Computational Complexity of Loss Networks

Graham Louth*
Analysys Limited, 8-9 Jesus Lane,
Cambridge CB5 8BA, England

Michael Mitzenmacher†
Department of Electrical Engineering and Computer Science,
University of California, Berkeley, California 94720, USA

Frank Kelly
Department of Pure Mathematics and Mathematical Statistics,
University of Cambridge, 16 Mill Lane, Cambridge CB2 1SB, England

June 4, 1993

Abstract

In this paper we examine the theoretical limits on developing algorithms to find blocking probabilities in a general loss network. We demonstrate that exactly computing the blocking probabilities of a loss network is a #P-complete problem. We also show that a general algorithm for approximating the blocking probabilities is also intractable unless RP=NP, which seems unlikely according to current common notions in complexity theory. Given these results, we examine implications for designing practical algorithms for finding blocking probabilities in special cases.

1 Introduction

Loss networks are a powerful model for computer and telecommunications networks with limited resources. One aspect of the model is that customers can be turned away, or blocked, if resources are already being used to capacity. It is of great practical significance to know the probability that a customer is blocked. As customers can be of different types, depending upon the resources they wish to choose, these probabilities are collectively referred to as the blocking probabilities of the network. There exist explicit formulae for blocking probabilities, but

*Supported by the Science and Engineering Research Council
†Supported by the Winston Churchill Foundation

nevertheless they seem difficult to calculate, and much effort has been directed at finding approximations and asymptotic results.

In this paper, we examine the theoretical limits on developing algorithms to find blocking probabilities in a general loss network. In particular, we demonstrate that exactly computing the blocking probabilities of a loss network is a #P-complete problem. Since the #P-complete problems are at least as hard as NP-complete problems and in fact appear much harder, a polynomial time algorithm to find blocking probabilities becomes extremely unlikely. One natural reaction to this result would be to consider algorithms which only approximate the blocking probabilities instead of determining them exactly. We also show, however, that a general algorithm for approximating the blocking probabilities is also intractable unless RP=NP. Given these results, we examine the implications for designing practical algorithms for finding blocking probabilities for loss networks in special cases.

In many respects loss networks resemble the Ising model of statistical mechanics. Our work in this paper has been partly motivated by the important recent paper of Jerrum and Sinclair ([10], see also the review [28]), who have presented an efficient randomized algorithm to approximate the partition function of an arbitrary ferromagnetic Ising model to any specified degree of accuracy, even though the exact calculation is a #P-complete problem. It seems, however, that a loss network resembles the non-ferromagnetic, or 'spin-glass', case of the Ising model, where Jerrum and Sinclair have shown that even approximation is difficult.

2 A loss network model

We begin by defining the model of a loss network. Here we primarily follow the description given by Kelly in [15], which provides an introduction to the theory of loss networks and an overview of recent work on the subject. Our description is based on the canonical example of a loss network, a telephone network.

Consider a network of nodes connected by links labelled $j = 1, 2, \ldots, J$. Link j holds C_j circuits, where $C_j \in \mathbf{Z}_+$, the non-negative integers. A route is defined by a subset of links; the routes are labelled $r = 1, 2, \ldots, R$. Notice here that we do not restrict routes to be connected paths in the graph representing the network; as we shall see, however, our results hold even in this restricted case. A call on route r requires A_{jr} circuits from link j, where again $A_{jr} \in \mathbf{Z}_+$. It is assumed that customers requesting route r arrive as a Poisson stream with rate ν_r, and that the streams for the various routes are independent. An arriving call requesting route r is accepted only if at the time of arrival there are at least A_{jr} circuits available on each link j for $j = 1, 2, \ldots, J$. An accepted call holds those circuits simultaneously and exclusively for the duration of the call, which is independent of earlier arrival times and call durations. Assume also that calls on route r have identically distributed call durations with unit mean. If a call is not accepted, it is lost; the caller neither queues for service nor retries later.

We let $n_r(t)$ be the number of calls in progress using route r at time t. We also define the column vectors $n(t) = (n_r(t) : r = 1, \ldots, R)$ and $C = (C_j : j = 1, \ldots, J)$ and the matrix $A = (A_{jr} : j = 1, \ldots, J, r = 1, \ldots, R)$. Then the stochastic process $(n(t), t \geq 0)$ has a unique stationary distribution and under this distribution $\pi(n) = \mathbf{P}(n(t) = n)$ is given by

$$\pi(n) = G(C)^{-1} \prod_{r=1}^{R} \frac{\nu_r^{n_r}}{n_r!}, \quad n \in S(C), \tag{1}$$

where

$$S(C) = \{n \in \mathbf{Z}_+^R : An \leq C\} \tag{2}$$

and $G(C)$ is the normalizing constant or partition function

$$G(C) = \left(\sum_{n \in S(C)} \prod_{r=1}^{R} \frac{\nu_r^{n_r}}{n_r!} \right). \tag{3}$$

