

**COMPLEXITY OF NETWORK RELIABILITY AND  
OPTIMAL DATABASE PLACEMENT PROBLEMS**

**Donald B. Johnson  
Larry Raab**

**Technical Report PCS-TR91-167**

# Complexity of Network Reliability and Optimal Database Placement Problems

Donald B. Johnson\*      Larry Raab†  
Dartmouth College‡

October 25, 1991

## Abstract

A fundamental problem of distributed database design in an existing network where components can fail is finding an optimal location at which to place the database in a centralized system or copies of each data item in a decentralized or replicated system. In this paper it is proved for the first time exactly how hard this placement problem is under the measure of data availability. Specifically, we show that the optimal placement problem for availability is  $\#P$ -complete, a measure of intractability at least as severe as  $NP$ -completeness. Given the anticipated computational difficulty of finding an exact solution, we go on to describe an effective, practical method for approximating the optimal copy placement. To obtain these results, we model the environment in which a distributed database operates by a *probabilistic graph*, which is a set of fully-reliable vertices representing sites, and a set of edges representing communication links, each operational with a rational probability. We prove that finding the optimal copy placement in a probabilistic graph is  $\#P$ -complete by giving a sequence of reductions from  $\#Satisfiability$ . We generalize this result to networks in which each site and each link has an independent, rational operational probability and to networks in which all the sites or all the links have a fixed, uniform operational probabilities.

## 1 Introduction

Determining the optimal placement of a resource, be it a file, database, or data object, is one of the most well-studied problems in computer science. Research into the “file assignment problem”, or FAP as it is now known[3, 8], dates back to Chu in 1969[4] and even earlier when viewed as the single commodity warehouse problem[16]. This paper differs from all others of which we are aware in that the

---

\*e-mail address:djohnson@dartmouth.edu

†e-mail address:raab@dartmouth.edu

‡Department of Mathematics and Computer Science. Hanover, N.H. 03755

measure that we wish to optimize is availability, which is defined as the probability that an arbitrary node in the network is connected to the site containing the file or data object. In addition, we show that this problem is  $\#P$ -complete, not  $NP$ -complete as is frequently shown for other location problems, and is therefore *at least* as “hard” as  $NP$ -complete problems.

In [7], Dowdy and Foster present a survey of research dealing with FAP, including a description of fourteen models and a list of twenty-one others. These and other more recent models with approximate solutions are discussed in [10]. Although the models vary considerably, they all attempt to minimize some cost measure (such as storage or communication cost) or maximize throughput. Although some of these models include an availability constraint, they neither maximize availability nor define it as above.

Our interest in this availability measure is motivated by our work with database replica control protocols.[12, 15, 13] These protocols attempt to increase the accessibility of a data object by replicating that object throughout the network. Our work has shown that, given the database consistency constraints, there is a non-trivial bound on the benefits of replication over an *optimally located* non-replicated data object.[15] Thus, it is natural to attempt a complexity characterization and an approximation algorithm for solving this optimal location problem, both of which we present in this paper.

The most general form of the optimal database placement problem is as follows: given a set of sites, communication links, rational reliability probabilities on both the sites and links, and a distribution of access requests, find the *optimal* site. A site  $x$  is optimal if and only if placing the data object at site  $x$  maximizes availability. Availability is defined as the probability that an access request submitted according to the access request distribution occurs at a site that can communicate with site  $x$ . A rational reliability  $\frac{p}{q}$  for a site (and similarly for a link) is the steady-state probability that the site is operational. Therefore,  $\frac{p}{q} = \frac{MTTF}{MTTF+MTTR}$ , where  $MTTF$  is the mean time to failure for a site, and  $MTTR$  is the mean time to recovery for a site.

Thus, the optimal location at which to place the sole copy of a data item in a distributed environment is a function of the network topology, the site and link reliabilities, and the access request distribution. We show that since the underlying graph reliability problems are  $\#P$ -complete, so also is this *optimal placement* problem.  $\#P$ -complete implies, among other things, that an efficient (polynomial) solution to this problem can be found only if  $P = NP$ . In this paper we prove that the simplified problem where the sites are infallible, links operate with probability one-half, and the access request distribution is uniform (that is,  $\frac{1}{n}$  of the accesses is submitted to each of the  $n$  sites) is  $\#P$ -complete. We call this the *simplified model* and call the graph representing such a network a *probability graph*. Using the simple technique of restriction[9], we generalize this result to networks in which each site and each link has an independent, rational operational probability, to net-

works with fixed, uniform, rational site or link probabilities, and to arbitrary access request distributions.

Because this problem is computationally difficult, we cannot expect to find an efficient, exact solution. Since the necessity of finding the best possible database location remains, we also give a practical, efficient method of approximating on-line the optimal copy placement in general networks. Furthermore, we describe in section 4 situations in which this method may be preferable to an exact off-line calculation.

