Finding The Best Edge-Packing for Two-Terminal Reliability is NP-hard

Venkatesh Raman

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1

Abstract. Finding the probability that there is an operational path between two designated vertices in a probabilistic computer network is known to be NP-hard. Edge-packing is an efficient strategy to compute a lower bound on the probability. We prove that finding the set of paths that produces the best edge-packing lower bound is NP-hard.

1. Introduction and motivation.

Computer networks are often modelled as *probabilistic graphs* [1]. The nodes in the graph represent the communication centers in the network, and the edges represent communication links. The nodes are assumed to be perfectly reliable. However, the edges are assumed to fail in a statistically independent manner; we associate with each edge a probability that it operates. Using this model, a number of reliability measures can be formulated.

When two vertices s and t are designated in the network, it is of interest to determine the reliability with which a message originating from vertex s will reach vertex t. This is the *two-terminal reliability* of G, $Rel_2(G)$, more formally defined as the probability that there exists an operational path in G from s to t. The problem of determining $Rel_2(G)$ in a general graph G is known to be $\#P$-complete, although it can be solved in polynomial time for some restricted classes of graphs [1].

Faced with the difficulty of finding the exact two-terminal reliability of a network, one resorts to searching for good lower and upper bounds. A number of efficient strategies to find good bounds on the reliability are given in [1]. These strategies exploit the elegant combinatorial structure of the model. One strategy to compute a lower bound is "edge-packing". Here the idea is to partition the edges of the graph into subgraphs for which the reliability is efficiently computable. Then a lower bound for the two-terminal reliability is the probability that there is an operational path from s to t in at least one of these subgraphs.

In this paper, we prove that finding a set of paths that produces the best edge-packing lower bound is NP-hard. Section 2 describes the decision version of the problem and Section 3 proves it NP-complete. The proof uses gadgets similar to the ones developed in [2]. We conclude the paper with a brief discussion of the significance of our result and the proof technique, and with an open problem.

2. Edge-packing by paths.

When all edge operation probabilities are the same, say p, the two-terminal reliability of a path from s to t of length l is p^l. Suppose that G has f edge-disjoint paths of length $l_1, l_2, \ldots, l_f$ between s and t; then

$$Rel_2(G) \geq 1 - \prod_{1 \leq i \leq f} (1 - p^{l_i}). \tag{1}$$

Hence to get the best lower bound using this technique, one has to maximize the right hand side (RHS) of the inequality (1). This is equivalent to minimizing $\prod_{1 \leq i \leq f}(1 - p^{l_i})$. This gives rise to two competing goals in getting a collection of paths yielding a good bound: maximizing f, the number of paths, and minimizing l_i's, the lengths of each path. Using Minimum Cost Network Flows [1], one can get a collection of paths with the minimum total number of edges. Itai, Perl, and Shiloach [2] have proved that given f and l, determining whether a graph has f edge-disjoint paths, each of length at most l, is NP-complete. However, both these techniques fall short of producing the best edge-packing set of paths — the set of paths that maximizes the RHS of inequality (1). We modify the proof of Itai, Perl, and Shiloach and extend their result to prove NP-completeness for the reliability maximization problem.

The decision version of our problem which we prove NP-complete is as follows:

Problem. *Given a graph G, edge operation probability p, a constant $0 \leq M \leq 1$) and two distinguished vertices s and t of G, does G have a collection of f edge-disjoint paths between s and t, of length $l_1, l_2, \ldots, l_f$ respectively, such that*

$$\prod_{1 \leq i \leq f} (1 - p^{l_i}) \leq M. \tag{2}$$

In the rest of this paper, unless otherwise mentioned, a set of paths means a collection of edge-disjoint paths between two vertices labelled s and t. We call a set of paths satisfying inequality (2) the *M-good set of paths*. Also p represents the edge operation probability, which is a rational value of polynomial length between 0 and 1. The length of a path is the number of edges in the path.