This result is easy to verify in the case when call distributions are exponentially distributed by noting that the distribution $\pi(n)$ given in (1) satisfies the detailed balance conditions

$$\pi(n) \cdot \nu_r = \pi(n + e_r) \cdot (n_r + 1), \quad n, n + e_r \in S(C),$$

where $e_r = (I[r' = r], r' = 1, \ldots, R)$ is a unit vector corresponding to one call on route r (cf. [1]). The insensitivity of the distribution given in (1) to call duration distributions can be deduced from the work of Kelly [13] on general arrival rates to networks of quasi-reversible queues, or by direct application of results from the theory of generalized semi-Markov processes [2].

This model can be used naturally to model connections across a circuit-switched communication network, such as a telephone network system. The defining parameters of the system, however, are simply the matrix A, the capacity vector C, and the arrival rates ν_r. The freedom available in choosing the link-route matrix A makes this class of model applicable to other problems as well, such as database locking systems, local area networks, multiprocessor interconnection architectures, mobile radio, and broadband packet networks (see [8, 14, 16, 18, 19, 21, 23]).

Notice that the stationary probability that a call requesting route r is blocked, which we will write as L_r, can be written in terms of the partition function, as

$$L_r = 1 - \sum_{n \in S(C - Ae_r)} \pi(n),$$

from which we can derive

$$L_r = 1 - G(C - Ae_r) G(C)^{-1}. \tag{4}$$

The explicit simple forms for the equilibrium distribution and the partition function would seem to suggest that we have found a complete solution to this problem. However, computing the partition function, $G(C)$, directly is quite difficult, since it requires summing over the state space $S(C)$, which, as (2) demonstrates, may grow rapidly with the number of routes or with the capacities of links. Various more refined methods have been proposed, and will be briefly discussed in Section 7, but all require an effort which grows quickly with the size of the network. We will demonstrate that computing $G(C)$ is in fact difficult in a well-defined sense; moreover, we shall see that a fundamental problem in computing $G(C)$ comes from the difficulty in computing the vectors that lie in the state space $S(C)$. Using this, we will show that computing the blocking probabilities is #P-complete.

3 The Loss Partition Problem

Surprisingly, the reason that computing the partition function is intrinsically difficult need not have anything to do with the practical problems of computing with real numbers or the rapid growth of the state space $S(C)$ when the links are given large capacities. Although these features of the problem do complicate it, by examining the problem in the restricted case where all capacities are 1 and the arrival rates are uniform and integer-valued, we can show that even when these issues are disregarded the problem remains for all intents and purposes intractable. Of course, the problem of computing the partition function in general is at least as hard as it is in this special case, so our results provide a lower bound on the worst-case complexity of the problem.

We formalize the notion of the problem of computing the partition function of a loss network, in the case where there is a uniform arrival rate and link capacities are 1, by defining the following problem:

Loss Partition

INSTANCE: A matrix $A = (A_{jr} : j = 1, \ldots, J, r = 1, \ldots, R)$ with entries in $\{0,1\}$ and a natural number ν.

QUESTION: What is the partition function

$$G = G(A, \nu) = \sum_{n \in \{0,1\}^R} I[An \leq 1] \prod_{r=1}^{R} \nu^{n_r}. \tag{5}$$

In both the above and the following we often abuse notation by using 1 to refer to the column vector whose entries are all 1 when the context makes the meaning clear. Notice that the instance of a Loss Partition problem corresponds to finding the partition function given in (3) of a loss network where all link capacities are 1 and all arrival rates are ν. Also, as the arrival rate is uniform across all routes, it is possible to write the partition function as

$$G = \sum_{k=0}^{R} N_k \nu^k,$$

where

$$N_k = \sum_{n \in \{0,1\}^R} I[An \le 1] I[1^T n = k]$$

is the number of feasible configurations with exactly k calls in progress.

We will show that Loss Partition is #P-complete. Recall that a function is in #P if it can be computed by a counting Turing machine of polynomial time complexity [27]. More intuitively, we call a problem in which one wishes to find the number of distinct solutions an enumeration problem. An enumeration problem lies in #P if there is a nondeterministic polynomial time algorithm such that for each instance of the problem, the number of distinct nondeterministic computations that lead to the acceptance of the instance is exactly the number of distinct solutions to the problem. Similarly, a problem is #P-complete if it is in #P and any other problem in #P can be reduced to it in polynomial time. The extensions of most NP-complete problems to enumeration problems can be shown to be #P-complete using *parsimonious transformations* [5].

