n$, and solving (4) we can obtain g_i, and hence $\bar{g}_i$ for $j=0,\cdots,n$. The value of the first nonzero $\bar{g}_i$ then solves the minimum cardinality directed network cut problem.

8B. MINIMUM CARDINALITY UNIDIRECTED (s,t)-CUT $\propto$ UNIDIRECTED CONNECTEDNESS RELIABILITY. Given undirected graph G, vertices s, t, write the network reliability polynomial of G with respect to s as above

$$g(G,s;p)=(1-p)^n\sum_{i=0}^{n} g_i\left(\frac{p}{1-p}\right)^i,$$

where g_i is the number of sets of cardinality i whose complement admits a path from every vertex to s. Consider now the graph G' obtained from G by replacing the vertices s and t with the vertex v'_{st} in every edge in which either appears. The network reliability polynomial of G' with respect to v'_{st} is

$$g(G', v'_{st};p)=\sum_{i=0}^{n} g'_i p^i(1-p)^{n-i}=(1-p)^n\sum_{i=0}^{n} g'_i\left(\frac{p}{1-p}\right)^{n-i}.$$

Now g'_i is the number of sets of edges in G' of cardinality i whose complement admits a path from every vertex of G' to v'_{st}, or equivalently, the number of sets of edges in G of cardinality i whose complement admits a path from every vertex of G to either s or t. Therefore, g'_i-g_i is the number of sets of edges in G of cardinality i whose complement admits a path from every vertex to s or t but does not admit a path from every vertex to *both* s and t. Such a set in particular contains an (s,t)-cut. Let k be the cardinality of a minimum cardinality (s,t)-cut. Then the complement of any set of k edges that contains an (s,t)-cut must allow a path from every vertex to either s or t (otherwise, an edge could be added to the component containing a vertex not connected to either s or t and still not allow a path from s or t). Thus g'_k-g_k is the number of minimum cardinality (s,t)-cuts in G. As in problem 8A, by evaluating $g(G,s;p_i)$ and $g(G',v'_{st};p_i)$ for $0<p_0<\cdots<p_n$, we can obtain g_i and g'_i for $i=0,\cdots,n$, and in particular, the value g'_k-g_k. This completes the reduction.

9. MINIMUM CARDINALITY NETWORK CUT $\propto$ CONNECTEDNESS RELIABILITY APPROXIMATION. Suppose we are given $G=(V,E)$ and $s\in V$. We produce this reduction by showing how to compute the g_i successively for $i=0,1,\cdots$, using as a subroutine an algorithm for the connectedness reliability approximation problem. Suppose we have computed g_i for $i=0,1,\cdots,k-1$; define

$$\alpha=\sum_{i=0}^{k-1} g_ip^i(1-p)^{n-i};$$

then for $0<p<1$ we have

$$\begin{aligned}g(G,s;p)-\alpha&=\sum_{i=k}^{n} g_ip^i(1-p)^{n-i}\\&=p^k(1-p)^{n-k}\left[g_k+\frac{p}{1-p}\sum_{i=k+1}^{n} g_i\left(\frac{p}{1-p}\right)^{i-k-1}\right].\end{aligned}$$

Using the fact that $0\leqq g_i\leqq\binom{n}{i}$ for $i=k+1,\cdots,n$, we obtain the inequalities

$$\frac{g(G,s;p)-\alpha}{p^k(1-p)^{n-k}}\geqq g_k$$

and

$$\begin{aligned}&\frac{g(G,s;p)-\alpha}{p^k(1-p)^{n-k}}\\&\quad\leqq g_k+\frac{p}{1-p}\sum_{i=k+1}^{n}\binom{n}{i}\left(\frac{p}{1-p}\right)^{i-k-1}\\&\quad=g_k+\frac{p}{1-p}\sum_{i=k+1}^{n}\left[\frac{n!}{(n-k-1)!}\right]\left[\frac{(i-k-1)!}{i!}\right]\left[\binom{n-k-1}{i-k-1}\left(\frac{p}{1-p}\right)^{i-k-1}\right]\\&\quad\leqq g_k+\frac{p}{1-p}\left[\frac{n!}{(n-k-1)!}\right]\left[\frac{1}{(k+1)!}\right]\left[\frac{1}{(1-p)^{n-k-1}}\right]\\&\quad=g_k+\binom{n}{k+1}\frac{p}{(1-p)^{n-k}}.\end{aligned}$$