We begin by listing each of the problems which we use to prove our  $\#P$ -completeness result. Each of these problems are interesting probability graph problems in their own right. In section 3, we prove that each of these problems are  $\#P$ -complete. We also generalize the main complexity result to include classes of networks with a uniform, fixed link reliability and networks with a uniform, fixed site reliability. This latter class includes such networks as radio broadcast networks[1] and single bus networks like Ethernet. The final section gives an efficient on-line method for approximating the optimal database location based upon the history of the network.

## 2 Problem Definitions

In this section we define each of a sequence of combinatorial problems that we use to prove that finding the optimal location of a single copy is  $\#P$ -complete. The first two problems,  $\#SAT$  and  $CONNECTEDNESS$ , were shown to be  $\#P$ -complete by Cook[6] and Valiant[18], respectively. The other three problems are shown to be  $\#P$ -complete in section 3.

We maximize availability by maximizing  $\mathcal{E}[v]$ , the expected size of the component containing a site  $v$ .  $\frac{\mathcal{E}[v]}{n}$  is the availability achieved on a network with a single copy located at site  $v$ , since, in this simplified model, access requested are submitted uniformly at random, and only requests submitted to sites within the component containing  $v$  will be granted. Therefore site  $v$  is a *optimal location* if and only if  $\mathcal{E}[v] \geq \mathcal{E}[u]$  for all sites  $u$ .

In the questions which follow, by “the expected component size of vertex  $v$ ” we mean the expected size of the component containing  $v$ . Also, if more than one vertex has maximal expected component size,  $OPTLOC$  may return any one of these vertices.

### 1. $\#SAT$ ( $\#SAT$ )

INSTANCE: A logical formula  $F$  in  $n$  variables.

QUESTION: How many different truth assignments which satisfy  $F$  are there to the  $n$  variables?

### 2. $CONNECTEDNESS$ ( $CON$ )

INSTANCE: A probability graph  $G = (V, E)$ , and vertices  $v_1, v_2 \in V$ .

QUESTION: What is the probability that vertices  $v_1$  and  $v_2$  are connected?

3. **EXPECTED SIZE (*EXPSZ*)**

INSTANCE: A probability graph  $G = (V, E)$ , and vertex  $v \in V$ .

QUESTION: What is the expected component size of vertex  $v$ ?

4. **BOUNDED EXPECTED SIZE (*BEXPSZ*)**

INSTANCE: A probability graph  $G = (V, E)$ , vertex  $v \in V$ , and a rational number  $B$ .

QUESTION: Has  $v$  expected component size greater than or equal to  $B$ ?

5. **OPTIMAL LOCATION (*OPTLOC*)**

INSTANCE: A probability graph  $G = (V, E)$ .

QUESTION: Which  $v \in V$  has the largest expected component size?

### 3 Reductions

In this section we either prove or cite proofs for each of the problems defined in the previous section. The first three problems are proved elsewhere and citations are given. The remaining two problems are shown to be  $\#P$ -complete. We include a subsection with two related Lemmas that are used in section 3.3.

#### 3.1 Preliminary Reductions

**Theorem 1:** *#SAT is #P-complete.*

Proof: In [6] Cook proved that *SAT* is *NP*-complete. Valiant defined  $\#P$ -complete in such a way that *SAT* is *NP*-complete implies that *#SAT* is  $\#P$ -complete[18].  $\square$

**Theorem 2:** *CON is #P-complete.*

Proof: A reduction from *#SAT* to *CON* is given by Valiant in [18].  $\square$

**Theorem 3:** *EXPSZ is #P-complete.*

Proof:

A reduction by the authors from *CON* to *EXPSZ* in a more general context is given in [15]. We restrict the proof in this paper to probability graphs.

Let  $G = (V, E)$  be a probability graph, and let  $\mathcal{P}(c(u, w))$  represent the probability that vertices  $u$  and  $w$  are connected.

Then it is not difficult to show that the expected size of the component containing  $v \in V$ ,  $E[v]$ , is equal to  $\sum_{w \in V} \mathcal{P}(c(v, w))$ [15]. Thus *EXPSZ* is in  $\#P$  since we can solve *EXPSZ* with  $|V|$  queries to an *CON* oracle, and *CON* is in  $\#P$ .

We show that *EXPSZ* is  $\#P$ -hard using a Turing reduction from *EXPSZ* to *CON*. We solve *CON* by calculating the expected component size of a vertex in each of two networks.

Let  $G = (V, E)$  and  $u, v \in V$  be an instance of *CON*.