3. The main result.

Theorem. *Determining whether a probabilistic graph has a M-good set of paths, for a given M, is NP-Complete.*

Proof: It is trivial to see that the problem is in NP. To show completeness, we reduce the known NP-Complete problem, *satisfiability* of boolean formulas in conjunctive normal form to the problem of finding whether a given graph has a M-good set of paths.

Let $C = C_1 \wedge C_2 \wedge \cdots C_n$ be a boolean expression, i.e. an instance of the satisfiability problem. Let $X = \{X_1, \ldots, X_k\}$ be the set of variables. Without loss of generality, assume that in C, X_i and $\overline{X}_i$, each occurs m_i times ($1 \leq i \leq k$). To construct such an expression from a given arbitrary boolean expression see [2]. Let $m = \sum_{i=1}^{k} m_i$.

Now we construct in polynomial time, an instance of our problem from the given instance of the satisfiability problem. It consists of a graph $G(V, E)$ as described below, edge probability p on each edge and the constant $M = (1-p^5)^m (1-p^7)^n$.

The graph G

The graph G is the union of k subgraphs $G_1, G_2, \ldots, G_k$, that have only the vertices s and t in common. The subgraph G_i is associated with the variable X_i.

(1) The edge $X_{ik}(\overline{X}_{ik})$ of G_i corresponds to the kth occurrence of $X_i(\overline{X}_i)$ in C. The end vertices of $X_{ik}(\overline{X}_{ik})$ are $u_{ik}(\overline{u}_{ik})$ and $v_{ik}(\overline{v}_{ik})$.

(2) Each pair of vertices v_{ik} and $\overline{v}_{ik}$ is joined to a vertex f_{ik} by the edges (v_{ik}, f_{ik}) and $(\overline{v}_{ik}, f_{ik})$. Each f_{ij} is joined to t.

(3) Each pair of vertices $\overline{u}_{ik}$ and u_{ik} is joined to a vertex d_{ik}, which in turn is joined to a vertex e_{ik}. Each e_{ik} is directly joined to s.

(4) Each pair of vertices $\overline{u}_{ik}$ and $u_{i(k+1 \bmod m_i)}$ is joined to a vertex $g_{i(k+1 \bmod m_i)}$ and each g_{ik} in turn is joined to s.

(5) There is a pair of vertices c_i and b_i corresponding to each clause C_i of the boolean expression C. Each b_i is adjacent to t and c_i. There is an edge between c_j and $v_{ik}(\overline{v}_{ik})$ if and only if the kth occurrence of $X_i(\overline{X}_i)$ is in c_j.

Thus, the vertex set of G consists of m vertices each of type d_{ik}, e_{ik}, f_{ik} and g_{ik}; $2m$ vertices each of type u_{ik} and v_{ik}; n vertices each of the form c_i and b_i and the two vertices s and t.

Figure 1 shows the graph G associated with the boolean expression $C = (X_1, \overline{X}_1, X_2) \wedge (X_1, \overline{X}_1, X_2) \wedge (\overline{X}_1, \overline{X}_2, X_3) \wedge (X_1, \overline{X}_2, \overline{X}_3)$, with $n = 4$, $k = 3$, $m = 6$, $m_1 = 3$, $m_2 = 2$, $m_3 = 1$.

We claim that G has a M-good set of paths (with $M = (1-p^5)^m(1-p^7)^n$) if and only if the expression C is satisfiable. It is straightforward to see that the following two Lemmas prove our claim.

