Integral Expressions for the Numerical Evaluation of Product Form Expressions Over Irregular Multidimensional Integer State Spaces

D. Manjunath
Department of Electrical Engg.
Indian Institute of Technology, Bombay
Powai, Mumbai 208 016 INDIA
dmanju@ee.iitb.ernet.in

Biplab Sikdar
Department of ECSE
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
bsikdar@networks.ecse.rpi.edu

Abstract

We consider stochastic systems defined over irregular, multidimensional, integer spaces that have a product form steady state distribution. Examples of such systems include closed and BCMP type of queuing networks, polymerization and genetic models. In these models the system state is a vector of integers, $\mathbf{n} = [n_1, \cdots, n_M]$ and the steady state solution has product form of the type $\pi(\mathbf{n}) = \prod_{i=1}^{M} f_i(n_i)$. To obtain useful statistics from such product form solutions, $\pi(\mathbf{n})$ has to be summed over some subset of the space over which it is defined. We consider situations when these subsets are defined by a set of equalities and inequalities with integer coefficients, as is most often the case and provide integral expressions to obtain these sums. Typically, a brute force technique to obtain the sum is computationally very expensive. Algorithmic solutions are available for only specific forms of $f_i(n_i)$ and shapes of the state space. In this paper we derive general integral expressions for arbitrary state spaces and arbitrary $f_i(n_i)$. The expressions that we derive here become especially useful if the generating functions $f_i(n_i)$ can be expressed as a ratio of polynomials in which case, exact closed form expressions can be obtained for the sums. We demonstrate the wide applicability of the integral expressions that we derive here through three examples in which we model finite highway cellular systems, copy networks in multicast packet switches and a BCMP queuing network modeling a multiuser computer system.

Keywords: Product form solutions, Queuing networks, Normalizing constant.

1 Introduction

Consider a stochastic system whose state is defined by a $M-$dimensional vector $\mathbf{n} = [n_1, n_2, \cdots, n_M]$, where n_i are non negative integers for $i = 1, 2, \cdots, M$. In many systems, not all $\mathbf{n}$ will be defined and let $\mathcal{S}$ be the set of defined states. In this paper we will assume that $\mathcal{S}$ is defined by a set of linear constraints in n_i. In many models of such systems the steady state probability of the system being in state $\mathbf{n}$, $\pi(\mathbf{n})$, has a product form expression of the type

$$\pi(\mathbf{n}) = \begin{cases} (G(\mathcal{S}))^{-1} \prod_{i=1}^{M} f_i(n_i) & \mathbf{n} \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $G(\mathcal{S})$ is the normalization constant obtained to make $\pi(\mathbf{n})$ sum to unity over the state space $\mathcal{S}$. Performance measures like marginal distributions of n_i and their moments are obtained by summing $\pi(\mathbf{n})$ over a subset of $\mathcal{S}$. This summation is, typically, computationally expensive if a brute force enumeration technique is used. For some specific situations like closed queuing networks and stochastic knapsack models where the $f_i(n_i)$ have the form $\rho_i^{n_i}/n_i!$ or $\rho_i^{n_i}$, efficient recursive algorithms to sum over the state space are available. Queuing networks with constant population are considered in [1] stochastic knapsacks in [2]. In this paper we derive a transform technique to sum functions of the form $\prod_{i=1}^{M} f_i(n_i)$ over a state space defined by a set of linear equality and/or inequality constraints with integer coefficients. With this method, if the generating function of $f_i(n_i)$ can be written as a ratio of polynomials in z, the transform variable, then a closed form expression for the sum is obtained. Before describing the technique, we discuss examples of some models where our technique can be used.

Example 1: In [3], Kaufman discusses complete and partial buffer sharing policies in a communication node. Messages arrive according to a Poisson process to a communication node. M classes of messages share a total of K buffers. A class i message requires b_i buffers. In the partial sharing policy, K_i buffers are dedicated to class i messages and K_0 buffers belong to the common pool. A message of class i is admitted if b_i buffers are available from class i and/or common pools. If it is not admitted, the message is considered lost. Let n_i be the number of class i messages in the node. Let $\mathbf{n} = [n_1, n_2, \cdots, n_M]$ be the state of the system and $\mathcal{S}$ the state space for $\mathbf{n}$ under a given sharing policy. In the complete sharing policy, all the buffers form the common pool and the state space $\mathcal{S}$ is given by

$$\mathbf{n} \in \mathcal{S} \ \text{ if} \sum_{i=1}^{M} n_i b_i \leq K$$

The complete sharing policy is like the stochastic knapsack described in [2]. Foschini and Gopinath [4] also use a similar model for sharing memory in a multiprocessor system. Kaufman shows that the steady state solution for this model is of the form of Eqn. 1. An example of a parameter of interest would be the probability of the loss of messages of class i. To obtain this we will have to sum $\pi(\mathbf{n})$ over a subset of $\mathcal{S}$ all of whose elements are vectors $\mathbf{n}$ such that there are no buffers left to accommodate a new class i message.

Example 2: Consider a cellular system with M cells. Assume Poisson call arrivals, arbitrary holding time distributions, no hand offs and a maximum packing channel assignment algorithm. Let n_i denote the number of active calls in cell i and the vector $\mathbf{n} = [n_1, n_2, \cdots, n_M]$ denote the state of the system. The set of admissible states for $\mathbf{n}$, $\mathcal{S}$, depends on the geographical layout of the cells and the channel assignment algorithm used. It can be shown that $\mathcal{S}$ is defined by

$$\mathbf{n} \in \mathcal{S} \quad \text{if} \quad B\mathbf{n} \leq \mathbf{N}$$

where B is a $M \times M$ matrix and $\mathbf{N}$ is an $M-$dimensional vector. In [6], Everitt shows how to choose the matrix B and the vector $\mathbf{N}$ from the geographical layout of the cells, the number of channels in the system, the distribution of the channels and the channel assignment algorithm. To calculate the blocking probability P_{B_i}, of a call arriving to cell i, we first define the set $\mathcal{S}_{B_i} \subseteq \mathcal{S}$ that contains all $\mathbf{n}$ satisfying the blocking conditions for cell i. Then,

$$P_{B_i} = \frac{G(\mathcal{S}_{B_i})}{G(\mathcal{S})} \quad \text{where} \quad G(\mathcal{S}_{B_i}) = \sum_{\mathbf{n} \in \mathcal{S}_{B_i}} \left\{ \prod_{i=1}^{M} f_i(n_i) \right\}$$

Example 3: Consider a discrete time copy network like the one described in [7], that we call the Lee copy network (LCN). The structure of the LCN is shown in Fig. 1. Time is slotted and all the inputs are synchronized such that packet arrivals occur at the beginning of a slot. An $M \times N$ copy network works as follows. Inputs are numbered from 1 to M at the beginning of a slot by a scheduler. Let c_i be the number of copies requested by port i. In the simplest form of scheduling, an acyclic service discipline can be used in which the input ports are numbered starting from the top of the network. At the beginning of a slot the running adder obtains the running sum $\sum_{j=1}^{i} c_j$ for $i = 1 \cdots M$, i.e., at all the ports. In this service policy, port i is serviced in a slot (all the copies requested by the input packet are made) if $\sum_{j=1}^{i} c_j \leq N$ in that slot. This is because the copy network can only make N copies in a slot. Let $f_i(c_i)$ be the probability that input i requests c_i copies. In this simple design if in a slot the sum of all the copy requests exceeds N, then only the first k inputs that satisfy the conditions

$$\sum_{i=1}^{k} c_i \leq N \quad \textbf{and} \quad \sum_{i=1}^{k+1} c_i > N$$

will be "served" and the requests from the other inputs, $k+1, K+2, \cdots N$, will be discarded. It is of interest to know the probability that a request at input i will be served. It is also of interest to

