TECHNICAL RESEARCH REPORT

Performance Analysis and Multi-Objective Design for Multirate Multihop Loss Networks

by Mingyan Liu, John S. Baras

CSHCN T.R. 99-22
(ISR T.R. 99-45)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Performance Analysis and Multi-Objective Design for Multirate Multihop Loss Networks

Mingyan Liu, John S. Baras
Center for Satellite and Hybrid Communications Networks
University of Maryland
College Park, MD 20742 *

Abstract

In this paper, we consider a class of loss networks where multiple traffic classes are present, each has different bandwidth requirement, and each traffic stream is routed according to an adaptive routing scheme. We propose a fixed-point method, a.k.a. reduced load approximation, to estimate the end-to-end blocking probability for such networks. The approximation scheme is shown to be asymptotically correct in a natural limiting regime, and it gives conservative estimates of blocking probabilities under heavy traffic load. Simulation results are provided to compare performance estimates obtained from our analytical approximation scheme and discrete event simulations. We also show how this analytical approximation scheme can be linked with numerical mathematical programming tools to help design a network, by selecting network design parameters via trade-off analysis, even with several design objectives. In one application we use the multi-objective optimization tool CONSOL-OPTCAD to design trunk reservation parameters and balance link capacity. In another application we use automatic differentiation to get sensitivities of blocking probabilities w.r.t. offered traffic load.

keywords

Fixed point method, Loss network, Network optimization and design.

*This work was supported by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC3-528

1 Introduction

This paper is focused on the evaluation of end-to-end call blocking probabilities in a class of loss networks, as well as applying such evaluation to network design. A loss network is a circuit switched network, where a call requires a fixed amount of bandwidth on every link on a path between the source and destination. A more detailed definition of loss networks can be found in [1]. If the network has the required bandwidth on those links when the request arrives, the call is admitted and it will be holding the requested capacities for some time; otherwise the call is rejected. The blocking probability associated with a loss network is the probability that a call finds the network unavailable when it arrives and is thus rejected. A telephone system is a typical loss network. An ATM network can also be viewed as a loss network and the connection level blocking probabilities can be calculated by applying the concept of effective bandwidth [2],[3].

The Erlang formula:

$$E(\nu, C) = \frac{\nu^C}{C!}[\sum_{n=0}^{C} \frac{\nu^n}{n!}]^{-1}, \tag{1}$$

established the loss probability of a single link with C units of bandwidth when calls arrive as a single Poisson process with rate ν. When a fixed route is associated with each source-destination node pair, a loss network can be modeled as a multidimensional Markov process with the dimension of the process state space being the product of the number of routes allowed in the network and the number of service/call classes. When alternative routes are present in addition to fixed routes, the Markov process no longer has a product form, and solving it exactly [1] is not practical, when dealing with large networks with hundreds of thousands of routes and integrated services with multiple service rates, since the computational complexity is both exponential in the number of routes and exponential in the number of service classes. This leads to the need for the development of computational techniques that provide accurate estimates within reasonable time frame.

The reduced load approximation, also called the Erlang fixed-point method, has been proposed for this scenario and has been studied intensively [1],[4],[5],[6]. It is based on two assumptions:

(a) *link independence assumption*: Blocking occurs independently from link to link.

(b) *Poisson assumption*: Traffic flow to each individual link is Poisson and the corresponding traffic rate is the original external offered rate thinned by blocking on other links on the path, thus called the reduced load.

Most of the earlier works in fixed-point method either studied the multirate traffic situation with fixed routing, such as the Knapsack approximation and Pascal approximation in [5], or focused on state-dependent routing schemes with single traffic rate [4], or multirate service with single link (resource) [7]. In [6] Greenberg and Srikant proposed a fixed-point

method to approximate blocking probabilities in a multirate multihop network using sequential routing, but additional computational effort in solving the associated network reliability problem is needed. It is also a common assumption that there exist a direct or two-hop routes between sources and destinations.

We focus our attention on the evolving integrated service networks which have the following characteristics:

(a) The networks are typically much sparser and have a more hierarchical topology. Thus, the assumption of the existence of a direct route between source and destination nodes does not hold in most instances.

(b) Routes can comprise a much larger number of hops (typically around 5 or 6) and there are typically a large number of possible routes between source and destination nodes.

(c) The presence of different traffic classes characterized by widely varying bandwidth requirements and different mean holding times must be considered.

Motivated by the above, we propose to use adaptive routing in combination with the fixed-point method to calculate call blocking probabilities. Most of our work is motivated by [6] and [8]. One of the main reasons we are interested in developing an analytical approximation algorithm is to use it for network design and optimization. Mitra etc. [3],[9],[10] has developed an approximation scheme for VPN design based on loss network models. In this paper we link our algorithm to mathematical programming tools to get trade-offs on load balancing, resource allocation and call admission control.

The organization of the paper is as follows: The next section describes the network model and the adaptive routing schemes proposed for the approximation. Section III is the proposed fixed-point approximation method. Asymptotic correctness analysis is given out in Section IV and Section V we present approximation results compared to simulation results. In Section VI and VII we give two applications of linking the approximation scheme to the optimization tool CONSOL-OPTCAD and the automatic differentiation package ADIC, respectively. Section VIII concludes the paper.

2 Network Model

Consider a network with N nodes and J links, each indexed by j. C_j denotes the capacity of link j, in unit bandwidth/circuit. R is the set of all node pairs, each indexed by r. The total number of node pairs is thus $N(N-1)/2$. For each node pair r, there is an associated set of M ordered routes, each indexed by m, representing the m^{th} route in that set. Therefore, pair (r, m) uniquely defines a specific route. The network supports a total of S classes of traffic, indexed by s, and thus (r, s) uniquely defines a specific incoming call request. The

bandwidth requirement of such a call on link j is denoted by b_{js}. Note that for different node pairs the classification of calls does not have to be the same. So strictly speaking calls (r_1, s) and (r_2, s) can have different bandwidth requirement on a same link if we allow r_1 and r_2 to have different sets of traffic classes. However, we choose to use this notation because a single uniform classification can always be achieved by increasing the number of classes.