To show that Loss Partition is #P complete, we first show that it lies in #P.

Theorem 1 *Loss Partition is in #P.*

Proof: Consider the nondeterministic Turing machine which, on being given an instance of Loss Partition, nondeterministically chooses a column vector $n \in \{0,1\}^R$ and a number from 1 to ν^{m_n}, where m_n is the number of ones in the vector n. The Turing machines accepts if $An \le 1$ and rejects otherwise. All of this can clearly be accomplished in time polynomial in the input. The number of accepting computations is

$$\sum_{n \in \{0,1\}^R} I(An \le 1) \nu^{m_n} = \sum_{n \in \{0,1\}^R} I(An \le 1) \prod_{r=1}^{R} \nu^{n_r}.$$

The final expression is just the partition function as given in (5). Thus Loss Partition lies in the class #P. □

To show that Loss Partition is #P-complete, we consider the following NP-complete and #P-complete problems.

Set Packing

INSTANCE: A collection C of finite sets and a positive integer $K \le |C|$.

QUESTION: Does C contain a subcollection of at least K mutually disjoint sets?

#Set Packing

INSTANCE: A collection C of finite sets and a positive integer $K \leq |C|$.

QUESTION: How many subcollections of at least K mutually disjoint sets does C contain?

These problems can be shown to be NP-complete and #P-complete respectively by reducing to them one of several similar problems, including Independent Set and #Independent Set [5]. Notice that in a capacity one loss network, two routes must be disjoint in order for there to be a feasible configuration with a call on each route. Thus there seems to be an intuitive connection between loss networks and the Set Packing problem. We make this intuition explicit in the theorem below.

Theorem 2 *Loss Partition is #P-complete.*

Proof: We have shown in Theorem 1 that Loss Partition is in #P. We now present a polynomial time reduction from #Set Packing to Loss Partition, which suffices to prove the theorem.

Let C and K be an instance of #Set Packing. Without loss of generality, we may suppose the elements of the sets in C are simply the integers from 1 to J for some J and that the sets in the collection C are distinct. We can create in polynomial time an instance of Loss Partition corresponding to C as follows. Let $R = |C|$ and associate a route with each set in C. The links correspond to the integers 1 to J which are elements of the sets of C. The matrix A is determined by letting A_{jr} be 1 if j is an element of the rth route (set) in C and 0 otherwise. Together with an integer ν this defines an instance of Loss Partition.

The output of Loss Partition is the partition function

$$G = \sum_{k=0}^{R} N_k \nu^k.$$

Recall that N_k is just the number of distinct feasible configurations having exactly k calls in progress. But for each such feasible configuration there is a corresponding subcollection of k disjoint sets of C. Moreover, there are certainly at most 2^R such subcollections of size k, since there are only 2^R subcollections in total.

Now take the instance of Loss Partition described above with $\nu = 2^{2R}$. The output can be thought of as an integer in base $2R$, and the N_k are simply the digits of this number. Thus the N_k can be found by repeatedly dividing by 2^{2R} and looking at the remainders. Knowing the N_k allows one to compute $\sum_{k=K}^{R} N_k$, which solves the instance of #Set Packing. Given an algorithm for Loss Partition, this computation can be done in polynomial time. □

The above proof of Theorem 1 uses an arrival rate ν exponential in R. Alternatively, one could note that G is a polynomial of degree R in ν with coefficients N_k. A polynomial algorithm for Loss Partition would allow one to find the value of the polynomial when $\nu = 1, 2, \ldots, R+1$. Using these $R+1$ values, one can compute the N_k efficiently. Indeed one could even use $R+1$ distinct rational values of ν, if Loss Partition were defined more generally to allow rational as well as integer arrival rates (see, for example, [27], fact 5).

One complaint that might be offered concerning the above result is that the routes described in the theorem may not correspond with connected paths in a graph, as might be expected in many application areas. However, it is a relatively simple exercise to extend the above argument to show that Loss Partition is #P-complete even if one restricts the problem to the case where all routes are connected paths.

Theorem 3 *Loss Partition is #P-complete in the restricted case where all routes are connected paths.*

Proof: As in the above problem, we reduce from #Set Packing. Given an instance of #Set Packing, we create an instance of Loss Partition defined by a graph such that each set corresponds to one route in the graph, two routes share a link if and only if their corresponding sets intersect, and all routes are connected paths in the graph.