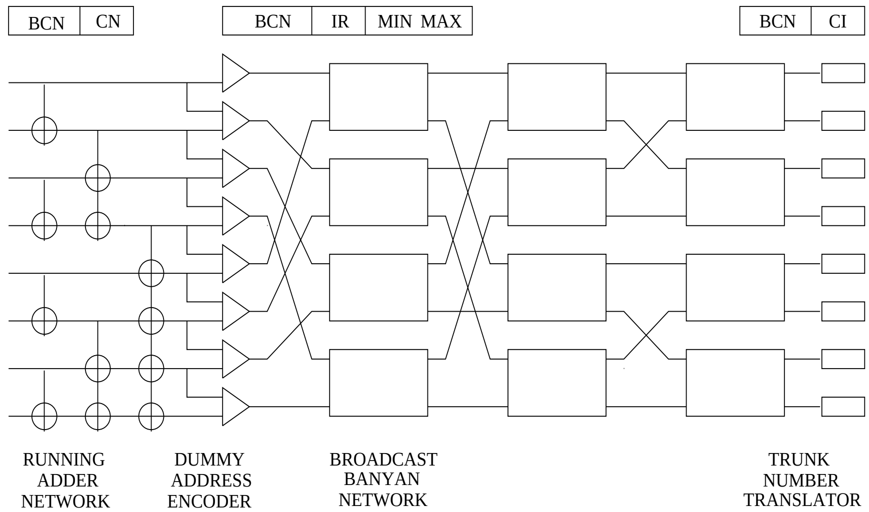

Figure 1: Lee's Copy Network for a Multicast Packet Switch [7]

know the throughput of the copy network. Lee [7] obtains a Chernoff bound on the probability of copy requests from i being served. We provide an exact analysis using techniques developed here.

Example 4: Consider a $N \times N$ Knockout switch described in [20]. In each slot a number of inputs will want to transmit on a given output. Let n_i be the number of packets from the inputs wanting to go to output i. In the Knockout switch, if $n_i \leq L$, all packets are transferred to the output. If $n_i > L$, only L are transferred to the output and $n_i - L$ are dropped. It is of interest to know the throughput and hence loss probability of this switch architecture. Let $\mathbf{n} = [n_1, n_2, \cdots, c_N]$, represent the system state in a slot. If we assume saturation input, a packet is present in every input in every slot, then only those $\mathbf{n}$ satisfying $\sum_{i=1}^{N} n_i = N$ will be a valid state. In [20], an upper bound on the loss probability from a port is given as $\sum_{i=L+1}^{N} (n_i - L) f_i(n_i)$. The exact loss probability can be obtained as

$$\sum_{\mathbf{n} \in \mathcal{S}_l} (n_i - L) \pi(\mathbf{n})$$

where $\mathcal{S}_l$ is the set of $\mathbf{n}$ that will result in a loss from input i, i.e., the set of $\mathbf{n}$ that satisfy $n_i > L$ and $\sum_{i=1}^{N} n_i = N$.

In the context of queuing networks transform based techniques to sum the product form solution over a state space where the total population in the system is a constant are known. In [8], Harrison reports a closed form expression for a closed queuing network with M single server nodes and N jobs in the network, i.e., this result only considers $f_i(n_i) = \rho_i^{n_i}$ and a state space defined by $\sum_{i=1}^{M} n_i = N$. Gordon [9] derives this result differently and extends it to closed queuing networks with multiple servers, i.e., he allows $f_i(n_i)$ of the form

$$f_i(n_i) = \begin{cases} \frac{\rho^{n_i}}{n_i} & \text{for } n_i \leq m_i \\ \frac{\rho^{m_i}}{m_i} & \text{for } n_i > m_i \end{cases}$$

Gerasimov [10] obtains results very similar to that obtained by Gordon and extends it to BCMP queuing networks with multiple classes of jobs but he restricts himself to closed queuing networks without infinite server queues. In all these results the case where the state space is specified by more than one constraint like in the examples given above has not been discussed. In this paper we obtain analytical expressions for the the sum of $\pi(\mathbf{n})$ over an irregular state space defined by a set of linear constraints with integer coefficients. Our interest is not to prove the existence of a product form solution to these systems. Rather, we assume the existence of the product form solution and obtain analytic expressions for the normalizing constant in these networks. In Section 2 we discuss the types of constraints that are applicable to our method and explore the single constraint case. In Section 3 we extend the algorithm to the multiple constraint case and in Section 4 we discuss numerical evaluation of the sum using our technique. Examples are discussed in Section 5 in which we apply our technique in the analysis of three different systems. We conclude with discussions in Section 6.

2 Single Constraint

As described earlier, we assume that the system state is a vector $\mathbf{n} = [n_1, n_2, \cdots, n_M]$, an M-dimensional vector of integers and the steady state probability of the system is given by Eqn. 1. We first consider the summing of $\pi(\mathbf{n})$ over the state space defined by a single constraint. We first discuss the equality constraint which can be applied to a closed queuing network as a special case. Next, we consider the "less than or equal" constraint that which can be used to solve the stochastic knapsack with arbitrary $f_i(n_i)$.

2.1 The Equality Constraint

Let us first consider a single equality constraint on the state space given by

$$\sum_{i=1}^{M} n_i = N \tag{2}$$

where N is a positive integer. Denote by $\mathcal{S}_{eq}(N,M)$ the set of all vectors $\mathbf{n}$ that satisfy the above constraint. Define $G(\mathcal{S}_{eq}(N,M))$ as

$$G\left(\mathcal{S}_{eq}(N,M)\right) \equiv \sum_{\mathbf{n}\in\mathcal{S}_{eq}(N,M)} \prod_{i=1}^{M} f_i(n_i)$$

In the context of a closed queuing network, $G(\mathcal{S}_{eq}(N,M))$ is the normalizing constant in a network of M queues with a constant population of N.

Define $\delta(k)$ as

$$\delta(k) \equiv \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases}$$

which, in turn, can be written in terms of the following contour integral on the complex plane

$$\delta(k) = \oint z^{k-1} dz$$

where the integration is on the unit circle and z is a complex variable. (This can also be thought of as the inverse z-transform of $\delta(k)$, the discrete impulse function.)