Calls arrive at the network as Poisson processes with an offered load λ_{rs}. A call is accepted if some route has available bandwidth on each of its links to accommodate this call, and the call is routed on that route and holds the bandwidth for a duration with mean time μ_{rs}. If none of the routes are available, the call is rejected. The end-to-end blocking probability of a call (r, s) is denoted by B_{rs}. Throughout this paper the links are considered to be duplex and bi-directional. We use trunks, units of circuits and units of bandwidth interchangeably.

The type of call admission control considered in our model is *trunk reservation*. As Kelly pointed out in [1], if alternative routes use more network resources than first-choice routes, then allowing a blocked call to attempt an alternative route may actually increase the loss probability of a network, and this effect may become even more pronounced if a blocked call can attempt a sequence of alternative routes. An explanation for this phenomenon is that if a link accepts an alternatively routed call, it may later have to block a directly routed call which will then attempt to find a two-hop or multihop route elsewhere in the network. A natural response would be for the link to reject an alternatively routed call if the free circuits on the link are below a certain level. This is the call admission control of the trunk reservation type. An attempted alternatively routed call is only accepted if on each link of the alternative route the number of occupied circuits is less than $C_j - b_{js} - r_s$ where r_s is the *trunk reservation parameter*, which may vary with link and class.

The common routing policies which have been studied are fixed routing, alternative routing, sequential alternative routing and adaptive alternative routing. We focus on the last. One important scheme of this kind is called the *Least Loaded Routing* (LLR), where the call is first tried on the direct route, if there is one. If it cannot be setup along the direct route, the two-link alternative route with the largest number of point-to-point free circuits is chosen. A version of LLR was implemented in the AT&T long-distance domestic network [1].

A direct extension of LLR to networks where routes tend to have a larger number of hops instead of direct or two-hop routes is a min-max scheme: Pick the link which has the minimum free bandwidth for each route, then pick the route which has the maximum free bandwidth on this link.

Each source-destination node pair r is given a list of alternative routes M_r. When a call arrives, each route on the list is evaluated to determine the number of free circuits on its links.

Let C_j^f denote the free/available bandwidth on link j when the call of type (r, s) arrives

and consider a route (r, m). Then the route is in a state of admitting this call if and only if

$$C_j^f \geq b_{js}, \text{for all} j \in (r, m)$$

under no trunk reservation admission control, or

$$C_j^f \geq b_{js} + r_s, \text{for all} j \in (r, m)$$

under trunk reservation admission control.

Consider a route (r, m) which is presently available for a type (r, s) call. The *most congested link on the route* is defined as the link with the fewest free circuits on this route:

$$\mathcal{L}_{rm} = argmin_{j \in (r,m)} C_j^f. \tag{2}$$

When there are more than one routes available in the alternative route set, the one with the maximum free bandwidth on its most congested link is selected for accepting the call. If none of the routes are admissible, then the call is blocked.

This maximal residual capacity routing scheme tries to avoid bottlenecks on a route. However, while choosing the route which has the most free bandwidth, we might end up taking the longer or the longest routes in the available set and thus using more network resources. This could eventually force calls arriving later to be routed on their longer/longest route as well. Therefore, using trunk reservation along with this routing scheme is a valid choice especially when traffic is heavy.

A more general way of deciding the routing can be expressed as a cost function which takes into account both the length of the route and the congestion level of the route. (For optimization on routing and blocking, Mitra and Morrison present an elaborate form of network revenue in [3] and investigate a network optimization problem. Our focus here is limited to admission control.) A simple cost function for route (r, m) could be:

$$w_1 \sum_{j \in (r,m)} b_{js} - w_2 C_{\mathcal{L}_{rm}}^f \tag{3}$$

where the first term is the total bandwidth that would be occupied if the route is chosen, and the second term indicates the level of congestion on the route. w_1 and w_2 are weighting parameters. The route which minimizes this cost is chosen. Clearly a longer route will increase the cost. And if w_1 is zero, this becomes the min-max routing we just described.

3 The Fixed Point Method

The fixed point is achieved by mappings between the following four sets of unknown variables:

ν_{js}: the reduced load/arrival rate of class-s calls on link j;

a_{js}: the probability that link j is in a state of admitting class-s calls;

$p_j(n)$: the steady state occupancy probability distribution of link j, i.e., the probability that exactly n units of circuits are being used on link j. n takes any integer value between 0 and C_j, the capacity of link j;

q_{rms}: the probability that a call request (r, s) is attempted on route (r, m). This originates from the fact that different routes have different levels of congestion.

First, we fix a_{js} and q_{rms} to get ν_{js}. Then we let ν_{js} be fixed to get $p_j(n)$ and a_{js}. Finally we fix $p_j(n)$ to get q_{rms}. By repeated substitution, the equilibrium fixed point can be solved for all four sets of unknowns. The mappings are illustrated in the figure bellow:

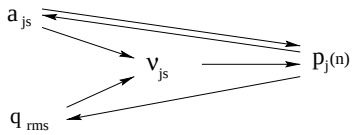

Figure 1: Mappings between variables.

3.1 Mapping 1: a_{js}, $q_{rms} \longrightarrow \nu_{js}$

Define ν_{jrms} as the arrival rate on link j contributed by traffic (r, s) on route (r, m), given that link j is in a state of admitting a class-s call. Then it is given by the reduced load approximation as:

$$\nu_{jrms} = \lambda_{rs} q_{rms} I[j \in (r, m)] \prod_{i \in (r,m), i \neq j} a_{is} \tag{4}$$

where I is the indicator function. The aggregated load of class-s calls on link j is given by

$$\nu_{js} = \sum_{(r,m)} \nu_{jrms}. \tag{5}$$

3.2 Mapping 2: $\nu_{js} \longrightarrow a_{js}$, $p_j(n)$

Given ν_{js}, we can compute the link occupancy probabilities $p_j(n)$ for each link in the network. This can be done by either using Kaufman's simple recursion [11] when there is no trunk reservation present, or using approaches proposed by Bean, Gibbens and Zachary in [12] and [7] as suggested by Greenberg in [6]. By the link independence assumption, this mapping is conducted on a per-link basis, and each link is calculated separately and similarly.

In the absence of admission control, classical product-form holds for describing the equilibrium call blocking probabilities [1]. In [11], Kaufman gives a simple one dimensional recursion for calculating the link occupancy distribution probabilities.