Again let $R = |C|$. The graph for the Loss Partition instance consists of vertices

$$V = \{v_{i,j} : 1 \leq i \leq R, 1 \leq j \leq R+1\} \cup \{w_{\{i,j\},k} : 1 \leq i < j \leq R, 1 \leq k \leq 2\}.$$

We identify the vertex $w_{\{i,j\},k}$ with the vertex $w_{\{j,i\},k}$ when $j > i$, so each such vertex has two equivalent labels. The edges of the graph are

$$\begin{aligned} E = \; & \{(v_{i,j}, v_{i,j+1}) : 1 \leq i,j \leq R\} \cup \{(v_{i,j}, w_{\{i,j\},1}) : 1 \leq i,j \leq R, i \neq j\} \cup \\ & \{(w_{\{i,j\},1}, w_{\{i,j\},2}) : 1 \leq i < j \leq R\} \cup \{(w_{\{i,j\},2}, v_{i,j+1}) : 1 \leq i,j \leq R, i \neq j\}. \end{aligned}$$

The links of the loss network correspond to the edges.

Intuitively, we describe the routes as follows. For the ith set in C there corresponds a route which is a path from $v_{i,1}$ to $v_{i,R+1}$. The route will contain the edge $(v_{i,j}, v_{i,j+1})$ unless the ith and jth set have a non-empty intersection (where, for convenience, we say that a set does not intersect itself); in this case, the route detours through the edges $(v_{i,j}, w_{\{i,j\},1}), (w_{\{i,j\},1}, w_{\{i,j\},2})$, and $(w_{\{i,j\},2}, v_{i,j+1})$. It is clear that the routes corresponding to the ith and jth set, where $i \neq j$, intersect on the link $(w_{\{i,j\},1}, w_{\{i,j,\}2})$ only if the sets intersect, and the routes will not intersect otherwise. The size of this instance of Loss Partition is polynomial in the size of the input, proving the theorem. □

Theorems 2 and 3 shows that a generalized algorithm for exactly computing the partition function of a loss network is by current standards infeasible, even in the case where all capacities are one, the arrival rate is uniform, and all routes are connected paths. One natural inclination after seeing this result might be to consider algorithms for approximating the partition function instead of finding it exactly. For example, rejection sampling from the truncated Poisson process or simulation of the actual stochastic process are both potential means of approximating the partition function. Our next argument, however, shows that unless a widely held belief in complexity theory is false we cannot even hope to find an efficient approximation algorithm for Loss Partition.

4 Probabilistic Complexity Classes and Approximation Algorithms

In order to consider efficient approximation algorithms, we shall describe exactly what we mean by a randomized algorithm and briefly examine some complexity classes that arise once we expand our conception of an algorithm to include randomized algorithms.

In the most basic model, a randomized algorithm is one which can be run on a standard Turing machine that has an extra tape containing a string of random bits which can be read by the algorithm. This modified version is called a probabilistic Turing machine. Alternatively, one can imagine that the algorithm is allowed to flip a fair coin at any point and use the result.

The notion of a randomized algorithm leads to new complexity classes. We limit the discussion to classes of decision problems for convenience. First, let us say that a probabilistic Turing machine recognizes a decision problem if it returns the correct yes/no answer with probability greater than 1/2 for each individual instance of the problem. The machine works in polynomial time if it recognizes the decision problem in some number of steps bounded by a polynomial in the input. The error bound of a probabilistic Turing machine is the least upper bound of the probability of failure taken over all instances. The error bound, by definition, is at most 1/2 if a probabilistic Turing machine recognizes a decision problem.

We now recall the following complexity classes:

BPP, the class of *bounded probabilistic* polynomial time problems, is the class of problems recognized by a polynomial time probabilistic Turing machine with error bound $c < 1/2$.

RP, the class of *random* polynomial time problems, is the class of problems recognized by a polynomial time probabilistic Turing machine with zero probability of error when the correct answer is no.

It is clear that RP $\subseteq$ NP. Also, note that any problem in either BPP or RP can be recognized by a probabilistic Turing machine with an arbitrarily small error bound simply by testing the problem some large (but polynomial) number of times. From this we can easily show that RP $\subseteq$ BPP. It has not been proven whether any of these inclusions are proper; however, it is widely thought that RP $\neq$ NP.