Rewrite the constraint in Eqn. 2, as $\sum_{i=1}^{M} n_i - N = 0$. $G\left(\mathcal{S}_{eq}(N,M)\right)$ can then be rewritten as

$$\begin{aligned} G\left(\mathcal{S}_{eq}(N,M)\right) &= \sum_{\mathbf{n}\in\mathcal{S}_{eq}(N,M)} \prod_{i=1}^{M} f_i(n_i) \\ &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} \left[\prod_{i=1}^{M} f_i(n_i)\right] \delta(n_1 + \cdots + n_M - N) \\ &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} \left[\prod_{i=1}^{M} f_i(n_i)\right] \oint z^{(n_1+\cdots+n_M-N-1)}\, dz \\ &= \oint \left[\frac{1}{z^{(N+1)}}\right] \prod_{i=1}^{M} \mathcal{F}_i(z)\, dz \end{aligned}$$

Here $\mathcal{F}_i(z)$ is the $z-$transform of $f_i(n_i)$. Evaluation of the last integral involves taking the sum of the residues of the integrand at the poles inside the unit circle of the integrand. This result is similar to that obtained by Gordon [9] for closed queuing networks except that we generalize the result to include any $f_i(n_i)$. We can further generalize Eqn. 2 to include constraints of the form

$$b_1 n_1 + b_2 n_2 + \cdots + b_M n_M = N \tag{3}$$

where b_is and N are integers and $G\left(\mathbf{S}_{eq}(N,M)\right)$ is

$$G\left(\mathbf{S}_{eq}(N,M)\right) = \oint \left[\frac{1}{z^{(N+1)}}\right] \prod_{i=1}^{M} \mathcal{F}_i(z^{b_i})\, dz \tag{4}$$

Constraints of the type in Eqn. 3 have been used by Kelly [11] to describe product form models for social interactions.

2.2 Inequality Constraints

We first look at the "less than or equal to ($\leq$)" type of inequality constraint like those used in defining the stochastic knapsack and then comment on the "greater than ($>$)" type of inequality constraint. Let $\mathcal{S}_{le}(N, M)$ be the set of $\mathbf{n}$ satisfying the constraint

$$b_1 n_1 + b_2 n_2 + \cdots + b_M n_M \leq N$$

where the b_is and N are integers. Denote by $\mathcal{S}_{le}(N, M)$ the set of all $\mathbf{n}$ that satisfy this constraint and let $G(\mathcal{S}_{le}(N, M))$ be defined as follows.

$$G\left(\mathcal{S}_{le}(N, M)\right) \equiv \sum_{\mathbf{n} \in \mathcal{S}_{le}(N,M)} \prod_{i=1}^{M} f_i(n_i)$$

Before we proceed we first define $\Phi_N(k)$ as follows,

$$\Phi_N(k) \equiv \begin{cases} 1 & \text{for } k \leq N \\ 0 & \text{for } k > N \end{cases}$$

$\Phi_N(k)$ is a delayed discrete step function reversed in time. The contour integral representation for this function is

$$\begin{aligned} \Phi_N(k) &= \sum_{i=0}^{N} \delta(k - i) \\ &= \oint \left[\frac{z^{(N+1)} - 1}{z - 1} \right] \left[\frac{z^k}{z^{(N+1)}} \right] dz \end{aligned}$$

Proceeding as before, we can obtain the following.

$$G(\mathcal{S}_{le}(N, M)) = \oint \left[\frac{z^{(N+1)} - 1}{z^{(N+1)}(z - 1)} \right] \prod_{i=1}^{M} \mathcal{F}_i(z^{b_i})\, dz \tag{5}$$

Note that Eqn. 5 is identical to Eqn. 4 except for the additional term, $(z^{N+1} - 1)/(z - 1)$ in the integrand. We can obtain $G(\mathcal{S}_{le}(N, M))$ by adding another dimension to $\mathbf{n}$ and converting the inequality constraint to an equality constraint. This is similar to introducing a slack variable to convert an inequality constraint into an equality constraint in a linear programming problem.

If the state space is constrained by a "greater than (>)" inequality of the type

$$b_1 n_1 + b_2 n_2 + \cdots + b_M n_M > N$$

a technique similar to the one for the "$\leq$-constraint" cannot be used because that would result in evaluating the contour integral over the unit circle of a function that has a pole at $z = 1$. To handle this constraint, we proceed as follows. Define the set of $\mathbf{n}$ satisfying the above constraint by $\mathcal{S}_{gt}(N, M)$. $G(\mathcal{S}_{gt}(N, M))$ be the sum of $\pi(\mathbf{n})$ over the set $\mathcal{S}_{gt}(N, M)$. Define $G_{tot}(N, M)$ as

$$G_{tot}(N, M) \;=\; \sum_{n_1=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} \prod_{i=0}^{M} f_i(n_i) \;=\; \prod_{i=0}^{M} \mathcal{F}_i(1)$$

It can be seen that

$$G(\mathcal{S}_{gt}(N, M)) = G_{tot}(N, M) - G(\mathcal{S}_{le}(N, M))$$

where $G(\mathcal{S}_{le}(N, M))$ is the sum of $\pi(\mathbf{n})$ over the set of $\mathbf{n}$ satisfying the constraint $\sum_{i=1}^{M} b_i n_i \leq N$. The integral formula for $G(\mathcal{S}_{gt}(N, M))$ is derived as follows

$$\begin{aligned} G(\mathcal{S}_{gt}(N, M)) \;&=\; \oint \frac{\prod_{i=1}^{M} \mathcal{F}_i(1)}{z} - \\ &- \left[\frac{z^{(N+1)} - 1}{z^{N+1}(z-1)} \right] \prod_{i=1}^{M} \mathcal{F}_i(z^{b_i})\, dz \end{aligned} \tag{6}$$

3 Multiple Constraints

We now describe the method for summing $\pi(\mathbf{n})$ over a state space defined by multiple constraints by extending the method developed in Section 2. Let $\mathbf{C}$ be the set of constraints and without loss of generality, let the first p of these be equality constraints and the next $(q-p)$ be inequality constraints of the *less than or equal to* type. They are defined as follows.

$$\begin{aligned} \sum_{j=1}^{M} b_{ij} n_j = N_i \qquad & \text{for } i = 1 \cdots p \\ \sum_{j=1}^{M} b_{ij} n_j \leq N_i \qquad & \text{for } i = p+1 \cdots q \end{aligned} \tag{7}$$

Let $\mathcal{S}(\mathbf{C}, M)$ the set of all vectors $\mathbf{n}$ satisfying the set of constraints in $\mathbf{C}$. Let $G(\mathcal{S}(\mathbf{C}, M))$ be defined as

$$G(\mathcal{S}(\mathbf{C}, M)) \equiv \sum_{\mathbf{n} \in \mathcal{S}(\mathbf{C}, M)} \prod_{i=1}^{M} f_i(n_i) \tag{8}$$

Define for $i = 1, \cdots, p$

$$\hat{\delta}_i \equiv \delta(b_{i1}n_1 + b_{i2}n_2 + \cdots + b_{iM}n_M - N_i)$$

and for $i = p+1, \cdots, q$

$$\hat{\Phi}_i \equiv \Phi_{N_i}(b_{i1}n_1 + b_{i2}n_2 + \cdots + b_{iM}n_M)$$

Note that $\hat{\delta}_i$ and $\hat{\Phi}_i$ can be written as

$$\begin{aligned}
\hat{\delta}_i &= \oint z_i^{(b_{i1}n_1+b_{i2}n_2+\cdots+b_{iM}n_M)} \left[\frac{1}{z_i^{N_i+1}}\right] dz \\
\hat{\Phi}_i &= \oint z_i^{(b_{i1}n_1+b_{i2}n_2+\cdots+b_{iM}n_M)} \left[\frac{z_i^{N_i+1}-1}{z_i^{N_i+1}(z_i-1)}\right] dz
\end{aligned}$$