The probability that a class-s call is admitted to link j is given by

$$a_{js} = 1 - \sum_{n=C_j-b_{js}+1}^{C_j} p_j(n) = \sum_{n=0}^{C_j-b_{js}} p_j(n). \tag{6}$$

Admission control destroys the product form of the link occupancy probabilities $p_j(n)$, which in turn destroys the efficient exact computation of those probabilities in [11]. To solve for these probabilities, we need to solve for the equilibrium distribution of the associated Markov chain, whose state space is a lattice embedded in the simplex $\sum_s b_{js} n_s \leq C,\ n_s \geq 0$. The computational cost is prohibitive, even for moderate C and $S = 2$. A method to compute the aggregated occupancy probabilities $p(n)$ at a cost linear in C is needed. Approaches proposed in [12],[7] transform the problem into a one-dimensional one by assuming that while n_s, the number of calls in progress of a class s varies, the proportion of such calls in progress remains fixed (or varies slowly).

In [6] the following method is used. Let α_{js} denote the average number of calls of type s in progress on link j,

$$\alpha_{js} = a_{js}\nu_{js}/\mu_{rs}, \tag{7}$$

since calls enter into service at rate $a_{js}\nu_{js}$ and depart at rate $\alpha_{js}\mu_{rs}$.

Consider the one-dimensional Markov chain, for any given state n and call class s, with the following state transition rates:
From state n to state $n + b_{js}$, $\nu_{js} I(C_j - n \geq r_s + b_{js})$;
From state n to state $n - b_{js}$, $\mu_{rs} n \frac{\alpha_{js}}{\sum_t \alpha_{jt}} I(n \geq b_{js})$.
The probability of admitting a call of class s is given by

$$a_{js} = 1 - \sum_{n=C_j-b_{js}-r_s+1}^{C_j} p_j(n) = \sum_{n=0}^{C_j-b_{js}-r_s} p_j(n). \tag{8}$$

Note that $p_j(n) \longrightarrow \alpha_{js} \longrightarrow p_j(n)$ forms another fixed-point problem, which can be solved by iteration to get the equilibrium distribution $p_j(n)$ and thus a_{js}.

If we use the cost function presented in the previous section to make routing decisions, a trunk reservation scheme will no longer be necessary since the idea of trunk reservation admission control is to prevent routing calls onto those longer routes and the cost function has already taken the length of the route into consideration.

3.3 Mapping 3: $p_j(n) \longrightarrow q_{rms}$

Given $p_j(n)$, define for link j, the probability of no more than n trunks are free (at most n trunks are free) as:

$$t_j(n) = \sum_{k=0}^{n} p_j(C_j - k). \tag{9}$$

Consider the case of no trunk reservation admission control, and use the min-max routing scheme extended from LLR described in the previous section, the probability of attempting a call of (r, s) on route (r, m) is the probability that all routes before the m^{th} route on the routing list have fewer free trunks on their most congested links, and all routes after the m^{th} route have at most the same number of free trunks on their most congested links, which can be expressed as:

$$\begin{aligned} q_{rms} &= \sum_{n=1}^{C_{\mathcal{L}_{rm}}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) \prod_{k=1}^{m-1} t_{\mathcal{L}_{rk}}(n-1) \cdot \\ &\quad \prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}(n). \end{aligned} \tag{10}$$

We consider steady state and the free bandwidth on link j C_j^f is replaced by $E[C_j^f]$, the expected average free capacity. Thus $\mathcal{L}_{rm}$, the most congested link on a route, becomes the statistically most congested link as:

$$\mathcal{L}_{rm} = argmax_{j \in (r,m)} z_j \tag{11}$$

where z_j is defined as the link load of link j:

$$z_j = \sum_{rms} \frac{\nu_{jrms} b_{js}}{\nu_{rs} C_j}, \tag{12}$$

which is also the long term average of link utilization.

When there is admission control with trunk reservation parameter r_s, this probability becomes:

$$\begin{aligned} q_{rms} &= \sum_{n=1}^{C_{\mathcal{L}_{rm}}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) t_{\mathcal{L}_{r1}}(n-1) \cdot \\ &\quad \prod_{k=2}^{m-1} t_{\mathcal{L}_{rk}}(n + r_s - 1) \cdot \\ &\quad \prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}(n + r_s) \end{aligned} \tag{13}$$

assuming that we do not impose trunk reservation on the first route in a set, since naturally that would be the shorted one among all even if it is not the direct route.

If we use the cost function proposed in the previous section, then since the choice of m^{th} route for routing the call indicates that all the routes before m^{th} route have a higher cost than the m^{th} route, and all the routes after the m^{th} route have at least the same cost, the probability of attempting the call on the m^{th} route can be expressed as:

$$q_{rms} = \sum_{n=1}^{C_{\mathcal{L}_{rm}}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) \cdot$$
$$\prod_{k=1}^{m-1} t_{\mathcal{L}_{rk}}[\frac{w_1}{w_2}(\sum_{j\in(r,k)} b_{js} - \sum_{j\in(r,m)} b_{js}) + n - 1] \cdot$$
$$\prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}[\frac{w_1}{w_2}(\sum_{j\in(r,k)} b_{js} - \sum_{j\in(r,m)} b_{js}) + n]. \quad (14)$$

3.4 End-to-end blocking probabilities

Finally, the end-to-end blocking probability for calls of class s between source-destination node pair r is given by

$$B_{rs} = 1 - \sum_{m} q_{rms} \prod_{j\in(r,m)} a_{js}. \quad (15)$$

Repeated substitution is used to obtain the equilibrium fixed point. And the end-to-end blocking probabilities can be calculated from the fixed point.

4 Asymptotic Correctness

By using Brouwer's fixed point theorem, it's easy to show that there exists a fixed point under the proposed fixed point approximation. In this section, we analyze the asymptotic correctness of our fixed point approximation. First we give a steady state explanation for q_{rms}, and formulate an optimization problem to solve for the most probable state for a single link. Then we establish an limiting regime and show that under the specified limiting regime, the blocking probability converges to our fixed point approximation.

We make the following observations. Under steady state, for traffic stream (r, s), the probability of attempting the call on the m^{th} route is q_{rms}. In reality, when a call request comes, using an adaptive routing scheme, the call is routed according to the actual traffic load in the network at that point in time. This type of traffic dispersement is called "metering" [13]. However, in our approximation the routing is modeled as if that each traffic stream has fixed probabilities to be routed onto a set of routes, and those probabilities add up to 1. This method is called "randomization". The metering method generally gives a better performance over randomization. Therefore, our approximation represents a conservative estimate, especially under heavy traffic, of the end-to-end call blocking probabilities.