We shall also consider randomized algorithms which approximate a desired result. In discussing approximation algorithms, we will use the standard of a *fully polynomial randomized approximation scheme* established by Karp and Luby in [12]. Given non-negative real numbers a, b, and c, we say that b *approximates* a within ratio $1 + c$ if

$$a(1 + c)^{-1} \leq b \leq a(1 + c).$$

If f is a function from problem instances to the real numbers, a randomized approximation scheme for f is a randomized algorithm which, when given an instance x of a problem and a real number $\epsilon \in (0, 1]$, yields a real number that approximates $f(x)$ within ratio $(1 + \epsilon)$ with a probability of at least $3/4$. By using repeated trials one can reduce the probability of error from $1/4$ to any desired value $\delta > 0$, by running the algorithm $0(\log \delta^{-1})$ times and using the median of the results ([10],[11]). Finally, a *fully polynomial randomized approximation scheme*, or *fpras*, is a randomized approximation scheme that runs in time polynomial both in the size of the problem instance given as input and ϵ^{-1}. An fpras approximates a function efficiently, although it naturally can use larger time in order to gain accuracy.

We now show that approximating the partition function is also infeasible in a specific, complexity-based sense. We do this by showing that the existence of an fpras for Loss Partition would yield a polynomial time randomized algorithm for Set Packing, and hence for all problems in NP. Since it is widely believed that RP $\neq$ NP, it seems unlikely that an fpras for Loss Partition exists.

Theorem 4 *There can be no fpras for Loss Partition unless RP = NP.*

Proof: Suppose we are given an instance of Set Packing with a collection C of sets and integer K. As in Theorem 2, generate a corresponding instance of Loss Partition. In this case we choose $\nu = 2^{2R}$. With this choice of ν, we see that if there is a feasible configuration with K calls in progress, then $G = \sum_{k=0}^{R} N_k \nu^k \geq 2^{2RK}$, while if there is no such configuration, then $G = \sum_{k=0}^{R} N_k \nu^k \leq 2^R 2^{2R(K-1)} = 2^{-R} 2^{2RK}$. Since feasible configurations corresponds to collections of disjoint sets in C, $G \geq 2^{2RK}$ if C has a subcollection of K disjoint sets, and $G \leq 2^{-R} 2^{2RK}$ otherwise.

Now suppose there exists an fpras for Loss Partition. Then these two cases are distinguishable by the fpras, and using the fpras provides a randomized algorithm for deciding the Set Packing question. Thus Set Packing $\in$ BPP. Since BPP is closed under polynomial time reductions, and Set Packing is NP-complete, this yields that NP $\subseteq$ BPP. However, by the work done by Ko [17], this implies that RP = NP. $\Box$

An entirely similar argument shows that there can be no fpras for Loss Partition even in the restricted case where all routes correspond to connected paths unless RP = NP. However we have been unable to recast the proof of Theorem 4 so that only bounded or polynomially growing arrival rates are used: it remains possible that an fpras may exist for Loss Partition in the restricted case of low arrival rates.

5 Theoretical Implications for Computing Loss Probabilities

The complexity of computing the partition function leads to some interesting conclusions regarding the ability to compute blocking probabilities for general loss networks. In particular, recall by equation (4) that

$$G(C) = \frac{1}{1-L_r} G(C - Ae_r). \tag{6}$$

Thus if a tool existed for finding the exact blocking probabilities on a loss network, $G(C)$ could be determined recursively in at most $\sum_j C_j$ steps. A polynomial time method for finding blocking probabilities would therefore immediately yield a polynomial time algorithm for solving Loss Partition. By what we have shown in Theorem 2, such an algorithm would be extremely unlikely.

Moreover, we can demonstrate similarly that an fpras for $L_r/(1-L_r)$, the odds of a call being lost on route r, yields an fpras for the partition function.

Theorem 5 *There can be no fpras for the odds of a call being blocked on a given route of a loss network unless RP = NP.*

Proof: Since $1/(1-L_r) = 1 + L_r/(1-L_r)$, an fpras for $L_r/(1-L_r)$ immediately yields one for $1/(1-L_r)$; take the estimate given by the fpras and add 1 to find a suitable estimate for $1/(1-L_r)$.

Now suppose there is an fpras for $1/(1-L_r)$. Then we can create an fpras that approximates the partition function within ratio $(1+\epsilon)$ as follows. Let $c = \sum_j C_j$ and

$$\epsilon' = \frac{\min(\frac{1}{4}, \frac{\epsilon}{2})}{c}.$$

Compute an approximation for $G(C)$ recursively from equation (6) by using the fpras for $1/(1-L_r)$, approximating it at each step within a ratio of $(1+\epsilon')$ with probability at least $1-(4c)^{-1}$. Then with probability at least $3/4$ the final estimate E for $G(C)$ satisfies

$$(1+\epsilon')^{-c} G(C) \leq E \leq (1+\epsilon')^{c} G(C),$$

but

$$(1+\epsilon')^{c} \leq \exp(c\epsilon') < 1 + c\epsilon' + (c\epsilon')^2 < (1+\epsilon).$$

Since c is polynomial in the input size, this yields an fpras for $G(C)$. By Theorem 4, this would imply that RP = NP. □

It would seem that Theorem 5 should be extended to show that the blocking probabilities themselves cannot be approximated by an fpras unless RP = NP. However, this does not seem to follow immediately. The problem lies in the case where the blocking probability is extremely close to 1. For example, if L_r is close to 1, unless ϵ is chosen small enough, the approximation algorithm might simply return 1. For this problem, such a return value would be useless, since the recursive algorithm for approximating $G(C)$ requires using values for $1/(1 - L_r)$.