$G(\mathcal{S}(\mathbf{C}, M))$ can now be derived as

$$\begin{aligned}
G(\mathbf{C}, M) &= \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C},M)} \prod_{i=1}^{M} f_i(n_i) \\
&= \sum_{n_1=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} \prod_{i=1}^{M} f_i(n_i)\hat{\delta}_1 \cdots \hat{\delta}_p \hat{\Phi}_{p+1} \cdots \hat{\Phi}_q \\
&= \oint \left[\frac{1}{z_1^{N_1+1}}\right] \cdots \oint \left[\frac{1}{z_p^{N_p+1}}\right] \\
&\quad \oint \left[\frac{z_p^{N_{p+1}+1}-1}{z_{p+1}^{N_{p+1}+1}(z_{p+1}-1)}\right] \cdots \oint \left[\frac{z_q^{N_q+1}-1}{z_q^{N_q+1}(z_q-1)}\right] \\
&\quad \prod_{i=1}^{M} \mathcal{F}_i(z_1^{b_{1i}} z_2^{b_{2i}} \cdots z_q^{b_{qi}}) dz_1 \cdots dz_q
\end{aligned} \tag{9}$$

In the above expression, the first p contour integrals correspond to the first p equality constraints. The subsequent $(q-p)$ integrals correspond to the *less than or equal to* type of constraints. The $\mathcal{F}_i(z)$ are the z-transforms of $f_i(n_i)$ and the contour integrals are evaluated over the unit circle.

4 Evaluation of the Contour Integrals

We now consider the evaluation of the contour integral. All integrals are evaluated on the unit circle. From the residue theorem, for any function of the complex variable z and a closed contour C in the complex plane,

$$\oint_C \mathcal{F}(z) = \sum \text{residues of } \mathcal{F}(z) \text{ at poles inside } \mathcal{C}$$

Our contour of integration is the unit circle. Note that if the $\mathcal{F}_i(z)$ have poles inside the unit circle, they can be suitably scaled to move them out of it. In Eqn. 9 the only poles of the integrand inside the unit circle corresponding to the i^{th} integration is at $z_i = 0$. The order of this pole is $(N_i + 1)$. Therefore, to evaluate the integrals in the expression for $G(\mathbf{C}, M)$ in Eqn. 9 we need to evaluate the residues of the functions at $z_i = 0$ $(i = 1 \cdots q)$. Thus, the evaluation of $G(\mathbf{C}, M)$ can be accomplished by the following algorithm

$G_0 = \prod_{i=1}^{M} \mathcal{F}_i(z_1^{b_{1i}} z_2^{b_{2i}} \cdots z_q^{b_{qi}})$;

for k **from** 1 **to** p **do**

$G_k =$ **residue** $\left(G_{k-1} * \left[\frac{1}{z_k^{N_k+1}}\right], z_k = 0\right)$;

for k **from** $p+1$ **to** q **do**

$G_k =$ **residue** $\left(G_{k-1} * \left[\frac{1}{z_k^{N_k+1}}\right] \left[\frac{z_k^{N_k+1}-1}{z_k-1}\right], z_k = 0\right)$;

$G(\mathbf{C}, M) = G_q$;

The evaluation of these residues involves only differentiation and taking the limit of the derivative as $z_i \to 0$. *i.e.* at the k^{th} step in the above algorithm, the residue is given by (see for example [12])

$$\text{residue}\left(G_{k-1} * \left[\frac{1}{z_k^{N_k+1}}\right], z_k = 0\right) = \lim_{z_k \to 0} \frac{1}{N_k!} \frac{d^{N_k}}{dz_k^{N_k}} [G_{k-1}] \quad \text{for } k = 1 \cdots p \tag{10}$$

$$\text{residue}\left(G_{k-1} * \left[\frac{1}{z_k^{N_k+1}}\right] \left[\frac{z_k^{N_k+1}-1}{z_k-1}\right], z_k = 0\right) = \lim_{z_k \to 0} \frac{1}{N_k!} \frac{d^{N_k}}{dz_k^{N_k}} \left[G_{k-1} * \left[\frac{z_k^{N_k+1} - 1}{z_k - 1}\right]\right] \quad \text{for } k = (p+1) \cdots q \tag{11}$$

The total number of differentiations at the k^{th} step in the above algorithm is N_k and the algorithm needs $\sum_{k=1}^{q} N_k$ differentiations to evaluate $G(\mathbf{C}, M)$. Note that left hand side of Eqns. 10 and 11 is the evaluation of the N_k^{th} coefficient of the Taylor series of G_{k-1} and $G_{k-1}(z_k^{N_k+1} - 1)/(z_k - 1)$ respectively.

If $\mathcal{F}_i(z)$ can be represented as a ratio of polynomials, then we can have a partial fraction expansion of G_0 and there are well known techniques to evaluate the residues in this case (see for example [13]).

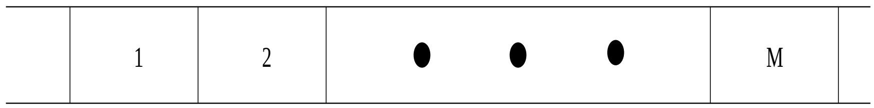

Figure 2: Cells in a highway cellular system

The evaluation of the residue in this situation is very efficient and can be shown to be independent of M and N_k and Eqn. 9 will have a closed form expression.

5 Application Examples

In this sections we consider three examples for the application of our technique. First, we obtain an analytical model to obtain the blocking probability in any cell of a finite highway cellular system. Next we use this technique to obtain an exact analysis of the blocking probability in the copy network of the Lee Multicast Switch. Finally, we consider application of this technique in solving a BCMP queuing network

5.1 Analysis of a Highway Cellular System

Consider a one dimensional highway cellular communication system with M cells and K channels. We assume that each cell, except the first and the last cell, has exactly two neighbors corresponding to the cells on the left and right as shown in Fig. 2. A channel that is being used in a cell is not available for use in that cell and its neighbors. If there are K channels available in the system, then an incoming call is accepted into the system if a channel can be found that is not being used in the cell or in any of its neighbors.

We consider a system that uses maximum packing strategy for channel assignment and has no hand off calls. We assume that the call arrival process to a cell is Poisson with rate λ and call duration is arbitrarily distributed with mean 1. Let the state of the system be denoted by the vector $\mathbf{n} = [n_1 \cdots n_M]$, where n_i is the total the number of active calls in cell i. The solution for the steady state probability for the state of the system has a product form and is [14]

$$\text{Prob}(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^{M} \frac{\lambda_i^{n_i}}{n_i!}$$