Accepting this assumption, since random splitting of a Poisson process according to a fixed probability distribution results in processes which are individually Poisson, we have $\nu_{rms} = \lambda_{rs} \cdot q_{rms}$, the equivalent offered load onto route (r, m) from traffic (r, s), which is still a Poisson process.

Define a vector $\mathbf{n} = \{n_{rms}\}$, where n_{rms} is the number of calls in progress on route (r, m) from traffic stream (r, s). For clearer notation purposes, let b_{jrms} be the bandwidth requirement on link j from call (r, s) on route (r, m), and define vector $\mathbf{b} = \{b_{jrms}\}$. Also define vector $\mathbf{C} = \{C_j\}$ to be the link capacity. For a single link the stationary distribution $\pi(\mathbf{n})$ is given by:

$$\pi(\mathbf{n}) = G(\mathbf{n})^{-1} \prod_{(r,m)} \prod_{s} \frac{\nu_{rms}^{n_{rms}}}{n_{rms}!}, \quad \mathbf{n} \in A(\mathbf{C}), \tag{16}$$

where

$$A(\mathbf{C}) = \{\mathbf{n} > \mathbf{0} : \mathbf{b} \cdot \mathbf{n} \leq \mathbf{C}\} \tag{17}$$

defines the set for all feasible $\mathbf{n}$ under the link capacity constraint, and $G(\mathbf{n})$ is the normalizing factor:

$$G(\mathbf{n}) = \sum_{\mathbf{n}} (\prod_{(r,m)} \prod_{s} \frac{\nu_{rms}^{n_{rms}}}{n_{rms}!}). \tag{18}$$

Following Kelly's method in [1], we form the optimization problem of maximizing $\pi(\mathbf{n})$ to find the most probable state $\mathbf{n}$:

$$\begin{aligned} &\text{Max} \sum_{(r,m)} \sum_{s} (n_{rms} log \nu_{rms} - log \nu_{rms}!) \\ &\text{S.t. } \mathbf{n} \geq 0, \ \mathbf{b} \cdot \mathbf{n} \leq \mathbf{C}. \end{aligned} \tag{19}$$

Using Sterling's formula $\log n! \approx n \log n - n$, and replacing $\mathbf{n}$ by real vector $\mathbf{x} = \{x_{rms}\}$, the primal problem becomes:

$$\begin{aligned} &\text{Max} \sum_{(r,m)} \sum_{s} x_{rms} (\log \nu_{rms} - \log x_{rms} + 1) \\ &\text{S.t. } \mathbf{x} \geq 0, \ \mathbf{b} \cdot \mathbf{x} \leq \mathbf{C}. \end{aligned} \tag{20}$$

The objective function is differentiable and strictly concave over $x_{rms} \geq 0$; the feasible region is a closed convex set. Therefore there exists a unique maximum. Using Lagrangian method, the maximum can be found to be:

$$x_{rms}(\mathbf{y}) = \nu_{rms} \cdot \exp(-\sum_{j} y_j b_{jrms}). \tag{21}$$

where $\mathbf{y} = \{y_j\}$ is the Lagrangian multiplier. The constraints become

$$\mathbf{x}(\mathbf{y}) \geq 0, \ \mathbf{C} - \mathbf{b} \cdot \mathbf{x}(\mathbf{y}) \geq 0. \tag{22}$$

By introducing transformed variables

$$d_j = 1 - exp(-y_j) \tag{23}$$

we can rewrite the maximum in (21) as

$$x_{rms} = \nu_{rms} \cdot \prod_{j \in (r,m)} (1 - d_j)^{b_{jrms}}, \tag{24}$$

and d_j is any solution to the following:

$$\begin{aligned} &\sum_{(r,m)} \sum_{s} b_{jrms} \nu_{rms} \cdot \prod_{i} (1 - d_i)^{b_{irms}} \\ &\begin{cases} = C_j & \text{if} \quad d_j > 0 \\ \leq C_j & \text{if} \quad d_j = 0 \end{cases} \end{aligned} \tag{25}$$

and $d_j \in [0, 1)$.

Using the limiting scheme due to Kelly [1], we consider a sequence of networks indexed by N with increasing link capacity and offered traffic load. In addition, we also allow the number of alternative routes for each source-destination node pair to increase with N. This results in the following limiting regime:

$$\begin{aligned} &\frac{\lambda_{rs}(N)}{N} \longrightarrow \lambda_{rs}, \quad \frac{C_j(N)}{N} \longrightarrow C_j \ \text{ as } \ N \longrightarrow \infty \\ &\text{and } \sum_{M} q_{rms} = 1 \quad M \longrightarrow \infty \end{aligned} \tag{26}$$

with λ_{rs}/C_j fixed, and M is the total number of alternative routes. Let

$$k_{jrms} = \frac{\nu_{rms}}{C_j} = \frac{\lambda_{rs} q_{rms}}{C_j} \tag{27}$$

also be fixed based on our assumption with q_{rms}.

Following from [1], the blocking probability $B_{rms}(N)$, which is the stationary probability that a call from source (r, s) is accepted by route (r, m) is given by:

$$1 - B_{rms}(N) = q_{rms} \prod_{j} (1 - d_j)^{b_{jrms}} + o(1) \tag{28}$$

where d_j is the solution to (24).

