Algorithms for Sensitivity Analysis of Markov Systems Through Potentials and Perturbation Realization

Xi-Ren Cao, *Fellow, IEEE,* and Yat-wah Wan

***Abstract*—We provide algorithms to compute the performance derivatives of Markov chains with respect to changes in their transition matrices and of Markov processes with respect to changes in their infinitesimal generators. Our algorithms are readily applicable to the control and optimization of these Markov systems, since they are based on analyzing a single sample path and do not need explicit specification of transition matrices, nor infinitesimal generators. Compared to infinitesimal perturbation analysis (IPA), the algorithms have a wider scope of application and require nearly the same computational effort. Numerical examples are provided to illustrate the applications of the algorithms. In particular, we apply one of our algorithms to a closed queueing network and the results are promising.**

***Index Terms*—Importance sampling, on-line optimization, performance sensitivity, perturbation methods, queueing networks.**

I. Introduction

PERTURBATION analysis is one of the single sample path-based performance sensitivity analysis techniques for discrete-event dynamic systems (see e.g., [7], [14], and [19]). The main objective of perturbation analysis is to obtain performance sensitivities with respect to system parameters by analyzing a single sample path of a discrete-event system. This research area is promising because of its practical usefulness: First, perturbation analysis saves a great amount of computation in simulation for system optimization, since many derivatives can be obtained by analyzing the same sample path; second and perhaps more importantly, the derivatives can be applied to on-line performance optimization of real world systems where changing the values of parameters to estimate derivatives is infeasible. (Papers regarding the applications of perturbation analysis to various engineering problems include [3], [6], [9], [17], [26], and [27], just to name a few.) The computation of the performance derivatives based on a single sample path of a system is especially useful in the optimization of modern communication systems where the rapid changing environment makes it hard to implement any off-line simulation methods.

Manuscript received May 13, 1996; revised June 29, 1997. Recommended by Associate Editor, Takahashi. The work of X.-R. Cao was supported in part by Hong Kong UGC under Grant HKUST 599/94E.

X.-R. Cao is with the Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong.

Y. Wan is with the Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Kowloon, Hong Kong.

Publisher Item Identifier S 1063-6536(98)04663-6.

It has been proved that the simple perturbation analysis algorithms (called the infinitesimal perturbation analysis) provide unbiased or strongly consistent estimates of the performance derivatives for many systems [9], [14], [19], [24]; it is also known that this is not true for many others [14], [19]. Much effort has been made to extend infinitesimal perturbation analysis (IPA) to cases where it does not work well; useful techniques that apply to different cases where IPA fails have been proposed (see, e.g., [2], [8], [10]–[13], [16], [20], and [25]). However, a general approach that is simple and applies to a wide class of problems is yet to be developed.

Recently, a new approach was proposed in [4] in this direction. The work was motivated by [10]–[13]. The approach is based on Markov models; it provides formulas for $\partial\eta/\partial Q$, the derivative of the steady-state performance measure η of a Markov process (or Markov chain) with respect to Q, the *change* in the infinitesimal generator A (or transition matrice P). It is shown in [4] that the quantities involved in the derivative formulas can be easily estimated by analyzing a single sample path of a Markov process (or Markov chain), and that the derivative estimates obtained using these formulas are strongly consistent. It is well known that IPA does not yield the correct estimates of performance derivatives for this problem. Since the Markov model is the most fundamental model for stochastic systems, the formulas developed in [4] provide a new widely applicable approach for sensitivity estimation. For example, η can be the expected number of customers in a closed queueing network, and Q the effect of a change in the routing matrix, the service rates, or both. With estimated $\partial\eta/\partial Q$, one can consider the optimal routing matrix, service rates, or both, with respect to η.

While principles and formulas are developed in [4], it stops short of providing practically applicable algorithms for sensitivity estimation. The goal of this paper is to develop algorithms based on the analysis in [4] that yield unbiased and strongly consistent estimates of performance sensitivities. The algorithms are based on a single sample path of a Markov system, and the computation involved is almost comparable to that of IPA.