Let  $G' = (V', E')$ , where  $V' = V \cup \{u'\}$  and  $E' = E \cup \{(u, u')\}$ .

$$\begin{aligned} \mathcal{E}_{G'}[v] &= \sum_{w \in V'} \mathcal{P}(c(v, w)) \\ &= \mathcal{P}(c(v, u')) + \sum_{w \in V} \mathcal{P}(c(v, w)) \\ &= \frac{p}{q} \mathcal{P}(c(v, u)) + \sum_{w \in V} \mathcal{P}(c(v, w)) \\ &= \frac{p}{q} \mathcal{P}(c(v, u)) + \mathcal{E}_G[v] \end{aligned}$$

Therefore,  $\mathcal{P}(c(v, u)) = \frac{q}{p} (\mathcal{E}_{G'}[v] - \mathcal{E}_G[v])$ , and calculating  $\mathcal{E}_G[v]$  must be  $\#P$ -complete since calculating  $\mathcal{P}(c(v, u))$  is  $\#P$ -complete.  $\square$

**Theorem 4:** *BEXPSZ* is  $\#P$ -complete.

Proof:

Clearly *BEXPSZ* is in  $\#P$  since we can solve *BEXPSZ* with one query to an *EXPSZ* oracle, and *EXPSZ* is in  $\#P$ .

We show that *BEXPSZ* is  $\#P$ -hard using a Turing reduction from *EXPSZ* to *BEXPSZ*.

Let  $G = (V, E)$  and vertex  $v$  be an instance of the *EXPSZ* problem. That is, we wish to determine  $C$ , the expected size of the component containing vertex  $v$ . Let  $n = |V|$  and  $m = |E|$ . Then there are  $2^m$  possible graph states, and the probability of any one state with  $k$  operational links,  $0 \leq k \leq m$ , is  $(\frac{p}{q})^k (1 - \frac{p}{q})^{m-k}$ . Therefore  $C$ ,  $1 \leq C \leq n$ , is a multiple of  $\frac{1}{q^m}$  and is one of  $nq^m$  possible values. Suppose, then, that we have an oracle which can solve *BEXPSZ*. Then we can use a binary search procedure to query this oracle until we find the exact value of  $C$ . This can be done in  $a \leq \lceil \log(nq^m) \rceil$  queries. Since  $m \leq \frac{n(n-1)}{2}$ ,  $a \leq \lceil \log(nq^{\frac{n(n-1)}{2}}) \rceil = O(n^2)$ .

Since *EXPSZ* is  $\#P$ -complete, and since we can solve *EXPSZ* with a polynomial number of queries to a *BEXPSZ* oracle, it must be that *BEXPSZ* is  $\#P$ -hard.  $\square$

### 3.2 Related Lemmas

We simplify the task of proving that *OPTLOC* is  $\#P$ -complete by establishing two Lemmas. The first Lemma states that the expected component size of vertex  $v$  in graph  $G$ ,  $\mathcal{E}_G[v]$ , is at least as large as the expected component size of any other vertex  $u$ ,  $\mathcal{E}_G[u]$ , times the probability the  $v$  and  $u$  are connected. We denote the probability

that two vertices  $u$  and  $w$  are connected by  $\mathcal{P}(c(u, w))$  and the probability that two vertices  $u$  and  $w$  are connected given that two vertices  $x$  and  $y$  are connected by  $\mathcal{P}(c(u, w) \mid c(x, y))$ .

**Lemma 5.1:** *Let  $G = (V, E)$  be a probability graph and  $u, v \in V$ . Then  $\mathcal{E}_G[v] \geq \mathcal{P}(c(v, u)) \mathcal{E}_G[u]$ .*

Proof:

$$\begin{aligned}
\mathcal{E}_G[v] &= \sum_{w \in V} \mathcal{P}(c(v, w)) \\
&\geq \sum_{w \in V} \mathcal{P}(c(v, u) \text{ and } c(u, w)) \\
&= \mathcal{P}(c(v, u)) \sum_{w \in V} \mathcal{P}(c(u, w) \mid c(v, u)) \\
&\geq \mathcal{P}(c(v, u)) \sum_{w \in V} \mathcal{P}(c(u, w)) \\
&= \mathcal{P}(c(v, u)) \mathcal{E}_G[u]
\end{aligned}$$

□

The following Lemma states that we can make any vertex  $v$  the *optimal* vertex by adding  $\lceil \frac{q}{p}(c+1) \rceil$  vertices, each adjacent to  $v$ .