One possible means of fixing this would be to bound L_r away from 1 and then choose an appropriate ϵ. This method does not seem feasible, however, for the following reason. By increasing the arrival rate of calls to the network, one increases the blocking probabilities. In particular, for a given instance of Loss Partition, increasing the number of digits in the arrival rate ν by some polynomial factor increases the arrival rate exponentially, which in turn could cause the blocking probabilities to approach 1 exponentially quickly. In other words, $1/(1 - L_r)$ can grow exponentially in the size of the problem instance. To approximate $1/(1 - L_r)$ within a ratio of $1 + \epsilon$, where ϵ^{-1} is bounded by a polynomial in the size of the input, would seem to require approximating L_r to within a ratio $1 + \epsilon'$, where $(\epsilon')^{-1}$ is exponential in the size of the input.

Indeed, this appears to be a general problem that must be considered when using fpras to approximate probabilities in a [0,1) range. We can, however, say something about the ability to approximate L_r by considering an algorithm which does not charge for the necessity of making better approximations as L_r approaches 1.

Theorem 6 *Unless RP = NP, there does not exist an algorithm which does the following: Given a loss network, a route r on the network, and $\epsilon \in (0, 1]$, the algorithm returns an estimate E for L_r in time bounded by a polynomial in the size of the loss network and ϵ^{-1} such that*

$$(1 + \epsilon(1 - L_r))^{-1} L_r \leq E \leq (1 + \epsilon(1 - L_r)) L_r$$

with probability at least 3/4.

Proof: Suppose such an algorithm existed. Then we use it to create an fpras for $1/(1 - L_r)$ as follows. Given an ϵ', approximate L_r with the algorithm above using $\epsilon = \frac{\epsilon'}{2}$. Simple algebraic manipulation then shows that $1/(1 - E)$ approximates $1/(1 - L_r)$ within a ratio of ϵ'. Thus the existence of an algorithm as described above implies RP = NP by Theorem 5. □

Notice that the algorithm described in the statement of Theorem 6 is very much like an fpras; in fact, it is apparent that if L_r can be bounded away from 1, such an algorithm is in fact equivalent to an fpras. Theoretically, the only difference is when L_r approaches 1, where such an algorithm grows more accurate without requiring extra time for the gain in efficiency.

6 The non-frustrated case

As we have seen, the #Set Packing problem can naturally be reduced to Loss Partition. By considering another natural reduction, from #Independent Set to Loss Partition, we find an interesting subcase of the Loss Partition problem.

To make the connection between the problems, we consider the following graph based on the routes of a loss network. Let $I(R, A) = ([R], E)$ be an undirected graph consisting of nodes $r \in [R]$ corresponding to the routes $r = 1, \ldots, R$ of the original loss system. The edge $e = \{r_1, r_2\} \in E$ if and only if routes r_1 and r_2 share at least one link of positive capacity, that is there exists $j \in \{i, \ldots, J\}$ with $C_j > 0$ such that $A_{jr_1} > 0$ and $A_{jr_2} > 0$. Call this the *route interaction graph* for the loss network defined by R and A.

Notice that many loss networks can share the same route interaction graph. Also, note that if all the capacities are 1, then a configuration $n \in \{0, 1\}^R$ satisfies $An \leq 1$ if and only if the set of routes $\{r : n_r = 1\}$ is an independent set on the route interaction graph. In other words, for a given instance of Loss Partition, the feasible configurations of i calls and the independent sets of size i on the graph are in a one-to-one correspondence.

It is easy to see that Theorems 2 and 3 can be modified to reduce #Independent Set, instead of #Set Packing, to Loss Partition. For example, given an instance of #Independent Set $G = (V, E)$ and b, in polynomial time one can construct the following instance of Loss Partition for which G is the route interaction graph. Let $R = |V|$ and associate a route with each element of V. For each edge $e = \{r_1, r_2\} \in E$, set $A_{er_1} = A_{er_2} = 1$, so that routes r_1 and r_2 have a link in common, and set $A_{er} = 0$ otherwise. Together with a positive integer ν this defines an instance of Loss Partition. Furthermore, it is clear that the corresponding route interaction graph is just G. By modifying this construction appropriately one can similarly reduce #Independent Set to the restricted case of Loss Partition where all routes are paths.