The rest of this paper is organized as follows. Section II describes the problem and summarizes the results in [4], leaving proofs and details to the original paper. The algorithms for Markov chains and Markov processes are presented in Section III and IV, respectively. All algorithms are illustrated by numerical examples with discussion. In Section V, we

1063–6536/98$10.00 © 1998 IEEE

apply an algorithm from Section IV to a closed queueing network. We first estimate the derivatives of the average number of customers in stations; then we estimate the derivatives of the probability of the number of customers in a station exceeding a prefixed threshold. The latter problem has a practical meaning in that it models the blocking probability in manufacturing and communication systems; it cannot be solved by IPA. Our results show that the approach yields accurate estimates with a reasonable computational and analytical complexity. Finally, we conclude our paper in Section VI.

II. The Model and Existing Results

Consider a regular, positive recurrent, and irreducible Markov process $X = \{X_t; t \geq 0\}$ with a countable state space $\mathcal{E} = \{1, 2, \cdots\}$ and an infinitesimal generator $A = [a_{ij}]$, where $a_{ii} < 0, a_{ij} \geq 0, i \neq j$, and $|a_{ii}| < K$, a finite constant. A real-valued function defined on $\mathcal{E}$, $f : \mathcal{E} \to \mathcal{R}$, is called the *performance function* associated with X, if a cost $f(i)$ is incurred per unit time when the process is at state i. Let $e = (1, 1, \cdots)'$, where the prime represents transpose, be the column vector whose components are all ones, and $\pi = (\pi_1, \pi_2, \cdots)$ be the steady-state probability vector of X. We know that π can be found from

$$\pi e = 1$$

and

$$\pi A = 0 \tag{1}$$

and the steady-state performance measure of X from

$$\eta = E_\pi(f) = \sum_{i=1}^{M} \pi_i f(i) = \pi f.$$

The sensitivity of η with respective to A is an interesting and important question. Suppose A changes to $A_\delta = A + \delta Q$, where δ is a very small positive real number, $Qe = 0$, and $q_{ij} \geq 0$ for $a_{ij} = 0, i \neq j$, and $q_{ii} \leq 0$ for $a_{ii} = 0$. Under this construction, A_δ is a well-defined infinitesimal generator, and hence its steady-state distribution π_δ and performance measure η_δ are also clearly defined. We are interested in derivatives defined by

$$\frac{\partial \eta}{\partial Q} = \lim_{\delta \to 0} \frac{\eta_\delta - \eta}{\delta}, \quad \frac{\partial \pi}{\partial Q} = \lim_{\delta \to 0} \frac{\pi_\delta - \pi}{\delta}$$

and we have

$$\frac{\partial A}{\partial Q} = \lim_{\delta \to 0} \frac{A_\delta - A}{\delta} = Q.$$

It is clear that for a fixed A, $\partial\eta/\partial Q$ varies when Q changes. Consequently, what we will present is in fact the derivative (sensitivity) of η with respect to the change in A *in the direction of* Q. The notation used here is consistent with that of the "directional derivative" in the calculus of multivariable functions. In the following, for compactness, we will put A behind the scene and express derivatives in terms of Q, or its variants.

Let $\{X_t^{(i)}; t \geq 0\} = \{X_t | X_0 = i; t \geq 0\}$ be a Markov process starting from state i, $i \in \mathcal{E}$. Assume that it satisfies the following assumption, which is held by all Markov processes with practical significance.

Assumption A:

$$E_\pi(|f|) = \sum_{i \in \mathcal{E}} \pi_i |f(i)| < \infty$$

and

$$\lim_{t \to \infty} E[|f(X_t^{\{i\}})|] = E_\pi(|f|), \qquad i \in \mathcal{E}.$$

For $\{X_t^{\{i\}}\}$ the expected cost incurred up to epoch $T(>0)$ is $E[\int_0^T f(X_t^{\{i\}})\, dt]$. The difference in cost incurred up to T by starting from state j rather than i should then be

$$E\left[\int_0^T f\left(X_t^{\{j\}}\right) dt\right] - E\left[\int_0^T f\left(X_t^{\{i\}}\right) dt\right].$$

It is proved in [4] that the limits of differences in costs by starting from different states exist.