From [14], the constraints on the state space for this system can be written as

$$n_i + n_{i+1} \leq K \quad \text{for } i = 1 \cdots M-1 \tag{12}$$

An incoming call to an internal cell j is blocked if the total number of calls in cell j and at least in one of its neighbors, $j-1$ or $j+1$, is K. The set of states, $\mathbf{n}$, in this situation can be represented by additional boundary conditions

$$n_j + n_{j+1} = K \tag{13}$$

$$n_j + n_{j-1} = K \tag{14}$$

Denote by $\mathcal{S}_{B1}$ the set of all states, $\mathbf{n}$, that satisfy constraints (12) and (13), by $\mathcal{S}_{B2}$ the set of all states, $\mathbf{n}$, that satisfy constraints (12) and (14) and by $\mathcal{S}_{B12}$ the set of all states, $\mathbf{n}$, that satisfy constraints (12), (13) and (14). The set $\mathcal{S}_{B12}$ is the intersection of sets $\mathcal{S}_{B1}$ and $\mathcal{S}_{B2}$. The state $\mathbf{n}$ is a blocking state if $\mathbf{n}$ is contained in the union of the sets $\mathcal{S}_{B1}$ and $\mathcal{S}_{B2}$ as defined above. Thus, the probability $P_B(j)$, that a call arriving to an internal cell j is blocked is given by

$$P_B(j) = \text{Prob}\left(\mathbf{n} \in (\mathcal{S}_{B1} \cup \mathcal{S}_{B2})\right) = \frac{G_{B1} + G_{B2} - G_{B12}}{G}$$

where

$$\begin{aligned} G_{B1} &\equiv \sum_{\mathbf{n}\in\mathcal{S}_{B1}} \prod_{i=1}^{M} \frac{\lambda_i^{n_i}}{n_i!} \\ G_{B2} &\equiv \sum_{\mathbf{n}\in\mathcal{S}_{B2}} \prod_{i=1}^{M} \frac{\lambda_i^{n_i}}{n_i!} \\ G_{B12} &\equiv \sum_{\mathbf{n}\in\mathcal{S}_{B12}} \prod_{i=1}^{M} \frac{\lambda_i^{n_i}}{n_i!} \end{aligned}$$

From this, we can obtain the blocking probabilities in any cell for various values of λ, M, and K. We can use a symbolic computation package like Maple or Mathematica and using the method described in this paper, we can obtain the symbolic expression for the residues in $(M-1)$ steps of the algorithm of section 4. Once the residues are evaluated and the expression for the blocking probability derived, the blocking probabilities for any call arrival rate, λ, can be easily evaluated. The blocking probabilities in the middle cell for various values of M, K and λ are obtained using our algorithm and shown in Table 1. In our calculations we have assumed $\lambda_i = \lambda$ for all i.

The symbolic computation that may be used to derive the formula for the blocking probability also has a computational cost. To understand the "computational advantage" of our analytical technique, 1 Channel (K=1)

M↓ $\lambda \rightarrow$	0.1K	0.2K	0.3K	0.4K	0.5K	0.6K	0.7K	0.8K	0.9K	1.0K
3	0.237	0.390	0.497	0.576	0.636	0.684	0.721	0.752	0.778	0.800
5	0.224	0.359	0.448	0.513	0.561	0.598	0.629	0.654	0.674	0.692
11	0.225	0.359	0.447	0.509	0.555	0.590	0.618	0.640	0.659	0.673
15	0.225	0.359	0.447	0.509	0.555	0.590	0.618	0.640	0.659	0.673
21	0.225	0.359	0.447	0.509	0.555	0.590	0.618	0.640	0.658	0.674

2 Channels (K=2)

M↓ $\lambda \rightarrow$	0.1K	0.2K	0.3K	0.4K	0.5K	0.6K	0.7K	0.8K	0.9K	1.0K
3	0.087	0.227	0.353	0.454	0.535	0.599	0.650	0.692	0.727	0.756
5	0.082	0.205	0.308	0.389	0.452	0.502	0.543	0.577	0.604	0.629
11	0.082	0.204	0.305	0.383	0.441	0.485	0.520	0.547	0.569	0.587
15	0.082	0.204	0.305	0.383	0.441	0.485	0.520	0.547	0.569	0.587
21	0.082	0.204	0.305	0.383	0.441	0.485	0.520	0.547	0.569	0.587

5 Channels (K=5)

M↓ $\lambda \rightarrow$	0.1K	0.2K	0.3K	0.4K	0.5K	0.6K	0.7K	0.8K	0.9K	1.0K
3	0.006	0.062	0.171	0.291	0.397	0.485	0.557	0.615	0.662	0.701
5	0.005	0.056	0.148	0.241	0.322	0.388	0.442	0.487	0.525	0.556
11	0.005	0.056	0.147	0.236	0.310	0.367	0.410	0.443	0.468	0.488
15	0.005	0.056	0.147	0.236	0.310	0.367	0.410	0.443	0.468	0.489
21	0.005	0.056	0.147	0.236	0.310	0.367	0.410	0.443	0.468	0.489

Table 1: Blocking Probabilities for the Highway Cellular Communication System

note that computation is involved in symbolically obtaining the residues in the $(M-1)$ steps of the algorithm of section 4. Once the residues are evaluated and the expression for the blocking probability derived, the blocking probabilities for any call arrival rate, λ, can be easily evaluated. We evaluated the blocking probability using a Maple program implementing our algorithm and a C program implementing the brute force evaluation technique (no other algorithmic technique is known) on a Sun Ultra 10 with 333 MHz UltraSparc IIi processor and 2MB cache. The execution times represented by the sum of the user and system times as reported by the Unix *time* command for the formula (Maple program) and the brute force technique (C program) are shown in Table 2. For our algorithm, we report separately the time and number of differentiations, $(M-1)K$, required to obtain the formula (which must be done once for each system) and the time to evaluate P_B for ten values of λ (ranging from $0.1K$ to K in steps of $0.1K$) from the formula. The portion of the code used to symbolically compute G, G_B, G_{B1} and G_{B2} is given in the appendix. For the brute force enumeration approach, we give the total size of the (unconstrained) state space ($(K+1)^M$) and the execution time to obtain P_B for the same set of λ. In the C program, every possible vector $\mathbf{n}$, $(K+1)^M$ combinations, is checked to see if it belongs to the set of admissible states. If it does, the product form expression corresponding to this state is computed and added to obtain normalizing constant and also to obtain the numerator of the blocking probability. Obviously for a small number of states the brute force technique is fast. The computational disadvantage begins to become significant when the number of states to check is of the order of a few tens of thousand.

5.2 Analysis of Copy Networks of Multicast Switches

We now apply the techniques discussed in this paper to the analysis of a copy network of the kind described in Section 1. Recall that in any slot, the sum of the number of copies requested by the active inputs, inputs with requests, may exceed N and in the model that we analyze here the first i inputs that satisfy the conditions $\sum_{j=1}^{i} c_j \leq N$ and $\sum_{j=1}^{i+1} c_j > N$ are served and those requests that cannot be served are lost. In the following we show how to calculate the probability that a request from input i is lost. If the requests that cannot be served are queued to be served in a subsequent slot we will also show how to do a queuing analysis using the techniques described in this paper.