By now proceeding as in Theorem 2, one could use an algorithm for Loss Partition to find the coefficients N_k, which correspond to the number of feasible configurations with k calls. Since this equals the number of independent sets of size k on G, knowing the N_k allows one to solve the #Independent Set problem. Similarly, the construction in Theorem 4 can be modified to make use of the Independent Set problem instead of Set Packing.

An interesting case arises when one examines the situation where the route interaction graph is bipartite. Let us call a network *non-frustrated* if the route interaction graph is bipartite and *frustrated* otherwise. Frustration is a simple measure of whether the interactions between routes along different paths through the network are in phase or out of phase. A frustrated network looks somewhat like a "spin-glass" in statistical mechanics, whereas a non-frustrated network is more like a regular lattice.

Now we can consider the subproblem of Loss Partition restricted where the route interaction graph is non-frustrated, which we shall call Non-Frustrated Loss Partition. By the same argument as the one presented above, we could show that Non-Frustrated Loss Partition is still #P-complete if #Independent Set is #P-complete when restricted to bipartite graphs. In fact, Provan and Ball [24] proved that both the problem of finding the total number of independent sets in a bipartite graph and the problem of finding the number of independent sets of largest size in a bipartite graph are #P-complete, and either of these quantities is easily derived from the N_k. Thus these problems can be reduced in polynomial time to Non-Frustrated Loss Partition, so it too is #P-complete.

Notice, however, that the argument used in Theorem 4 to show that the non-existence of an fpras for Loss Partition cannot be extended to this case. This is because for a general graph the problem of finding the maximum size of an independent set is NP-complete, while for a bipartite graph the corresponding problem can be solved in polynomial time. (For example, see [6].) The proof in Theorem 4 requires that the existence problem (either Set Packing or Independent Set) be NP-complete, and in the non-frustrated case, it is not.

This observation leads us to the following conjecture:

Conjecture *There exists an fpras for Non-Frustrated Loss Partition.*

Indeed, Jerrum and Sinclair [10] have found an fpras for the ferromagnetic case of the Ising problem using a transformation that yields a rapidly mixing Markov chain, even though the general case is #P-complete. It remains unclear whether their methods can be applied to find an fpras for this problem as well.

7 Discussion

The importance of loss networks as models has led to intense interest in computational algorithms, and many methods have been proposed to calculate the partition function (3) and the loss probabilities (4). For work on exact methods see Dziong and Roberts [4], Ross and Tsang [25] and the recent review of Conway et al. [3]. In practice direct simulation of the underlying stochastic process is often used to estimate loss probabilities; other approximation techniques use the truncated product form (1) as a basis for Monte Carlo estimation – see Harvey and Hills [7] and Ross and Wang [26] for methods based on refinements of acceptance-rejection sampling.

All of the methods proposed require an effort which grows quickly with the size of the network. This observation is largely explained by the work reported in this paper, and in particular by the explicit connections made between the Set Packing and Loss Partition problems. As we have seen these connections can be made even when all capacities in the loss network are one. The Loss Partition problem for variable capacities is at least as difficult: instead it is interesting to consider briefly a restricted version of the problem where the matrix A defining the topology of a loss network is fixed, and an instance of the problem is defined by the vectors ν and C of traffics and capacities. For this version of the problem the exact algorithm of Pinsky and Conway [22] has time complexity $\mathrm{O}\left(\prod_{j=1}^{J} C_j\right)$, and thus is polynomial in link capacities and exponential in the size of the input description necessary to define the link capacities. However the Monte Carlo estimation technique of Ross and Wang [26] requires a computational effort that is independent of link capacities, and indeed the limit results reviewed in [15] and the bounds obtained in a special case by Mitra [20] suggest that approximation becomes *simpler* with larger capacities.

Closed queueing networks form a further class of widely used model with partition function akin to that of the loss network model (see, for example, [13]). In some respects the model is richer, and it may be possible to delineate various complexity classes within the model. We conclude with an example to illustrate this point. Consider a network with J queues and J customers. Suppose each customer has a subset of the J queues that it cycles around, and suppose each queue serves at infinite rate when it contains two customers. Thus a state of the network is any feasible configuration with a single customer in each queue. Let $A_{ij} = 1$ if customer i visits queue j, and let $A_{ij} = 0$ otherwise. Then the number of states of the network, and hence the partition function under the simplest assumptions on service rates, is just the permanent of the matrix A. Now calculation of the permanent of a $0-1$ matrix is a well-studied problem, and has the intriguing property that for a wide class of matrices it is both #P-complete and yet an efficient approximation algorithm exists [9].