We observe that $d_j = p_j(C_j)$ forms a set of valid solution to (24), which is the probability that the link is fully occupied. Denote n' as the number of free circuits on link j (n is the number of circuits occupied), and p' as the distribution of n'. As $N \longrightarrow \infty$ and $C_j(N) \longrightarrow \infty$, the distribution $p'_j(n') = p_j(C_j(N) - n)$ converges weakly to the geometric distribution given by [14]:

$$p'_j(n') = (1 - p) p^{n'} \tag{29}$$

where p is the positive root of

$$1 = \sum_{rms} \frac{b_{jrms}}{\mu_{rs}} k_{jrms} p^{b_{jrms}}. \tag{30}$$

Hence

$$a_{js} = \sum_{n'=b_{jrms}}^{C_j(N)} p_j'(n') = p^{b_{jrms}} - p^{C_j(N)+1}$$
$$=^{C_j(N)\to\infty} p^{b_{jrms}} = (1-d_j)^{b_{jrms}}. \tag{31}$$

Then the asymptotic form (28) can be written as

$$\begin{aligned} 1 - B_{rms}(N) &= q_{rms} \prod_j (1-d_j)^{b_{jrms}} + o(1) \\ &= q_{rms} \prod_j a_{js} + o(1) \end{aligned} \tag{32}$$

Therefore we get the approximation

$$\begin{aligned} B_{rs}(N) &= 1 - \sum_m (1 - B_{rms}(N)) \\ &= {}^{M,N\longrightarrow\infty} 1 - \sum_m q_{rms} \prod_j a_{js}, \end{aligned} \tag{33}$$

which is the form we presented (15) in the algorithm.

Similarly, from (24), load on route (r,m) from source (r,s) becomes

$$x_{rms} = \nu_{rms} \cdot \prod_{j\in(r,m)} a_{js} = q_{rms}\lambda_{rs} \prod_{j\in(r,m)} a_{js}. \tag{34}$$

Therefore, as seen by any individual link i,

$$x_{irms} = q_{rms}\lambda_{rs} \prod_{j\in(r,m), j\neq i} a_{js}, \tag{35}$$

which is our first mapping in the algorithm.

5 Experiments And Evaluation

In this section we give two network examples to compare the approximation results with that of discrete event simulation.

The first example is a five-node fully connected network depicted in Figure 2.

Capacity for each link is set to be 100. There are three classes of connections, which are indexed 1, 2, and 3. They have bandwidth requirements of 1, 2, and 3, respectively. When call admission control is used under heavy traffic, the trunk reservation parameter for each class is 2, 4, and 6, respectively.

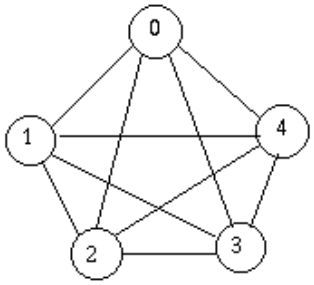

Figure 2: Topology of Example One.

For each node pair, the direct route and all two-hop routes are allowed. The direct route is listed first in the routing list, and the two-hop routes are listed in random order.

The detailed traffic rates are given in the Appendix of [15]. We have medium and heavy traffic. Heavy traffic rates are set to be double the medium rates. Simulation models were built using OPNET (OPtimized Network Engineering Tool). Both simulation and fixed point algorithm were run on a SunSparc 20 workstation.

We only display three node pairs here for comparison between fixed-point approximation (FPA) and discrete event simulation (DES). All simulations were run to get a 95% confidence interval. The results are listed in Table 1 through Table 3, with Table 1 showing results for medium traffic, and Tables 2 and 3 for heavy traffic with and without trunk reservation, respectively.

Although the traffic in this case is highly asymmetric, since routing is "symmetric", we observe that connections of the same class, with the same bandwidth requirement, encounter approximately the same blocking probability regardless of their source-destination node pair and input rate. However, they do vary slightly from one to another reflecting the random order in which the two-hop routes are listed.

The second example is borrowed from [6] with minor changes. The topology is derived from an existing commercial network and is depicted in Figure 2 below.

There are 16 nodes and 31 links, with link capacity ranging from 60 to 180 trunks. The detailed link-by-link traffic statistics and link capacities can be found in the Appendices of [6] and [15]. The traffic in the network consists of four types, namely class-1, 2, 3, and 4, and require bandwidth of 1, 2, 3, and 4 trunks, respectively. No admission control is employed in this experiment.

In routing, any node pair is allowed routes that have at most 4 hops. Multiple routes for one node pair are listed in order of increasing hops, with ties broken randomly. Each link is considered to have same unit length, so only the hop number is counted.

Table 1: Ex.1 with Medium Traffic

Pair	Class	FPA	DES
(0,3)	1	0.2234	(0.0341, 0.0348)
	2	0.3986	(0.2274. 0.2276)
	3	0.5354	(0.4730, 0.4735)
(1,2)	1	0.2239	(0.0358, 0.0360)
	2	0.3992	(0.2209, 0.2212)
	3	0.5357	(0.4701, 0.4710)
(2,4)	1	0.2224	(0.0336, 0.0340)
	2	0.3969	(0.2300, 0.2303)
	3	0.5333	(0.4673, 0.4677)
# Iterations		11	
CPU (sec.)		1.55	7.1×10^3

Table 2: Ex.1 with Heavy Traffic and no Trunk Reservation

Pair	Class	FPA	DES
(0,3)	1	0.4321	(0.1921, 0.1923)
	2	0.6720	(0.6059, 0.6070)
	3	0.8067	(0.8340, 0.8342)
(1,2)	1	0.4295	(0.1941, 0.1942)
	2	0.6680	(0.6076, 0.6078)
	3	0.8024	(0.8257, 0.8261)
(2,4)	1	0.4253	(0.1849, 0.1851)
	2	0.6633	(0.6002, 0.6004)
	3	0.7983	(0.8290, 0.8291)
# Iterations		9	
CPU (sec.)		1.24	1.3×10^4

Table 3: Ex.1 with Heavy Traffic and Trunk Reservation

Pair	Class	FPA	DES
(0,3)	1	0.0667	(0.0023, 0.0024)
	2	0.5494	(0.6117, 0.6120)
	3	0.8636	(0.9763, 0.9764)
(1,2)	1	0.0650	(0.0019, 0.0020)
	2	0.5446	(0.6041, 0.6043)
	3	0.8557	(0.9741, 0.9743)
(2,4)	1	0.0592	(0.0014, 0.0016)
	2	0.5249	(0.5765, 0.5766)
	3	0.8455	(0.9709, 0.9711)
# Iterations		12	
CPU (sec.)		10.86	1.3×10^4