Lemma 1. *C is satisfiable if and only if G has a collection of $m + n$ edge-disjoint paths in which m paths are of length 5 and n paths are of length 7.*

Lemma 2. *Any M-good set of paths in G (with $M = (1-p^5)^m(1-p^7)^n$) has m paths of length 5 and n paths of length 7.*

Proof of Lemma 1: Let $C = C_1 \wedge C_2 \wedge \cdots C_n$ be satisfiable. Let $t(X_i)$ be the truth value of X_i. If $t(X_i)$ = false (true) then every G_i contributes m_i paths of length 5 passing through the edge $X_{ik}(\overline{X}_{ik})$ for each k. These paths are of the

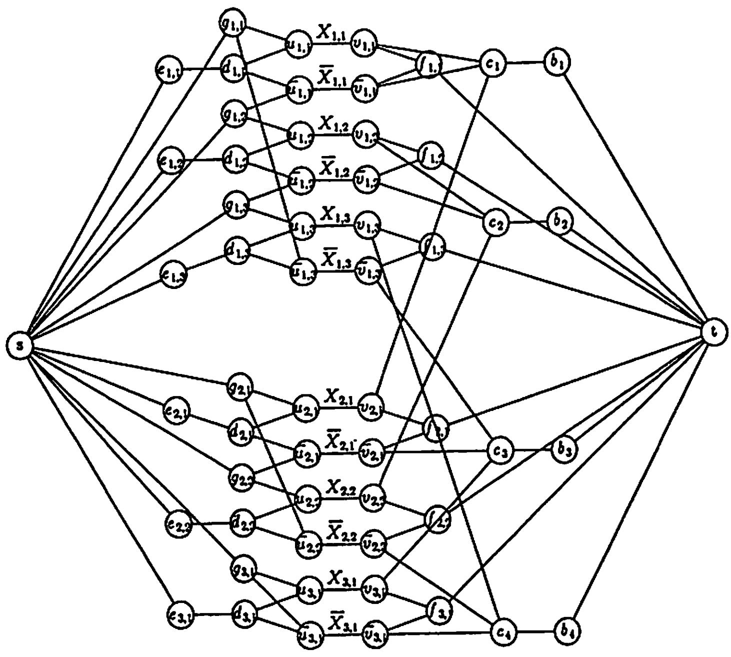

Figure 1

form $s \rightarrow g_{ik} \rightarrow u_{ik}(\overline{u}_{ik}) \rightarrow v_{ik}(\overline{v}_{ik}) \rightarrow f_{ik} \rightarrow t$. Thus we get n edge-disjoint paths of length 5.

As C is satisfiable, each C_j contains a literal X_i, such that $t(X_i)$ is true. Therefore the corresponding edge X_{ik} does not belong to any of the other m paths constructed before. Thus the path $(s, e_{ik}, d_{ik}, u_{ik}, v_{ik}, c_j, b_j, t)$ is disjoint from all m paths constructed before. Each C_j contributes one such path. So we get n edge-disjoint paths of length 7, which are edge-disjoint from m paths constructed before. Thus, if C is satisfiable, then G has m paths of length 5 and n paths of length 7, all edge-disjoint from each other.

Conversely, let G have m paths of length 5 and n paths of length 7, all edge-disjoint from each other. As the degree of t is exactly $m + n$, every edge incident with t must participate in a path. Also, a path of length 5 can only be of the form $(s, g_{ik}, u_{ik}(\overline{u}_{ik}), v_{ik}(\overline{v}_{ik}), f_{ik}, t)$, every path of length 5 must pass through a vertex of the form g_{ik}. As there are exactly m vertices of the form g_{ik}, all of them must be used to get m paths of length 5. Hence, if a path uses the edge X_{ik}, all the m_i paths corresponding to the variable i must use the edge of the type