**Lemma 5.2:** *Let  $G = (V, E)$  be a probability graph and  $v \in V$ . Let  $G' = (V', E')$ , where  $V' = V \cup \{x_i \mid 1 \leq i \leq \lceil \frac{q}{p}(c+1) \rceil\}$ , and  $E' = E \cup \{(v, x_i) \mid 1 \leq i \leq 2c+2\}$ . Then  $v$  is the unique optimal vertex in  $G'$ .*

Proof:

Let  $w \neq v$  be some vertex in  $V$ .

$$\begin{aligned}
\mathcal{E}_{G'}[v] &= \mathcal{E}_G[v] + \frac{p}{q} \lceil \frac{q}{p}(c+1) \rceil && \text{since link reliabilities} = \frac{p}{q} \\
&\geq \mathcal{P}(c(v, w)) \mathcal{E}_G[w] + \frac{p}{q} \lceil \frac{q}{p}(c+1) \rceil && \text{by Lemma 5.1} \\
&> \mathcal{E}_G[w] + \mathcal{P}(c(v, w)) \frac{p}{q} \lceil \frac{q}{p}(c+1) \rceil && \text{since } 0 \leq \mathcal{P}(c(v, w)) \leq 1 \text{ and} \\
&&& \mathcal{E}_G[w] < \frac{p}{q} \lceil \frac{q}{p}(c+1) \rceil \\
&= \mathcal{E}_{G'}[w] && \text{since all paths from } w \text{ to any } x_i \\
&&& \text{pass through } v
\end{aligned}$$

□

### 3.3 OPTLOC Reduction

In this section we use the previous reductions and Lemmas to prove that optimally placing a single copy is  $\#P$ -complete.

**Theorem 5:** *OPTLOC is  $\#P$ -complete.*

Proof:

Clearly *OPTLOC* is in  $\#P$  since we can solve *OPTLOC* using one query to an *EXPSZ* oracle for each  $v \in V$ , and *EXPSZ* is in  $\#P$ .

We show that *OPTLOC* is  $\#P$ -hard using a polynomial time reduction from *BEXPSZ* to *OPTLOC*. That is, we show that we can solve the *BEXPSZ* problem using a machine for solving the *OPTLOC* problem.

Let  $G_v = (V_v, E_v)$ ,  $A$ ,  $v \in V_v$  be an instance of the *BEXPSZ* problem, with  $n = |V_v|$  and  $m = |E_v|$ . We will use *OPTLOC* to determine in polynomial time whether or not  $\mathcal{E}[v] \geq A$ .

We know that  $\mathcal{E}[v] = \sum_{k=0}^m d_k \left(\frac{p}{q}\right)^k (1 - \frac{p}{q})^{m-k}$ , where each  $d_k$  is the sum of the sizes of the component containing site  $v$  in all states with exactly  $k$  operational links. Using the *binomial theorem*, this can be rewritten as  $\sum_{k=0}^m \sum_{j=0}^{m-k} \binom{m-k}{j} d_k (-1)^{m-k-j} \left(\frac{p}{q}\right)^{m-j}$ . If we subtract 1 (since site  $v$  is always operational), and we subtract  $D' \left(\frac{p}{q}\right)$  for as large a integer  $D'$  as possible, we are left with a positive rational number  $D''$  less than  $\frac{p}{q}$ . Thus we can rewrite  $\mathcal{E}[v]$  as  $1 + D' \left(\frac{p}{q}\right) + \sum_{i=2}^m d_i \left(\frac{p}{q}\right)^i$ , where each  $d_i$  is a non-negative integer less than  $q$  (i.e.  $\sum_{i=2}^m d_i \left(\frac{p}{q}\right)^i$  is the base- $\left(\frac{p}{q}\right)$  expansion of  $D''$ ).

We would like to express  $A$  in the same manner, as 1 plus  $A' \left(\frac{p}{q}\right)$  plus a base- $\left(\frac{p}{q}\right)$  expansion of  $A - 1 - A' \left(\frac{p}{q}\right)$ . But this expansion may not terminate in base- $\left(\frac{p}{q}\right)$ . Instead we define  $B$ , a terminating approximation of  $A$ , such that  $\mathcal{E}[v] \geq A$  if and only if  $\mathcal{E}[v] \geq B$ . We form  $B$  simply by truncating  $A$  after the  $m^{\text{th}}$  place and adding  $\left(\frac{p}{q}\right)^m$  if  $B \neq A$ . Thus for some sequence of positive integers  $b_i$  each less than  $q$ ,

$$B = 1 + B' \left(\frac{p}{q}\right) + \sum_{i=2}^m b_i \left(\frac{p}{q}\right)^i$$

We give the reduction below, an explanation following the reduction, and an example in Figure 1.