Now if r is an ε-approximation to g, it follows for $0<p<1$ that

$$\begin{aligned}g_k\leqq\frac{r+\varepsilon-\alpha}{p^k(1-p)^{n-k}}=\frac{(r-\varepsilon)-\alpha+2\varepsilon}{p^k(1-p)^{n-k}}&\leqq\frac{g(G,s;p)-\alpha+2\varepsilon}{p^k(1-p)^{n-k}}\\&\leqq g_k+\binom{n}{k+1}\frac{p}{(1-p)^{n-k}}+\frac{2\varepsilon}{p^k(1-p)^{n-k}}\\&=g_k+\frac{1}{(1-p)^{n-k}}\left[\binom{n}{k+1}p+\frac{2\varepsilon}{p^k}\right],\end{aligned}$$

so that, if we choose

$$p=\min\left\{1-2^{-1/(n-k)},\frac{1}{2}\binom{n}{k+1}^{-1}\right\}$$

and $\varepsilon=p^k/4$, then

$$\frac{1}{(1-p)^{n-k}}\left[\binom{n}{k+1}p+\frac{2\varepsilon}{p^k}\right]<\frac{1}{1/2}\left[\binom{n}{k+1}\frac{1}{2}\binom{n}{k+1}^{-1}+\frac{p^k/2}{p^k}\right]=2\left[\frac{1}{2}+\frac{1}{2}\right]=1.$$

Hence,

$$g_k=\left\lfloor\frac{r+\varepsilon-\alpha}{p^k(1-p)^k}\right\rfloor.$$

The proof is now complete.

10. MINIMUM CARDINALITY (s,t)-CUT $\propto$ (s,t)-CONNECTEDNES RELIABILITY APPROXIMATION. The reduction here is identical to that i problem 9. This completes the proof of the theorem.

3. Further discussion. We remark that problems 9 and 10 easily show th #P-completeness of the *α-approximation problem* (see [11], called the *point estimat* problem in [1] for the functions g and f. This problem is: given $\alpha<1$, $0\leqq p\leqq 1$, fin a number r such that $\alpha r<g(G,s;p)$ (respectively $f(G,s,t;p))<r/\alpha$. We should not that a seemingly more difficult unsolved problem involves the case where α (or ε) i constant, i.e. is not allowed to vary as part of the input list.

We complete our discussion by considering the complexity of certain reliability and counting problems for two special classes of graphs. One class is that of directed acyclic graphs, that is, graphs that have no closed (directed) paths. For these graphs the minimum cardinality (s, t)-cut problem (6) still remains #P-complete since the network constructed in the proof of the theorem is acyclic; hence, the (s, t)-connectedness problem (10) for acyclic graphs remains #P-complete. The directed network cut problem (7), however, is polynomial, and, in fact, the connectedness reliability problems (8 and 9) are also polynomial (see [3]). The second class of graphs is that of planar graphs (directed and undirected). Here, both the minimum cardinality network cut problem and the minimum cardinality (s, t)-cut problem are polynomial (see also [3]). The complexity of the reliability problems, however, are open questions. Table 2 summarizes known results for these classes of graphs.

TABLE 2

	Min. card. pathset	Min. card cutset	Rel. poly.	Rel. approx.
directed acyclic two terminal	*[3]	!TH	!TH	!TH
directed acyclic network	*[3]	*[3]	*[3]	*[3]
undirected and directed planar two terminal	*[3]	*[3]	?	?
undirected and directed planar network	*[10]	*[3]	?	?

The table entries have the same interpretation as those in Table 1.

REFERENCES

[1] M. O. BALL, *The complexity of network reliability computations*, Networks, 10 (1980), pp. 153–165.

[2] M. O. BALL AND J. S. PROVAN, *Bounds on the reliability polynomial for shellable independence systems*, SIAM J. Alg. Discr. Meth., 3 (1982), pp. 166–181.

[3] ———, *Calculating bounds on reachability and connectedness in stochastic networks*, Networks, 13 (1983), pp. 253–278.

[4] Z. GALIL, *On some direct encodings of nondeterministic Turing machines operating in polynomial time into P-complete problems*, SIGACT News, 6, 1 (1974), pp. 19–24.

[5] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

[6] J. N. HAGSTROM, *Computing rooted communication reliability is #P-complete*, unpublished manuscript, 1981.