Theorem 1: Under Assumption A, the limit

$$d_{ij} = \lim_{T \to \infty} \left\{ E\left[\int_0^T f(X_t^{\{j\}})\, dt\right] - E\left[\int_0^T f(X_t^{\{i\}})\, dt\right] \right\} \qquad i, j \in \mathcal{E}$$

exists, i.e., the above expression converges to a finite number when T goes to infinity.

These d_{ij}'s are called *perturbation realization factors*, and they form the *realization matrix* $D = [d_{ij}]$. There are various equivalent forms of d_{ij}'s. One form involves $S^{\{j\}}(i) = \inf\{t : t \geq 0, X_t^{\{j\}} = i\}$, the *first passage time* from state j to state i.

Theorem 2:

$$\begin{aligned} d_{ij} &= E\left[\int_0^{S^{\{j\}}(i)} f(X_t^{\{j\}})\, dt\right] - E[S^{\{j\}}(i)]\eta \\ &= E\left\{\int_0^{S^{\{j\}}(i)} [f(X_t^{\{j\}}) - \eta]\, dt\right\}. \end{aligned} \tag{2}$$

From Theorem 1, one can easily show that

$$d_{ij} = d_{ik} + d_{kj}, \quad i, j, k \in \mathcal{E}. \tag{3}$$

Such a functional form of d_{ij} is analogous to those in physics, and hence reveals that d_{ij} can be defined through *potential*, or *performance potential* associated with states: arbitrarily pick up $k^* \in \mathcal{E}$ and $c \in$ real number, the potential of states $\{g_i\}$ can be defined

$$g_{k^*} = c,$$

and

$$g_i = g_{k^*} + d_{k^* i}, \qquad i \neq k^* \tag{4}$$

which easily shows that

$$d_{ij} = g_j - g_i, \qquad \text{for all } i, j \in \mathcal{E}. \tag{5}$$

Since the datum level of a potential can be set arbitrarily, there can be uncountably many different versions of potential. One of them is

$$g = D'\pi'. \tag{6}$$

It is shown that

$$g = -A^{\#} f \tag{7}$$

where

$$A^{\#} = (A + e\pi)^{-1} - e\pi \tag{8}$$

is the *group inverse* of A. We have [4]

$$g_i = \lim_{T\to\infty} \left\{ E\left[\int_0^T f\left(X_t^{\{i\}}\right) dt\right] - T\eta \right\} \tag{9}$$

and hence for a fixed T, g_i can be estimated by

$$g_i(T) = E\left[\int_0^T f\left(X_t^{\{i\}}\right) dt\right] - T\eta. \tag{10}$$

The main result in [4] is as follows.

Theorem 3: The performance derivative can be calculated from the group inverse $A^{\#}$, the realization matrix D, or the potential vector g

$$\frac{\partial \eta}{\partial Q} = -\pi Q A^{\#} f \tag{11}$$

$$= \pi Q D' \pi' \tag{12}$$

$$= \pi Q g. \tag{13}$$

For any constant c, $g^* = g + ce$ can be chosen as a potential vector. g^* satisfies (5). Since $Qe = 0$, we have $Qg^* = Qg$. Thus, g in (13) can be replaced by g^*. In particular, we can simply use

$$g_i(T) = E\left[\int_0^T f\left(X_t^{\{i\}}\right) dt\right] \tag{14}$$

instead of (10) as an estimate of the potential vector in (13).