### Reduction:

Let  $G_u = (V_u, E_u)$  where

$$\begin{aligned} V_u &= \{u\} \cup \{u_i \mid 1 \leq i \leq B'\} \\ E_u &= \{(u, u_i) \mid 1 \leq i \leq B'\} \end{aligned}$$

Let  $G'_v = (V'_v, E'_v)$  where

$$\begin{aligned} V'_v &= V_v \cup \{v_{i,j,k} \mid 1 < i \leq m \text{ and } b_i \geq 1 \text{ and } 1 \leq j \leq b_i \text{ and } 1 \leq k \leq i-1\} \\ E'_v &= E_v \cup \{(v, v_{i,j,1}) \mid 1 < i \leq m \text{ and } b_i \leq 1 \text{ and } 1 \leq j \leq b_i\} \cup \\ &\quad \{(v_{i,j,k}, v_{i,j,k+1}) \mid 1 < i \leq m \text{ and } b_i \geq 1 \text{ and } 1 \leq j \leq b_i \text{ and } 1 \leq k < i-1\} \end{aligned}$$

Let  $G'_u = (V'_u, E'_u)$  where

$$\begin{aligned} V'_u &= V_u \cup \{u_{i,j,k} \mid 1 < i \leq m \text{ and } b_i \geq 1 \text{ and } 1 \leq j \leq b_i \text{ and } 1 \leq k \leq i\} \\ E'_u &= E_u \cup \{(u, u_{i,j,1}) \mid 1 < i \leq m \text{ and } b_i \leq 1 \text{ and } 1 \leq j \leq b_i\} \cup \\ &\quad \{(u_{i,j,k}, u_{i,j,k+1}) \mid 1 < i \leq m \text{ and } b_i \geq 1 \text{ and } 1 \leq j \leq b_i \text{ and } 1 \leq k < i\} \end{aligned}$$

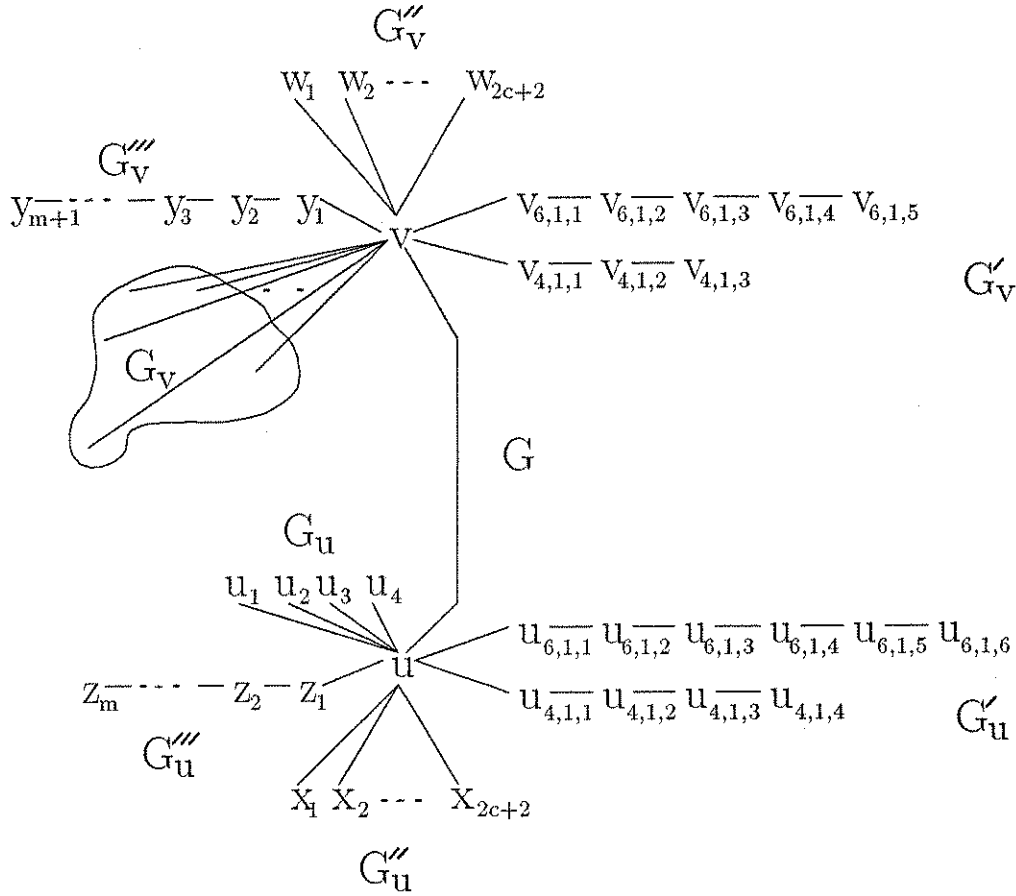


Figure 1: This figure represents the graph  $G$  given an initial graph  $G_v$  with  $\frac{p}{q} = \frac{1}{2}$  and  $B = 3\frac{5}{64}$ . (Therefore,  $B' = 4$ ,  $b_1 = b_2 = b_3 = b_5 = 0$ , and  $b_4 = b_6 = 1$ .) The name of each of the intermediate graphs is given near the portion of  $G$  which was introduced by that intermediate graph. (Note that vertex  $v$  is in  $G_v$ , although this is unclear from the figure.)