[7] G. H. HARDY, J. E. LITTLEWOOD AND G. POLYA, *Inequalities*, Cambridge Univ. Press, Cambridge, 1952.

[8] A. S. HOUSEHOLDER, *Principles of Numerical Analysis*, McGraw-Hill, New York, 1953.

[9] M. JERRUM, *On the complexity of evaluating multivariate polynomials*, Ph.D. thesis, Tech. Rep. CST-11-81, Dept. Computer Science, Univ. Edinburgh, 1981.

[10] G. KIRCHHOFF, *Über die Auflosung der Gleichungen, auf welche man sei der Untersuchung der linearen Verteilung Galvanischer Strome gerfuhrt wird*, Poggendorf's Ann. Phy. Chem., 72 (1847), pp. 497–508; English translation, IRE Trans. Circuit Theory, 5 (1958), pp. 4–8.

[11] A. ROSENTHAL, *Computing the reliability of complex networks*, SIAM J. Applied Math, 32 (1977), pp. 384–393.

[12] L. G. VALIANT, *The complexity of enumeration and reliability problems*, this Journal, 8 (1979), pp. 410–421.

SIAM J. COMPUT.
Vol. 12, No. 4, November 1983

0097-5397/83/1204-0014 $01.25/0

ASYMPTOTIC EXPANSIONS OF MOMENTS OF THE WAITING TIME IN A SHARED-PROCESSOR OF AN INTERACTIVE SYSTEM*

DEBASIS MITRA† AND J. A. MORRISON†

Abstract. An interactive computer system's service is characterized by the random waiting (or response) time perceived by users. This paper presents a novel solution to the problem of efficiently computing the second moment of the *waiting time* of a class of *large* interactive systems. The physical system consists of a bank of terminals, each of which asynchronously alternates between "thinking" and waiting for service from a CPU which operates under the processor-sharing discipline. The problem of obtaining higher order waiting time moments is quite different from that of obtaining CPU queue statistics. Only the first moment of the waiting time is obtainable from the CPU queue statistics.

The technique for arriving at the second moment of the waiting time consists of developing an asymptotic expansion in inverse powers of the number of terminals. Hence, as the system grows, quite fortuitously, fewer terms of the series require computation to achieve the desired accuracy. Beside its numerical advantages, the results give new insight since the leading terms of the series, which contain most of the information, are obtained explicitly. The novelty also rests on the fact that instead of solving matrix equations, the problem is turned into one of solving a second order differential equation. A simple two-dimensional recursion yields all the terms of the asymptotic expansion.

Key words. queueing networks, queueing theory, asymptotic expansions, waiting time moments

1. Introduction.

An interactive computer system's service is characterized by the random waiting (or response) time perceived by users of the system. This paper presents a novel approach to the problem of efficiently computing the first and second moments of the waiting time for *large* interactive systems. The physical system, see Fig. 1, consists of a bank of user terminals in series with a CPU which feeds back to the terminals. Each user spends alternating time periods in the "think" mode and the "waiting" mode; in the former, the user takes an independent amount of time to generate jobs with random service time requirements, while in the waiting mode the job, now transferred to the CPU, contends with other jobs for service. On completion of service the job returns to the terminal and a new cycle resumes.

The waiting time distribution for such a model has been considered in [1] and the moments have been given there in terms of the solution of a matrix equation. These equations have dimension $(N+1)$, where $(N+1)$ is the number of user terminals. Practical interest is focused on large systems, i.e. large N, and in this case the equations pose a computational challenge which is compounded by the equation's worsening conditioning with increasing system usage. Also, insight into the nature of solutions is less readily forthcoming. In this paper we give a quite novel technique for arriving at the second moment $E[W^2]$. Given here is an asymptotic expansion for $E[W^2]$ in inverse powers of N, so that the larger N is, fewer terms of the series require computation to achieve the desired degree of accuracy. Also, it is possible to give explicitly and simply the leading terms which contain most of the information and are most amenable to interpretation.

The novelty of the technique also rests on the fact that, instead of inverting matrices, we transform the problem into a differential equation for the generating

*Received by the editors September 1, 1982, and in revised form December 21, 1982. This paper was typeset by Carmela Patuto at Bell Laboratories, Murray Hill, New Jersey, using the troff program running under the Unix™ operating system. Final copy was produced on April 4, 1983.

†Bell Laboratories, Murray Hill, New Jersey 07974.