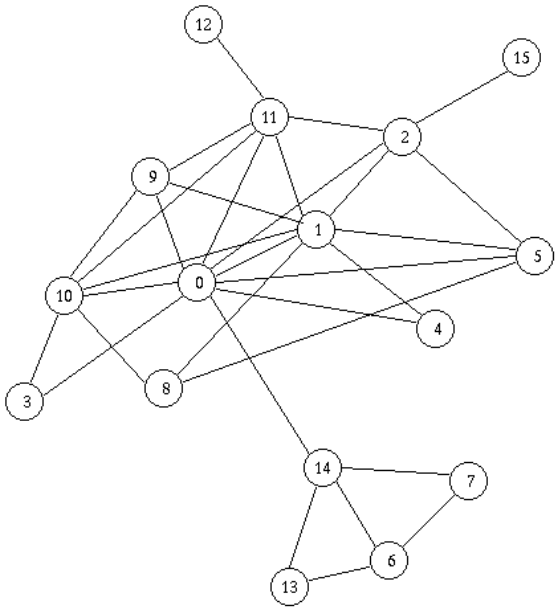

Figure 3: Topology of Example Network

Results for some selected node pairs and classes are listed in Table 4 through 8, each corresponding to a different traffic load. Table 4 corresponds to the "nominal" traffic which is provided in Appendix A. Tables 5 through 8 show the results for traffic 1.4, 1.6 and 1.8 times the nominal traffic, respectively.

The proposed fixed-point approximation gives conservative estimations generally, and it improves as the load gets heavier. These results strengthen the argument that these approximations are indeed very useful as estimators of worst case performance.

Table 4: Ex.2 Nominal Traffic.

Pair	Class	FPA	DES
(0,4)	4	0.0001	(0.0, 0.0)
(0,13)	1	0.0063	(0.0021, 0.0034)
(1,6)	1	0.0064	(0.0030, 0.0034)
(5,6)	3	0.0204	(0.0189, 0.0201)
(6,10)	2	0.0132	(0.0109, 0.0138)
(9,13)	4	0.0284	(0.0185, 0.0245)
# Iterations		18	
CPU (sec.)		94.1	3.7×10^4

Table 5: Ex. 2 1.2 Times The Nominal Traffic.

Pair	Class	FPA	DES
(0,4)	4	0.0032	(0.0, 0.0)
(0,13)	1	0.0365	(0.0351, 0.0369)
(1,6)	1	0.0369	(0.0303, 0.0311)
(5,6)	3	0.1146	(0.1103, 0.1137)
(6,10)	2	0.0735	(0.0543, 0.0573)
(9,13)	4	0.1641	(0.1213, 0.1268)
# Iterations		23	
CPU (sec.)		120.35	3.9×10^4

6 Application In Network Design Using CONSOL-OPTCAD

One of the main reasons that we are interested in developing an analytical approximation algorithm is because such an algorithm can be easily linked to mathematical programming tools to get network performance optimization and trade-off analysis. In this section, we link the proposed reduced load approximation method with CONSOL-OPTCAD [16], which is a tool for optimization-based design of large class of systems, and show how network parameter design can be realized.

Table 6: Ex. 2 1.4 Times The Nominal Traffic.

Pair	Class	FPA	DES
(0,4)	4	0.0182	(0.0122, 0.0179)
(0,13)	1	0.0744	(0.0729, 0.0766)
(1,6)	1	0.0773	(0.0697, 0.0701)
(5,6)	3	0.2295	(0.2262, 0.2278)
(6,10)	2	0.1474	(0.1420, 0.1483)
(9,13)	4	0.3071	(0.2794, 0.2848)
# Iterations		28	
CPU (sec.)		145.43	4.3×10^4

Table 7: Ex. 2 1.6 Times The Nominal Traffic.

Pair	Class	FPA	DES
(0,4)	4	0.0553	(0.0512, 0.0549)
(0,13)	1	0.1075	(0.0987, 0.1012)
(1,6)	1	0.1172	(0.1113, 0.1121)
(5,6)	3	0.3322	(0.3137, 0.3142)
(6,10)	2	0.2125	(0.2164, 0.2210)
(9,13)	4	0.4245	(0.3380, 0.3465)
# Iterations		24	
CPU (sec.)		125.55	5.6×10^4

Table 8: Ex. 2 1.8 Times The Nominal Traffic.

Pair	Class	FPA	DES
(0,4)	4	0.1126	(0.0025, 0.0026)
(0,13)	1	0.1355	(0.1492, 0.1500)
(1,6)	1	0.1563	(0.1445, 0.1466)
(5,6)	3	0.4197	(0.3922, 0.3940)
(6,10)	2	0.2691	(0.2572, 0.2583)
(9,13)	4	0.5190	(0.4791, 0.4793)
# Iterations		24	
CPU (sec.)		125.11	2.3×10^6

6.1 Design of Trunk Reservation Parameters

As a way of call admission control, trunk reservation regulates individual classes of traffic as well as their interrelationship. How to choose the combination of $\{r_1, r_2, ..., r_S\}$, where r_s is the trunk reservation parameter of class-s traffic and S is the total number of different classes the network carries, to get the best network performance is important. Here the network performance of interest is the average blocking probability of the network as a whole and the blocking probabilities that each individual class of traffic experiences.

This design problem is formulated as follows:

Design parameters: $r_1, r_2, ..., r_S$.

$$\begin{aligned} \text{Min} \quad & \frac{\sum_{(r,s)} \lambda_{rs} \cdot [1 - B_{rs}] \cdot B_{rs}}{\sum_{(r,s)} \lambda_{rs} \cdot [1 - B_{rs}]} \\ \text{S.t.} \quad & B_{rs} < bound_{rs}, \quad \text{all } (r,s) \end{aligned} \tag{36}$$

where $bound_{rs}$ is the desired upper bound for blocking probability.

Note that the objective function is a weighted average blocking probability over each class of traffic and each source-destination node pair.

In CONSOL-OPTCAD, we can provide two values, namely the good value and bad value, for each $bound_{r,s}$, indicating our level of satisfaction. Trade-off analysis can be carried out between minimizing the objective function value and satisfying the constraints, and thus decide the combination of $\{r_1, r_2, ..., r_S\}$.

By applying this model to our Example One of Section V, in the case with trunk reservation admission control, and by restricting the trunk reservation parameters to be less or equal to 5, we get the trade-off between the blocking probability of each class vs. the weighted average blocking probability, which is shown in Table 9(Numbers in bold are individual minima).

Because of the symmetry of Example One's topology, the same class of traffic encounters approximately the same blocking probability regardless of their source-destination node pair and input rate, although their input rates are counted in calculating the weighted average blocking probability. So the numbers displayed here are only distinguished by their classes but not their associated source-destination node pairs.