References

[1] BROCKMEYER, E., HALSTROM, H.L., AND JENSEN, A. (1948) *The Life and Works of A.K. Erlang*. Academy of Technical Sciences, Copenhagen.

[2] BURMAN, D.Y., LEHOCZKY, J.P., AND LIM, Y. (1984) Insensitivity of blocking probabilities in a circuit switched network. *Advances in Applied Probability* 21, 850-859.

[3] CONWAY, A.E., PINSKY, E. AND TRIDANDAPANI, S. (1993) Efficient decomposition methods for the exact analysis of circuit-switched networks. *J. Assoc. Comput. Mach.*

[4] DZIONG, Z. AND ROBERTS, J.W. (1987) Congestion probabilities in a circuit-switched integrated services network. *Performance Evaluation* 7, 267-284.

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

[6] HARARY, F. (1976) *Graph Theory*. Addison-Wesley, Reading, MA.

[7] HARVEY, C. AND HILLS, C.R. (1979) Determining grades of service in a network. In 9th International Teletraffic Congress.

[8] HUI, J.Y. (1990) *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer, Boston.

[9] JERRUM, M.R. AND SINCLAIR, A. (1989) Approximating the permanent. *SIAM Journal Computing* 18, 1149-1178.

[10] JERRUM, M.R. AND SINCLAIR, A. (1990) *Polynomial Time Approximation Algorithms for the Ising Model*. Technical Report CSR-1-90, Dept. of Computer Science, University of Edinburgh.

[11] JERRUM, M.R., VALIANT, L.G. AND VAZIRANI, V.V. (1986) Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* 43, 169-188.

[12] KARP, R.M. AND LUBY, M. (1983) Monte-Carlo algorithms for enumeration and reliability problems. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Longe Beach, CA, 56-64.

[13] KELLY, F.P. (1979) *Reversibility and Stochastic Networks*. Wiley, Crichester.

[14] KELLY, F.P. (1985) Stochastic models of computer communication systems. *J. Roy. Statist. Soc. Ser. B* 47, 379-395.

[15] KELLY, F.P. (1991) Loss Networks. *The Annals of Applied Probability*, Vol. 1, No. 3, 319-378.

[16] KELLY, F.P. (1991) Effective bandwidths at multi-class queues. *Queueing Systems* 9, 5-16.

[17] KO, KER-I. (1982) Some observations on probabilistic algorithms and NP-hard problems. *Information Processing Letters* 14, 39-43.

[18] LAGARIAS, J.C., ODLYZKO, A.M., AND ZAGIER, D.B. (1985) Realizable traffic patterns and capacity of disjointly shared networks. *Comput. Networks* 10, 275-285.

[19] LOUTH, G.M. (1991) *Stochastic Networks: Complexity, Dependence, and Routing*. Ph.D. thesis, University of Cambridge.

[20] MITRA, D. (1987) Asymptotic analysis and computational methods for a class of simple, circuit-switched networks with blocking. *Adv. Appl. Prob.* 19, 219-239.

[21] MITRA, D. AND WEINBERGER, P.J. (1984) Probabilistic models of database locking: Solutions, computational algorithms and asymptotics. *J. Assoc. Comput. Mach.* 31, 855-878.

[22] PINSKY, E. AND CONWAY, A.E. (1993) Computational algorithms for blocking probabilities in circuit-switched networks. *Annals of Operations Research*.

[23] PINSKY, E. AND YEMENI, Y. (1984) A statistical mechanics of some interconnection networks. In *Performance '84* (E. Gelenbe, ed.) North-Holland, Amsterdam.

[24] PROVAN, J. AND BALL, M. (1983) The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM Journal of Computing* 12, 777-788.

[25] ROSS, K.W. AND TSANG, D. (1990) Teletraffic engineering for product-form circuit-switched networks. *Adv. Appl. Prob.* 22, 657-675.

[26] ROSS, K.W. AND WANG, J. (1992) Monte-Carlo summation applied to product-form loss networks. *Prob. Eng. Inf. Sci.* 6, 323-348.

[27] VALIANT, L.G. (1979) The Complexity of Enumeration and Reliability Problems. *SIAM Journal of Computing* 8, 410-421.

[28] WELSH, D.J.A. (1990) The computational complexity of some classical problems from statistical physics. In *Disorder in Physical Systems* , Grimmett, G.R. and Welsh, D.J.A. (eds), Oxford University Press, 307-321.