Packet arrivals to port i is a Bernoulli process with rate ρ_i and the copy requests have a probability mass function $q_i(k)$. Let X_i be the number of copies requested by the input port i, $f_i(x_i)$ its probability

M,K	Num of Diffns	Time for Formula	Time for for P_B	Num of States	Time for Brute Force
11,1	10	0.25s	0.13s	2,048	0.02s
5,5	20	0.32s	0.14s	7,776	0.12s
15,1	14	0.29s	0.14s	32,768	0.25s
11,2	20	0.36s	0.13s	177,147	1.20s
21,1	20	0.36s	0.14s	2,097,152	6.19s
15,2	28	0.63s	0.14s	14,348,907	43.49s
11,5	50	7.07s	0.17s	362,797,056	1068.31s
21,2	40	1.57s	0.16s	10,460,353,203	9465.29s
15,5	70	42.85s	0.22s	4.701849E+11	523688.86s
21,5	100	343.22s	0.31s	2.193695E+16	too big

Table 2: Comparison of the time taken by our algorithm and the brute force technique to obtain the blocking probabilities in the highway cellular communication system

mass function and $\mathcal{F}_i(z)$ the moment generating function of $f_i(x_i)$. Then,

$$f_i(x_i) \equiv \Pr\{X_i = x_i\} = \begin{cases} 1-\rho_i, & x_i = 0 \\ \rho_i q_i(x_i), & x_i = 1, \cdots N \end{cases} \tag{15}$$

The copy request of input i is served if $X_1 + X_2 + \cdots + X_i \leq N$. The probability of loss at port i, $P_{loss}(i)$, is then given by

$$P_{loss}(i) = 1 - \sum_{\sum_{j=1}^{i} x_j \leq N} \prod_{j=1}^{i} f_j(x_j)$$

The summation on the RHS of the above equation is carried out over all possible combinations of copy requests from ports 1 to i that sum to less than or equal to N. Therefore, following Section 2.2, we can obtain the $P_{loss}(i)$s as follows

$$\begin{aligned} 1 - P_{loss}(i) &= \sum_{x_1=0}^{N} \cdots \sum_{x_i=0}^{N} \prod_{k=1}^{i} f_k(x_k) \Phi_N(x_1 + \cdots x_i) \\ &= \oint \left[\frac{z^{(N+1)} - 1}{z^{(N+1)}(z-1)} \right] \prod_{k=1}^{i} \mathcal{F}_i(z) dz \end{aligned}$$

Figure 3 compares the exact P_{loss} that we obtain here with P_{Ch} that Lee obtains in [7].

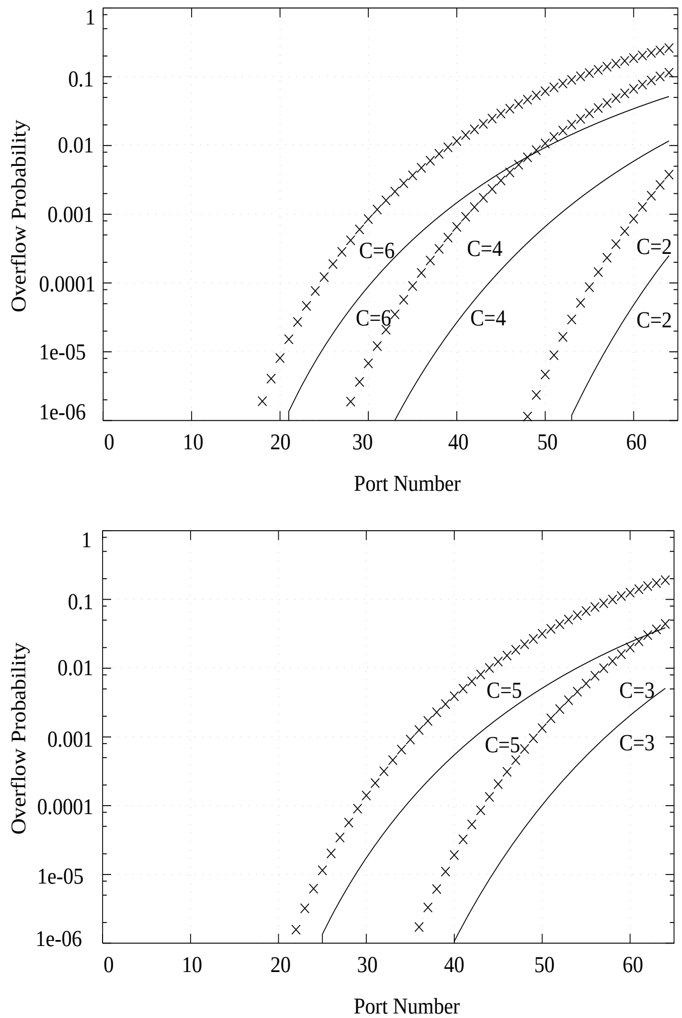

Figure 3: Overflow probabilities in a 64×64 copy network with deterministic copy requests of size 2,3,4,5 and 6. The broken lines denote the Chernoff bounds while the smooth lines represent the exact results.

Now consider the case when the requests that cannot be served are queued. Although many scheduling policies can be formulated and analyzed, to illustrate the use of our technique, we will consider the simplest scheduling policy in which the requests satisfying the conditions mentioned earlier are served queued and the others are queued. Proceeding sequentially from input 1, all the input ports whose copy requests can fully served in the slot are selected for service. Note that this policy selects a packet for service only if all the copies requested by it can be generated in the given slot. Since service always starts from port 1, the service rate varies with the port number and decreases as the port address increases.

The effective service rate at an input port, the rate at which the copy requests can be actually served, depends on the arrival processes and effective service rates at the preceding ports. We model each input port as a discrete time M/M/1 queue except for the first port which always gets to be served in every slot. At port i, let the effective service rate and the probability of its input queue being empty be denoted by $\mu(i)$ and $P_{0,i}$ respectively. The number of ports that can be served in a slot depends on the copy requests of the packets at the head of the queues. Let $f_{H,i}(k)$ be the probability mass function (pmf) of the number of copies requested by the packet at the head of input queue i. From our definition above, $1 - P_{0,i}$ is the probability that the head of queue i is non empty. Hence the probability mass function of the copies requested by a packet at the head of the queue i will be,

$$f_{H,i}(x_i) \equiv \text{Prob}\left\{X_i = x_i\right\} = \begin{cases} P_{0,i} & x_i = 0 \\ (1 - P_{0,i})\, q(x_i) & x_i > 0 \end{cases}$$

with $\mathcal{F}_{H,i}(z)$ as its moment generating function. Since port 1 is always served in each slot irrespective of the copy requests of the other ports, $\mu(1) = 1.0$. Now, for ports $i = 2, \cdots M$, if port i requests k copies, its request will be served in the slot only if the sum of the copies requested by ports 1 to $i-1$ is less than or equal to $N-k$. Thus,

$$\begin{aligned} \mu(i+1) &= \text{Prob}\left\{\text{a pkt at port } i+1 \text{ is served }\right\} \\ &= \sum_{k=1}^{N} q(k)\ \text{Prob}\left\{\sum_{j=1}^{i} X_j \leq N-k\right\} \\ &= \sum_{k=1}^{N} q(k)\left[\sum_{\sum_{j=1}^{i} x_j \leq (N-k)} \prod_{j=1}^{i} f_{H,j}(x_j)\right] \end{aligned}$$

Using the results of Section 2.2, it can be shown that $\mu(i+1)$ is given by

$$\mu(i+1) = \sum_{k=1}^{N} q(k) \oint \left[\frac{z^{(N-k+1)} - 1}{z^{N-k+1}(z-1)}\right] \prod_{j=1}^{i} \mathcal{F}_{H,j}(z) dz$$