Let  $c = \max(|V'_v|, |V'_u|)$

Let  $G''_v = (V''_v, E''_v)$  where

$$\begin{aligned} V''_v &= V'_v \cup \{w_i \mid 1 \leq i \leq \lceil \frac{q}{p}(c+1) \rceil\} \\ E''_v &= E'_v \cup \{(v, w_i) \mid 1 \leq i \leq \lceil \frac{q}{p}(c+1) \rceil\} \end{aligned}$$

Let  $G''_u = (V''_u, E''_u)$  where

$$\begin{aligned} V''_u &= V'_u \cup \{x_i \mid 1 \leq i \leq \lceil \frac{q}{p}(c+1) \rceil\} \\ E''_u &= E'_u \cup \{(u, x_i) \mid 1 \leq i \leq \lceil \frac{q}{p}(c+1) \rceil\} \end{aligned}$$

Let  $G'''_v = (V'''_v, E'''_v)$  where

$$\begin{aligned} V'''_v &= V''_v \cup \{y_i \mid 1 \leq i \leq m+1\} \\ E'''_v &= E''_v \cup \{(y_i, y_{i+1}) \mid 1 \leq i \leq m\} \cup \{(v, y_1)\} \end{aligned}$$

Let  $G'''_u = (V'''_u, E'''_u)$  where

$$\begin{aligned} V'''_u &= V''_u \cup \{z_i \mid 1 \leq i \leq m\} \\ E'''_u &= E''_u \cup \{(z_i, z_{i+1}) \mid 1 \leq i \leq m-1\} \cup \{(u, z_1)\} \end{aligned}$$

Let  $G = (V, E)$  where

$$\begin{aligned} V &= V'''_v \cup V'''_u \\ E &= E'''_v \cup E'''_u \cup \{(u, v)\} \end{aligned}$$

$\mathcal{E}[v] \geq B$  iff  $v$  is the optimal vertex in  $G$ . Since the size of  $V$  is less than  $2q(\frac{q}{p}+1)|E_v|^2 + 2|E_v| + (3\frac{q}{p}+1)|V_v| + 5\frac{q}{p}$ , the size of  $G$  is polynomial in the size of  $G_v$ . Therefore  $OPTLOC$  is  $\#P$ -hard since  $BEXPSZ$  is  $\#P$ -hard.

### Explanation and Correctness:

Clearly,  $\mathcal{E}[v] \geq B$  iff  $\mathcal{E}[v] + d \geq B + d$ , for some rational number  $d$  which we describe later. We, therefore, build a graph  $G'_u$  with optimal vertex  $u$  and  $\mathcal{E}_{G'_u}[u] = B + d$ . At the same time we form  $G'_v$  by augmenting  $G_v$  such that  $\mathcal{E}_{G'_v}[v] = \mathcal{E}_{G_v}[v] + d$ . We then augment both  $G'_u$  and  $G'_v$ , forming  $G''_u$  and  $G''_v$ , respectively, to ensure that either  $u$  or  $v$  or both are the optimal vertices in both  $G''_u$  and  $G''_v$ . We then augment both  $G''_u$  and  $G''_v$ , forming  $G'''_u$  and  $G'''_v$ , respectively, to ensure that  $u$  and  $v$  do not have the same expected size. At this point,  $\mathcal{E}_{G_v}[v] \geq B$  iff  $v$  is the optimal vertex in  $(V'''_u \cup V'''_v, E'''_u \cup E'''_v \cup \{(u, v)\})$ .

Since the expected number of operational links from  $u$  to some  $u_i$  is  $B'(\frac{p}{q})$ ,

$$\mathcal{E}_{G_u}[u] = 1 + B'(\frac{p}{q})$$

Connecting a vertex to a “chain” of  $k$  vertices increases the expected size of the component containing that vertex by  $\sum_{1 \leq j \leq k} \binom{p}{q}^j$ . We form  $G'_v$  from  $G_v$  by adding  $b_k$  chains of length  $k - 1$  for every  $b_k \geq 1$ . Therefore,

$$\mathcal{E}_{G'_v}[v] = \mathcal{E}_{G_v}[v] + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j$$

Likewise, we form  $G'_u$  from  $G_u$  by adding  $b_k$  chains of length  $k$  for every  $b_k \geq 1$ . Adding a chain of length  $k$  to  $G'_u$  and of length  $k - 1$  to  $G'_v$  produces a net increase of  $b_k \left(\frac{p}{q}\right)^k$  in the difference between  $\mathcal{E}_{G'_u}[u]$  and  $\mathcal{E}_{G'_v}[v]$ . Therefore,

$$\begin{aligned} \mathcal{E}_{G'_u}[u] &= 1 + B' \left(\frac{p}{q}\right) + \sum_{b_i \geq 1} \sum_{j=1}^i b_i \left(\frac{p}{q}\right)^j \\ &= 1 + B' \left(\frac{p}{q}\right) + \sum_{b_k \geq 1} b_k \left(\frac{p}{q}\right)^k + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j \\ &= B + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j \end{aligned}$$