As we can see from Table 9, the weighted average blocking probability and the blocking probability of class-1 type of traffic achieve their optimum at the same time with trunk reservation parameter choice of 1,4 and 5. The reason is obvious: since class-1 has much smaller trunk reservation requirement than class-2 and 3, together with its lowest bandwidth requirement, it has the highest priority and chances of being admitted into the network.

Table 9: Trunk Reservation Parameter Design

Wgtd Avg	B_1	B_2	B_3	(r_1, r_2, r_3)
0.265	**0.036**	0.580	0.803	(1,4,5)
0.273	0.062	0.526	0.840	(1,3,5)
0.290	0.067	0.562	0.792	(2,4,5)
0.292	0.112	0.467	0.861	(1,2,5)
0.307	0.116	0.485	0.819	(1,2,4)
0.309	0.119	0.492	0.824	(2,3,5)
0.420	0.571	**0.284**	0.565	(4,1,2)
0.406	0.319	0.671	**0.322**	(3,5,1)

We can also get the optimal value of the weighted average blocking probability over one of the trunk reservation parameter while fixing the other two.

6.2 Design of Link Capacity

Similar to the first application, link capacities are another set of parameters which is critical in network design, since link capacities greatly affect network performance. It is natural to expect that we should assign higher capacities to more frequently congested links, and save capacities on rarely used ones.

The formulation of the problem is similar to that of the trunk reservation parameter design, except that the design parameters are $\{C_j\}$, all j.

We applied this model to Example One of Section V, in the case of heavy traffic without trunk reservation admission control, and set the link capacities of all peripheral links ((0,1),(1,2),(2,3),(3,4),(4,0)) to be α and all non-peripheral links ((0,2),(0,3),(1,4),(1,3),(2,4)) capacities to be β. By varying α and β independently from 80 to 120, we found that the weighted average blocking probability decreased from 0.608660 to 0.511441 while α increases from 80 to 120 but it does not change subject to the change in β.

We have shown that the proposed fixed-point approximation scheme can be applied to optimization tools to get trunk reservation parameter design and link capacity design analysis. Since the proposed fixed-point approximation generally gives conservative estimates, we are sure that the optimization constraints will be satisfied.

7 Sensitivity Analysis Using Automatic Differentiation

The idea behind automatic differentiation (AD) is not a new one. Since differentiation of functions defined by formulas is a mechanical process done according to fixed rules, e.g. the chain rule, it is highly suitable for automation along the same lines as function evaluation. However, the technique of automatic differentiation did not become popular and really applicable until pretty recently, due to the remarkable work by Andreas Griewank and Christian Bischof, who is also the major contributor of the AD package ADIFOR for FORTRAN and the recently available ADIC for C [17]. They take the form of preprocessors, which take user's function evaluation code as input and generate output code that evaluate both function and function derivatives at the same time.

However, just because the derivatives computed by automatic differentiation are those defined by the statements that were executed by a particular program run, what was actually computed may differ significantly from the derivative of the function one intended to compute [18]. This is especially true in the iterative evaluation of a function defined implicitly or otherwise. Usually the iteration continues until the value of $f(x)$ meet certain criteria. However, this may not be true of the value of $f^{'}(x)$ or higher derivatives. On the other hand, many engineering problems are impossible or impractical to be expressed in an explicit or exact form, and we have to turn to approximations which often take an iterative form. The problems lie in the design of programs to which AD is to be applied and can be handled most effectively by the programmer, especially for pitfalls arising from branching or iteration. Interested readers are referred to [19] for convergence of derivatives of functions defined implicitly or iteratively, and [20] for an example of automatic differentiation procedure for (single) fixed point problems.

Our interest here is to use AD to compute derivatives of the blocking probabilities w.r.t. offered traffic load. Our approximation for blocking probabilities is an iterative process involving implicit function definition and multiple fixed points, shown as follows:

$$\begin{aligned} \nu &= f_1(\mathbf{x}, \mathbf{a}), \quad f_1 : \mathcal{R}^m \times \mathcal{R}^n \to \mathcal{R}^l \\ \mathbf{a} &= f_2(\mathbf{x}, \nu), \quad f_2 : \mathcal{R}^m \times \mathcal{R}^l \to \mathcal{R}^n \end{aligned} \tag{37}$$

where $\mathbf{a}$ denotes the vector of link admissibility probabilities, ν denotes the vector of link traffic load, and $\mathbf{x}$ denotes the set of independent variables including the distribution of traffic load, network resource allocation(link capacities) and admission control parameters. The fixed points are $(\mathbf{a}_*, \nu_*)$ and the blocking probabilities $B_* = B(\mathbf{a}_*, \nu_*)$.

Based on the approximation scheme described in Section III, given $\mathbf{x}$, $\nu \to \nu_*$ and $\mathbf{a} \to \mathbf{a}_*$ in the following way:

```
Given x and a_0,
Until ||ν_k − ν_{k−1}|| ≤ Tol_ν  or  ||a_k − a_{k−1}|| ≤ Tol_a
    ν_k = f_1(x, a_k)
    a_{k+1} = f_2(x, ν_k)
```

$k = k + 1$
$B_* = f(\mathbf{a}_*, \nu_*)$
where Tol_ν and $Tol_{\mathbf{a}}$ are stopping criteria.

Naturally we are interested in $\frac{\partial B}{\partial \mathbf{x}}$, the sensitivity of blocking probability with respect to network design parameters, and eventually we would like to solve the following optimization problem:

$$\min \ \ f(B), \quad \text{S.t.} \quad g(\mathbf{x}) \geq 0,$$

where $f(\cdot)$ is some cost/penalty function and $g(\cdot)$ is the constraints on network designs.