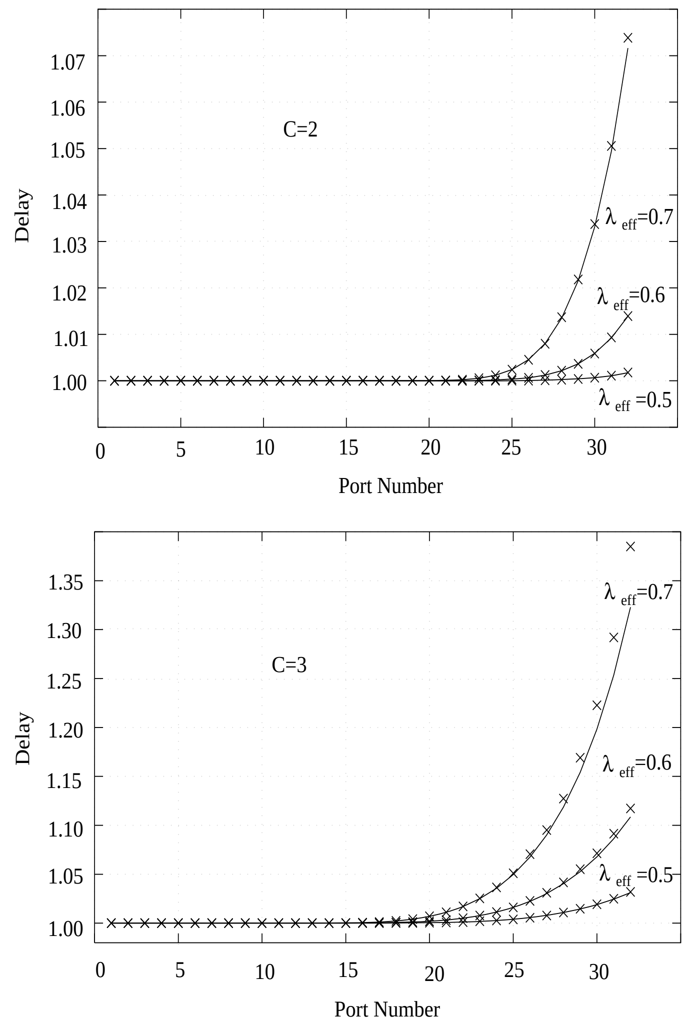

Figure 4: Delay characteristics of a 32×32 copy network for deterministic copy requests.

Using well known results for discrete time M/M/1 queues, (see, for example [16]), we can calculate the waiting time moments and other parameters. In Fig. 4 we show the mean delay for active requests at various inputs of a 32×32 copy network when the number of copies requested by an active input is deterministic. These results are obtained using the method outlined above. Also shown are the results from a simulation model.

We can also use the method described in this paper to analyze various other scheduling policies for the copying process in the copy network [15].

5.3 BCMP Networks

The product form solution has been extended to a very general class of queuing networks by Baskett et al in [17]. Such queuing networks are called BCMP networks in literature. Four types of service centers and multiple classes of jobs, each with a different service requirement and routing probabilities, were permitted. We assume that there are M nodes in the queuing network and that there are R classes of jobs. We denote by n_{ir} the number of class r jobs in node i. The state of node i will be denoted by $\mathbf{y_i}$ where

$$\mathbf{y_i} = [n_{i1}, n_{i2} \cdots n_{iR}]$$

We denote the state of the queuing network by the vector $\mathbf{Y}$ which is defined to be

$$\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2} \cdots \mathbf{y_M}]$$

In this section, we will describe a technique, similar to the one used for single class networks in section 3, to derive the normalizing constant. As before, we will assume that the constraints on the state space for the system are linear equalities and inequalities, except that instead of n_i, we will have n_{ir} as the variables.

$$\begin{aligned} \sum_{i=1}^{M}\sum_{r=1}^{R} b_{j,ir} n_{ir} &= N_j \qquad \text{for } j = 1 \cdots p \\ \sum_{i=1}^{M}\sum_{r=1}^{R} b_{j,ir} n_{ir} &\leq N_j \qquad \text{for } j = p+1 \cdots q \end{aligned}$$

Let $\mathbf{C}$ be the above set of constraints and $\mathcal{S}(\mathbf{C}, M, R)$ the set of $\mathbf{Y}$ satisfying these constraints. Without loss of generality, we will assume that the nodes $1 \cdots \hat{M}$ are first come first serve (FCFS), processor sharing (PS) or last come first serve (LCFS) queues and nodes $\hat{M}+1 \cdots M$ are infinite server (IS) queues. From [18], if we define $\hat{f}_i(\mathbf{y_i})$ to be

$$\hat{f}_i(\mathbf{y_i}) \equiv \begin{cases} n_i! \prod_{r=1}^{R} f_{ir}(n_{ir}) & \text{for } i = 1 \cdots \hat{M} \\ \\ \prod_{r=1}^{R} f_{ir}(n_{ir}) & \text{for } i = \hat{M}+1 \cdots M \end{cases}$$

the steady state probability of the system will be

$$\text{Prob}(\mathbf{Y}) \quad = \quad \frac{1}{G_{bcmp}(\mathbf{C}, M, R)} \prod_{i=1}^{M} \hat{f}_i(\mathbf{y_i})$$

$$= \frac{1}{G_{bcmp}(\mathbf{C}, M, R)} \left[\prod_{i=1}^{\hat{M}} n_i! \prod_{r=1}^{R} f_{ir}(n_{ir}) \right]$$
$$\left[\prod_{i=\hat{M}+1}^{M} \prod_{r=1}^{R} f_{ir}(n_{ir}) \right]$$

where $n_i = \sum_{r=1}^{R} n_{ir}$ and $G_{bcmp}(\mathbf{C}, M, R)$ is the normalizing constant for the network obtained as follows

$$G_{bcmp}(\mathbf{C}, M, R) = \sum_{\mathbf{Y} \in \mathcal{S}(\mathbf{C}, M, R)} \left[\prod_{i=1}^{\hat{M}} n_i! \prod_{r=1}^{R} f_{ir}(n_{ir}) \right] \left[\prod_{i=\hat{M}+1}^{M} \prod_{r=1}^{R} f_{ir}(n_{ir}) \right] \tag{16}$$

Notice that this is similar to the normalizing constant of single class networks except that there are MR terms in the product rather than M and there is an additional $n_i!$ term for each non-IS node. To simplify this, consider the Euler integral

$$n! = \int_0^\infty e^{-t} t^n dt$$

Substituting this for $n_i!$ in Eqn. 16, we get

$$G_{bcmp}(\mathbf{C}, M, R) =$$
$$\sum_{\mathbf{Y} \in \mathcal{S}(\mathbf{C}, M, R)} \left[\prod_{i=1}^{\hat{M}} \int_0^\infty e^{-t} \, t^{\sum_{r=1}^{R} n_{ir}} dt \prod_{r=1}^{R} f_{ir}(n_{ir}) \right] \left[\prod_{i=\hat{M}+1}^{M} \prod_{r=1}^{R} f_{ir}(n_{ir}) \right]$$
$$= \sum_{\mathbf{Y} \in \mathcal{S}(\mathbf{C}, M, R)} \left[\prod_{i=1}^{\hat{M}} \int_0^\infty e^{-t} \prod_{r=1}^{R} t^{n_{ir}} f_{ir}(n_{ir}) \, dt \right] \left[\prod_{i=\hat{M}+1}^{M} \prod_{r=1}^{R} f_{ir}(n_{ir}) \right] \tag{17}$$

Eqn. 17 is similar to Eqn. 8 and the techniques developed in section 3 can be used. The only difference is that there is an additional integration for every node that is not a IS queue. Using the same technique that was used in the derivation of Eqn. 9 we obtain

$$G_{bcmp}(\mathbf{C}, M, R) =$$

$$
\begin{aligned}
&\oint \left[\frac{1}{z_1^{N_1+1}}\right] \cdots \oint \left[\frac{1}{z_p^{N_p+1}}\right] \oint \left[\frac{z_p^{N_{p+1}+1} - 1}{z_{p+1}^{N_{p+1}+1}(z_{p+1}-1)}\right] \\
&\cdots \oint \left[\frac{1}{z_q^{N_q+1}}\right] \left[\frac{z_q^{N_q+1} - 1}{z_q - 1}\right] \\
&\left[\prod_{i=1}^{\hat{M}} \int_0^\infty e^{-t} \prod_{r=1}^{R} \mathcal{F}_{ir}(t z_1^{b_{1,ir}} \cdots z_q^{b_{q,ir}})\, dt\right] \\
&\left[\prod_{i=\hat{M}+1}^{M} \prod_{r=1}^{R} \mathcal{F}_{ir}(z_1^{b_{1,ir}} \cdots z_q^{b_{q,ir}})\right] dz_1 \cdots dz_q
\end{aligned} \tag{18}
$$

where $\mathcal{F}_{ir}(z)$ is the z-transform of $f_{ir}(n_{ir})$. Here, we have used the property that the z-transform of $a^n f(n)$ is $\mathcal{F}(az)$.