X_{ik}; otherwise, some g_{ik} vertex is left out. Thus in G_i, all the paths of length 5 contain edges of type X_{ik} and none of the type $\overline{X}_{ik}$, or vice versa. We set $t(X_i)$ = false in the first case and true in the latter case. Thus, $t(X_i)$ is true (false) if and only if the edges $X_{i1}, X_{i2}, \ldots, X_{i(m_i)}(\overline{X}_{i1}, \overline{X}_{i2}, \ldots, \overline{X}_{i(m_i)})$ do not belong to any path of length 5. So each of the other n paths of length 7 passes through one of the c_i vertices and one of the remaining $X_{ik}(\overline{X}_{ik})$ edges. A path of length 7 can only be of the form $(s, e_{ij}, d_{ij}, u_{ij}(\overline{u}_{ij}), v_{ij}(\overline{v}_{ij}), c_k, b_k, t)$. Furthermore, there are exactly n b_i vertices and each b_i vertex is of degree 2. So each of the n b_i, and hence n c_i, vertices participates in one of the n paths of length 7. If a path passing through c_j also passes through the edge X_{ik}, then X_i is in C_j and $t(X_i)$ = true has been already assigned (i.e. , $\overline{X}_i$ has been used in our collection already). Correspondingly there is a literal X_i in the clause C_j and $t(X_i)$ is true. Thus each C_j and hence C is satisfiable. ∎

Proof of Lemma 2: It suffices to show that for any collection S of edge-disjoint paths in G, which doesn't have m paths of length 5 and n paths of length 7, the LHS of inequality (2) is greater than $(1-p^5)^m(1-p^7)^n$.

As $0 \leq (1-p^l) \leq 1$ for any $l \geq 0$, it is easy to see that choosing less than $(m+n)$ edge-disjoint paths cannot decrease the LHS of inequality (2). Hence any collection of paths which minimizes the LHS of inequality (2) has $(m+n)$ paths. This also rules out the possibility of having a path of length more than 7 in our collection. Now a path passing through c_j can only be of length 6 or 7. If all n such paths are of length 7, then we end up with n paths of length 7 and m paths of length 5. So, assume that $k(0 < k \leq n)$ of the n paths are of length 6. These paths have to pass through vertices of the type g_{ik}.

Also, the m paths using the edges (f_{ik}, t) are of length 5 if they use vertices of type g_{ik} and are of length 6 otherwise. As k of the m g_{ik} vertices are already used, at most $m-k$ of these paths can be of length 5 and the remaining k paths must be of length 6 (Note: $m > n \geq k$). Thus S has, k paths of length 6 and $n-k$ paths of length 7 passing through vertices of type c_j, $m-k$ paths of length 5, and k paths of length 6 not passing through vertices of type c_j. Then the LHS of inequality (2) for the collection S is $(1-p^5)^{m-k}(1-p^6)^{2k}(1-p^7)^{n-k}$. This product is greater than $(1-p^5)^m(1-p^7)^n$ as $(1-p^6)^2 \geq (1-p^5)(1-p^7)$ as $k \geq 1$ and $p \geq 0$. ∎

4. Conclusions.

Our result removes any real hope of designing efficient algorithms for finding the set of paths that produces the best edge-packing bound. Furthermore, it gives an application of the proof technique developed in [2].

A natural analogy to packing by paths is edge-packing by cutsets. If a graph G has f edge-disjoint cutsets of lengths $c_1, c_2, \ldots, c_f$ respectively between s and t, then $Rel_2(G) \leq \prod_{1 \leq i \leq g}(1-(1-p)^{c_i})$. Edge-packing by cutsets would yield upper bounds in a manner similar to the lower bounds obtained by edge-packing by paths. To achieve the best upper bound using this technique, we are faced with the same competing goals: maximizing the number of cutsets and minimizing the lengths of each cutset. Recently Wagner [3] has given a polynomial algorithm that can be applied to compute a collection of edge-disjoint cutsets with the minimum total number of edges. However, results analogous to those of [2] and our result, are still open for cutsets.

Acknowledgement.

I would like to thank Charlie Colbourn for suggesting the problem and its analogous cutset version, and for his comments on earlier versions of this paper.

References

1. C. J. Colbourn, "The Combinatorics of Network Reliability", Oxford University Press, 1987.
2. A. Itai, Y. Perl, Y. Shiloach, *The complexity of finding maximum disjoint paths with length constraints*, Networks 12 (1982), 277–286.
3. D. K. Wagner, *Disjoint (s,t)-cuts in a Network*, Research Memorandum No. 88-5 (June 88), Purdue University.