Now  $\mathcal{E}_{G_v}[v] \geq B$  iff  $\mathcal{E}_{G'_v}[v] \geq \mathcal{E}_{G'_u}[u]$ . But *OPTLOC* tells which vertex is optimal in the entire graph, not which of  $u$  and  $v$  is better. Therefore we must ensure that either  $u$  or  $v$  is the optimal vertex. Clearly,  $\mathcal{E}_{G'_v}[v] \geq 1$  and  $\mathcal{E}_{G'_v}[t] \leq c$  for all  $t \in V'_v$ . By adding  $\lfloor \frac{q}{p}(c+1) \rfloor$  neighbors to  $v$ , we increase the expected component size of  $v$  by  $\left(\frac{p}{q}\right)^{\lfloor \frac{q}{p}(c+1) \rfloor}$  and ensure, by Lemma 5.2, that  $v$  is the optimal vertex in  $G''_v$ . Likewise for  $u$  in  $G'_u$ . Therefore,

$$\begin{aligned} \mathcal{E}_{G''_v}[v] &= \mathcal{E}_{G'_v}[v] + \frac{p}{q} \left\lfloor \frac{q}{p}(c+1) \right\rfloor \\ &= \mathcal{E}_{G_v}[v] + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j + \frac{p}{q} \left\lfloor \frac{q}{p}(c+1) \right\rfloor \\ \mathcal{E}_{G''_u}[u] &= \mathcal{E}_{G'_u}[u] + \frac{p}{q} \left\lfloor \frac{q}{p}(c+1) \right\rfloor \\ &= B + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j + \frac{p}{q} \left\lfloor \frac{q}{p}(c+1) \right\rfloor \end{aligned}$$

Now  $\mathcal{E}_{G_v}[v] \geq B$  iff  $v$  is the optimal vertex in  $(V''_u \cup V''_v, E''_u \cup E''_v)$ , provided  $\mathcal{E}_{G''_v}[v] \neq \mathcal{E}_{G''_u}[u]$ . If, however,  $\mathcal{E}_{G''_v}[v] = \mathcal{E}_{G''_u}[u]$ , or equivalently  $\mathcal{E}_{G_v}[v] = B$ , we cannot be sure which of  $u$  and  $v$  will be called *optimal*, since *OPTLOC* is indifferent

in this case. Therefore we introduce  $G_v'''$  and  $G_u'''$  such that,

$$\begin{aligned}
\mathcal{E}_{G_v'''}[v] &= \mathcal{E}_{G_v''}[v] + \sum_{j=1}^{m+1} \left(\frac{p}{q}\right)^j \\
&= \mathcal{E}_{G_v}[v] + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j + \frac{p}{q} \left\lceil \frac{q}{p}(c+1) \right\rceil + \sum_{j=1}^m \left(\frac{p}{q}\right)^j + \left(\frac{p}{q}\right)^{m+1} \\
\mathcal{E}_{G_u'''}[u] &= \mathcal{E}_{G_u''}[u] + \sum_{j=1}^m \left(\frac{p}{q}\right)^j \\
&= B + \sum_{b_i \geq 1} \sum_{j=1}^{i-1} b_i \left(\frac{p}{q}\right)^j + \frac{p}{q} \left\lceil \frac{q}{p}(c+1) \right\rceil + \sum_{j=1}^m \left(\frac{p}{q}\right)^j
\end{aligned}$$

Since  $B$  is a multiple of  $\frac{1}{q^m}$ ,  $\mathcal{E}_{G_v}[v] \geq B$  iff  $v$  is the optimal vertex in  $(V_u'' \cup V_v'', E_u'' \cup E_v'')$ .

We form  $G$  by connecting  $G_u'''$  and  $G_v'''$  with an edge from  $u$  to  $v$ . Clearly, “optimality” is preserved.

Therefore  $\mathcal{E}_{G_v}[v] \geq B$  iff  $v$  is the optimal vertex in  $G$ , and  $G$  can be achieved in time polynomial in the size of  $G_v$ . Since *BEXPSZ* is  $\#P$ -hard, *OPTLOC* is also  $\#P$ -hard.  $\square$

### 3.4 Generalizing

Since probability graphs model a subset of the networks with arbitrary, non-uniform link reliabilities and networks with both fallible sites and fallible links, the  $\#P$ -completeness result of the previous section applies to these more complex networks. Also, AboElFotouh and Colbourn have shown the  $\#P$ -completeness of the *CON* problem where vertices, rather than edges, are subject to failure[1]. Using this result, the proof given in this paper can easily be modified to include radio broadcast networks and other networks modeled by graphs with fallible vertices and infallible edges. This also includes single bus networks like Ethernet, where the link reliability can be factored out of the availability equation.