6 Discussion and Conclusion

In this paper we have developed a method to obtain analytical expression for the sum of a product form expression over an irregular $M-$dimensional integer space defined by multiple linear constraints with integer coefficients. We use $z-$transforms and contour integrals which in turn reduce to evaluating residues of a function of complex variables. In most cases this can be done manually, but with the availability of symbolic computation the formulae can be obtained "computationally". The method that we develop here can be used in many situations like for example in calculating the normalizing constant in product form queuing networks or in models that deal with vectors of independent integer random variables like the one on copy networks described here. Our technique is reasonably general and does not assume any specific form for $f_i(n_i)$, the terms in the product form expression or the shape of the state space like many of the algorithms that are currently available.

The examples demonstrate various aspects of the usefulness of our method. In the highway cellular system we show that with increasing number of cells and channels, defining the state space is quite difficult. Employing brute force techniques in the numerical evaluation of system performance measures can be quite time consuming, possibly impossible, for even small systems of about 21 cells and 5 channels. Our technique can yield the measures fairly quickly and efficiently. In the case of the copy network we demonstrate the utility of our technique for arbitrary $f_i(n_i)$ with the state space constrained by multiple, in this case two, constraints. We also extended our technique to obtaining analytical expressions for a general BCMP network. Note that the results of [8, 9] can be considered to be special cases of Eqn. 4 and the result from [10] can be considered to be a special case of Eqn. 18

with one equality constraint and no infinite server nodes.

Finally, we mention that Nelson [19] discusses many other models, such as genetic and polymerization models, which have a solution which also need a summing of a product form expression over an irregular multidimensional integer space. The method reported in this paper can be applied to all such systems.

A Sample Maple Program

We reproduce here the part of the Maple program that was used to calculate the call blocking probability in the highway cellular communication system.

Note that we have used $z[i]$ for z_i.

```
readlib(coeftayl):
#G0 is the product of the z-transforms of the first step of the algorithm
#m is the middle cell for which blocking probability is being calculated.
G0:=exp(lambda*z[1]):
for i from 2 to M-1 do G0:=G0*exp(lambda*z[i-1]*z[i]):  od:
G0:=G0*exp(lambda*z[M-1]):

#Multiplier for the ``less than or equal'' constraint
le_mul:=proc(z):   ((z^(K+1)-1)/(z-1)):   end:

#This set of contour integrations is common to G, GB1, GB2, GB12
G_pre:=G0:
for i from 1 to M-1 do      # corresponding to M-1 constraint eqns
  if ((i<>(m-1))and (i<>m)) then G_pre:=coeftayl(G_pre*le_mul(z[i]),z[i]=0,K): fi:
  od:

#We obtain G in this step with two additional integrations
G_new:=coeftayl(G_pre*le_mul(z[m-1]),z[m-1]=0,K):

#Previous step is common to GB1. Preserve G_new to use for GB1
G:=coeftayl(G_new*le_mul(z[m]),z[m]=0,K):
```

```
#Additional integration on G_new, saved before gives GB1
GB1:=coeftayl(G_new,z[m]=0,K):

#Because of symmetry GB2=GB1
GB2:=GB1:

#Now we obtain GB12 from G_pre another two integrations
G_new:=coeftayl(G_pre, z[m-1]=0, K):
GB12:=coeftayl(G_new,z[m]=0,K):

# Blocking probability is evaluated
PB:=((GB1+GB2-GB12)/G):
```

References

[1] J. P. Buzen, Computational Algorithms for Closed Queuing Networks with Exponential Servers, Communications of the ACM 16(1973) 527-531.

[2] Keith W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks,* Springer, Berlin, 1995.

[3] J. Kaufman, Blocking in a shared Resource Environment, IEEE Transactions on Communications, COM-29(1981) 1474–1481.

[4] G. J. Foschini and B. Gopinath, Sharing Memory Optimally, IEEE Transactions on Communications COM-31(1983) 352–360.

[5] S. S. Lam, Queuing Networks with Population Size Constraints, IBM Journal of Research and Development 21(1977) 370-378.

[6] D. E. Everitt, Traffic Engineering of the Radio Interface for Cellular Mobile Networks, Proceedings of the IEEE 82(1994) 1371-82.

[7] T. T. Lee, Nonblocking Copy Networks for Multicast Packet Switching, IEEE Journal on Selected Areas of Commun, 6(1988), 1455-1467.

[8] P. G. Harrison, On Normalizing Constants in Queuing Networks, Operations Research, 33(1985) 464-468.

[9] J. J. Gordon, The Evaluation of Normalizing Constants in Closed Queuing Networks, Operations Research 38(1990) 863-869.

[10] A. I. Gerasimov, Integral Method for Calculating Closed Queuing Networks, Problems of Information Transmission 28(1992) pp 184-194.

[11] F. P. Kelly, *Reversibility and Stochastic Networks*, Wiley, Chichester, 1979.

[12] L. A. Rubenfeld, *A First Course in Applied Complex Variables* (Wiley, 1985).

[13] J. J. D'Azzo and C. H. Houpis, *Linear Control System: Analysis and Design* (McGraw-Hill Book Company, 1981).

[14] D. E. Everitt and N. W. MacFayden, Analysis of Multicellular Mobile Radio Telephone Systems with Loss, British Telecom Journal, 1(1983) 37-45.

[15] B. Sikdar, Queuing Analysis of Scheduling Policies in Copy Networks of Space Based Multicast ATM Switches, Master's Thesis, Dept. of Electrical Engineering, Indian Institute of Technology, Kanpur, 1998.

[16] M. E. Woodward, *Communication and Computer Networks : Modelling with Discrete-Time Queues* (Pentech Press, London, 1993).

[17] F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacios, Open, Closed and Mixed Networks of Queues with Different Classes of Customers, Journal of the ACM 22(1975) 248-260.

[18] E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems* (Academic Press, 1980).

[19] R. Nelson, The Mathematics of Product Form Queuing Networks, ACM Computing Surveys 25(1993) 339–369.

[20] Y.-S. Yeh, M. G. Hluchyj and A. S. Acampora, "The Knockout Switch: A Simple Modular Architecture for High Performance Packet Switch," *IEEE Journal on Selected Areas in Communication,* vol SAC-5, no 8, 1987, pp 1274–1283.