Ideally we would want the derivative evaluation code generated by an AD preprocessor to take the form of the following:

Given $\mathbf{x}$, $\mathbf{a}_0$ and $\mathbf{a}_0^{'}$,
Until $\|\nu_k - \nu_{k-1}\| \leq Tol_\nu$
or $\|\mathbf{a}_k - \mathbf{a}_{k-1}\| \leq Tol_{\mathbf{a}}$
Until $\|\nu_k^{'} - \nu_{k-1}^{'}\| \leq Tol_{\nu'}$
or $\|\mathbf{a}_k^{'} - \mathbf{a}_{k-1}^{'}\| \leq Tol_{\mathbf{a}'}$
$\nu_k = f_1(\mathbf{x}, \mathbf{a}_k)$
$\nu_k^{'} = \frac{\partial f_1}{\partial \mathbf{x}} + \frac{\partial f_1}{\partial \mathbf{a}} \cdot \mathbf{a}_k^{'}$
$\mathbf{a}_{k+1} = f_2(\mathbf{x}, \nu_k)$
$\mathbf{a}_{k+1}^{'} = \frac{\partial f_2}{\partial \mathbf{x}} + \frac{\partial f_2}{\partial \nu} \cdot \nu_k^{'}$
$k = k + 1$
$B_* = f(\mathbf{a}_*, \nu_*)$
$B_{\mathbf{x}}^{'} = f_{\mathbf{a}}^{'} \mathbf{a}_{\mathbf{x}}^{'} + f_{\mathbf{x}}^{'} \nu_{\mathbf{x}}^{'}$

In this experiment, the AD package ADIC [17] is chosen to generate derivative code to calculate $\frac{\partial B}{\partial \mathbf{x}}$ for Example One described in Section V. There is no separate stopping criterion generated for derivative iteration by ADIC. The whole iteration is terminated when the function convergence criterion is satisfied. So additional stopping criterion for derivatives are added to the ADIC generated code.

The computational results are quite satisfying, and correspond to previous discussions, $B_{\mathbf{x}}^{'}$ did converge but was slower than B itself, 18 iterations vs. 11 iterations. Selected results are listed in Table 10. Independent variable $\mathbf{x}$ is chosen to be the offered traffic load of three traffic classes between source-destination node pair $(0, 1)$, so each blocking probability has three derivative values.

Table 10: Blocking Probability Derivatives

Pair,Class	$\frac{\partial B}{\partial x_1}$	$\frac{\partial B}{\partial x_2}$	$\frac{\partial B}{\partial x_3}$
$(0,1)0$	$5.66e-04$	$8.85e-04$	$1.03e-03$
$(0,2)2$	$7.91e-04$	$1.20e-03$	$1.36e-03$
$(0,3)0$	$6.68e-04$	$1.01e-03$	$1.15e-03$
$(2,3)0$	$3.32e-04$	$5.04e-04$	$5.71e-04$
$(3,4)1$	$4.19e-04$	$6.40e-04$	$7.29e-04$

8 Conclusions

In this paper we presented an approximation scheme of calculating the end-to-end, class-by-class blocking probability of a loss network with multirate traffic and adaptive routing scheme. It provides fairly good estimates of call blocking probabilities under normal and heavy traffic orders of magnitude faster than discrete event estimation. We also presented asymptotic analysis of the fixed point algorithm and showed that this algorithm gives conservative estimates in general. Two applications are provided as examples of how such analytical approximation schemes can be easily linked with mathematical programming tools to provide effective means for network design and trade-off analysis.

References

[1] F. P. Kelly, "Loss Networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, 1991.

[2] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Sel. Areas Commun.*, vol. 6, pp. 1598–1608, 1988.

[3] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM Network Design and Optimization: A Multirate Loss Network Framework," *IEEE Trans. Networking*, vol. 4, no. 4, pp. 531–543, 1996.

[4] S. Chung, A. Kashper, and K. W. Ross, "Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing," *IEEE Trans. Networking*, vol. 1, no. 1, pp. 105–115, 1993.

[5] S. Chung and K. W. Ross, "Reduced Load Approximations for Multirate Loss Networks," *IEEE Trans. Commun.*, vol. 41, no. 8, pp. 1222–1231, 1993.

[6] A. G. Greenberg and R. Srikant, "Computational Techniques for Accurate Performance Evaluation of Multirate, Multihop Communication Networks," *IEEE/ACM Trans. Networking*, vol. 5, no. 2, pp. 266–277, 1997.

[7] N. B. Bean, R. J. Gibbens, and S. Zachary, "Asymptotic Analysis of Single Resource Loss Systems in Heavy Traffic, with Applications to Integrated Networks," Preprint.

[8] A. Girard and M. A. Bell, "Blocking evaluation for networks with residual capacity adaptive routing," *IEEE Trans. Commun.*, vol. 37, pp. 1372–1380, 1990.

[9] J. A. Morrison, K. G. Ramakrishnan, and D. Mitra, "Refined Asymptotic Approximation to Loss Probabilities and Their Sensitivities in Shared Unbuffered Resources," *SIAM J. Appl. Math.*, vol. 59, no. 2, pp. 494–513, 1998.

[10] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "Virtual Private Networks: Joint Resource Allocation and Routing Design," *Proc. INFOCOM*, vol. 1, no. 1, pp. 480–490, 1999.

[11] J. S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Trans. Commun.*, vol. 29, no. 8, pp. 1474–1481, 1981.

[12] N. B. Bean, R. J. Gibbens, and S. Zachary, "The Performance of Single Resource Loss Systems in Multiservice Networks," Preprint.

[13] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd edition, 1994.

[14] F. P. Kelly, "Blocking Probabilities in large Circuit Switched Networks," *Adv. Appl. Prob.*, vol. 18, pp. 473–505, 1986.

[15] M. Liu, "Performance Evaluation for Multirate Multihop Communication Networks," *Master's Thesis*, www.isr.umd.edu/TechReports/ISR/ Tech. Rep. MS 97-8, Institute for Systems Research, 1997.

[16] M. K. H. Fan, A. L. Tits, J. Zhou, L. S. Wang, and J. Koninchx, "CONSOLE User's Manual," Tech. Rep. TR 87-212r2, Systems Research Center, 1990.

[17] Christian Bischof and Lucas Roh, *User's Guide to ADIC 1.0 Revision 1.0 Beta 1*, 1997.

[18] Louis B. Rall and George F. Corliss, "An introduction to automatic differentiation," pp. 1–18, 1990.

[19] Andreas Griewank, Christian Bischof, George Corliss, Alan Carle, and Karen Williamson, "Derivative convergence for iterative equation solvers," *Optimization Methods and Software*, vol. 2, pp. 321–355, 1993.

[20] Bruce Christianson, "Reverse accumulation and attractive fixed points," *Optimization Methods and Software*, vol. 3, pp. 311–326, 1994.