## 4 Approximating Optimal Placement

Although  $\#P$ -complete in general, the determination of the optimal location for the data item is solvable for some systems. Since often a network for an existing database is built incrementally around the database, the current location may be optimal. In addition, the single copy availability can be efficiently determined for regular network topologies[2, 11, 14], such as ring, single-bus, fully-connected, and

for series-parallel networks[5, 17]. Since, for these topologies, the single copy availability can be calculated in polynomial time by calculating the expected component size,  $\mathcal{E}[v] = \sum_{u \in V} \mathcal{P}(c(v, u))$ , for each site in  $V$ , the placement problem can be solved in polynomial time. It may also be possible to efficiently solve the placement problem for networks with fixed, deterministic routing algorithms, since the number of possible paths connecting two sites may not be a function of the size of the network, or the paths may be mutually independent.

Although calculating the expected component size is feasible in some special cases, it is unnecessary and perhaps undesirable to do so in real systems. Instead, each site can record the actual number of access requests submitted to sites within its component, and the site with the largest number can be made the location of the copy. We require that a site record the number of access requests, rather than the number of sites, to accommodate a nonuniform access request distribution. This method is guaranteed to maximize availability because the number of access requests “seen”, that is, submitted within a site’s component, is the same as the number of access requests that would be granted if the data object were located at that site, since communication is symmetric.

If the past network performance and the access request distribution are indicative of future behavior, then this technique leads to optimal copy placement. This method does not require a priori knowledge of the network topology, hardware reliability, or access request distribution, and adjusts automatically to unanticipated changes in any of these system parameters. These characteristics are precisely those necessary for an automated database relocation scheme[15]. Our experience with simulation indicates that this approach will be successful[12].

## 5 Conclusion

We have analyzed a fundamental database problem which seeks the optimal location for database objects. Here optimality is obtained, not by minimizing a cost metric, but by maximizing availability, that is, the probability that an arbitrary access request is submitted to a site which is connected to the site containing the data object. We have shown that this optimal placement problem and a number of related network reliability problems are  $\#P$ -complete, and therefore likely to be computationally tractable only in very small networks. Since the necessity of intelligent database placement in a computer network remains, we presented a method for approximating this location on-line, while the network is performing the useful work for which it was created.

## References

- [1] H. M. AboElFotouh and C. J. Colbourn. Computing the two-terminal reliability for radio broadcast networks. *IEEE Transactions on Reliability*, 1991. to appear.
- [2] Daniel Barbara and Hector Garcia-Molina. The reliability of voting mechanisms. *IEEE Transactions on Computers*, C-36(10):1197–1208, 1987.
- [3] R. G. Casey. Allocation of copies of a file in distributed systems. In *Proceedings AFIPS 1972 SJCC*, pages 617–625. AFIPS Press, 1972.
- [4] W. W. Chu. Optimal file allocation in a multi-computer information system. *IEEE Transactions on Computers*, 18:885–889, 1969.
- [5] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.
- [6] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third ACM Symposium on Theory of Computing*, pages 151–158. ACM, May 1971.
- [7] Lawrence W. Dowdy and Derrell V. Foster. Comparative models of the file assignment problem. *Computing Surveys*, 14(2):287–313, June 1982.
- [8] K. P. Eswaran. Placement of records in a file and file allocation in a computer network. *Information Processing 74, IFIPS*, 1974.
- [9] Michael R. Garey and David S. Johnson. *Computers and Intractability*. W.H. Freeman and Company, New York, 1979.
- [10] Bezalel Gavish and Olivia R. Liu Sheng. Dynamic file migration in distributed computer systems. *Communications of the ACM*, 33(2):177–189, February 1990.
- [11] E. N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30:1141–1144, 1959.
- [12] Donald B. Johnson and Larry Raab. Effects of replication on data availability. *International Journal of Computer Simulation*, 1991. to appear.
- [13] Donald B. Johnson and Larry Raab. Effects of replication on the duration of failure in distributed databases. Technical Report PCS-TR91-200, Dartmouth College, October 1991.
- [14] Donald B. Johnson and Larry Raab. Finding optimal quorum assignments for distributed databases. In *Proceedings of the 1991 International Conference on Parallel Processing*, volume 3, pages 214–218. CRC Press, August 1991.

- [15] Donald B. Johnson and Larry Raab. A tight upper bound on the benefits of replication and consistency control protocols. In *Proceedings of the 10th Symposium on Principles of Database Systems*, pages 75–81. ACM, May 1991.
- [16] C. V. Ramamoorthy and Benjamin W. Wah. The isomorphism of simple file allocation. *ACM Transactions on Computer Systems*, 32(3):221–232, March 1983.
- [17] A. Satyanarayana and R. K. Wood. A linear time algorithm for computing k-terminal reliability in series-parallel networks. *SIAM Journal of Computing*, 14:818–832, 1985.
- [18] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, August 1979.