There is a completely analogous problem in the setting of Markov chains. Let $X = \{X_n;\, n \geq 0\}$ be a positive irreducible Markov chain on the state space $\mathcal{E}$ with the transition probability matrix $P = [p_{ij}] \in \mathcal{E}$. We will use the same symbols π, f, and Q for Markov chains with the obvious modification in their interpretation. Suppose P changes to $P + \delta Q$, where $q_{ij} \geq 0$ for $p_{ij} = 0$, and $Qe = 0$. Again, our interest is in a derivative defined by $\partial\eta/\partial Q = \lim_{\delta\to 0}(\eta_\delta - \eta)/\delta$, which is the sensitivity of η with respect to Q, the change in the probability transition matrix P.

The sensitivity analysis of Markov chains follows directly from that of Markov processes. Given the transition probability matrix P, we can treat the Markov chain as the embedded chain of the Markov process with infinitesimal generator $A = P - I$, which is equivalent to saying that all infinitesimal rates out of states are equal to one, and

$$a_{ij} = \begin{cases} -[1 - p_{ii}], & \text{if } i = j \\ p_{ij}, & \text{if } i \neq j. \end{cases}$$

It can be shown that the performance measure of any embedded Markov chain is the same as the Markov process constructed as above [21]. With $A = P - I$, changing P to $P + \delta Q$ is equivalent to changing A to $A + \delta Q$, and derivatives in the direction of Q can be deduced from the corresponding results, e.g., Theorem 3, of Markov processes by substituting A with $P - I$.

To be complete, we list out terms relevant to our algorithms for Markov chains. Let $\{X_n^{\{i\}}\} = \{X_n | X_0 = i;\, n \geq 0\}$ be a Markov chain starting from state i, and $L^{\{j\}}(i) = \inf\{n | X_n^{\{j\}} = i;\, n \geq 0\}$ be the first passage time from state j to state i. We consider Markov chains that satisfy the following assumption.

Assumption B:

$$E_\pi(|f|) = \sum_{i\in\mathcal{E}} \pi_i |f(i)| < \infty$$

and

$$\lim_{t\to\infty} E[|f(X_n^{\{i\}})|] = E_\pi(|f|), \qquad i \in \mathcal{E}.$$

The realization factor for a Markov chain is found from

$$d_{ij} = E\left\{ \sum_{k=0}^{L^{\{j\}}(i)-1} f(X_k^{\{j\}}) \right\} - E[L^{\{j\}}(i)]\eta$$

$$= E\left\{ \sum_{k=0}^{L^{\{j\}}(i)-1} [f(X_k^{\{j\}}) - \eta] \right\} \tag{15}$$

which also defines the realization matrix D. The potential vector g remains unchanged as long as A is calculated from $P - I$; the discrete analog of (10) and (14) are

$$g_i(n) = E\left[\sum_{k=0}^{n-1} f\left(X_k^{\{i\}}\right)\right] - n\eta \tag{16}$$

and

$$g_i(n) = E\left[\sum_{k=0}^{n-1} f\left(X_k^{\{i\}}\right)\right]. \tag{17}$$

III. Algorithms for Markov Chains

We will concentrate on a practical approach by estimating π, η, g, and D, from a sample path. Our approach is documented in six algorithms: Algorithms 1c, 2c, and 3c for Markov chains; Algorithms 1p, 2p, and 3p for Markov processes. The estimation of π and η on a sample path is straightforward. Let $\epsilon^i(x)$ be the indicator function for state i, i.e., $\epsilon^i(x) = 1$ for $x = i$, and $= 0$ otherwise. Then

$$\pi_i = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \epsilon^i(X_k), \qquad w.p.1 \tag{18}$$

and

$$\eta = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f(X_k), \qquad w.p.1. \tag{19}$$

The estimation of $E[L^{\{j\}}(i)]$ and the expected total cost accumulated during $L^{\{j\}}(i)$ on a sample path is equally straightforward. Define two sequences of epochs $\{j_s\}$ and $\{i_s\}$ for *each* pair of states j and i, where

$$\begin{aligned} i_0 &= 0 \\ j_s &= \text{the epoch that } \{X_n\} \text{ first visits state} \\ &\qquad j \text{ after } i_{s-1}, \qquad s \geq 1 \\ i_s &= \text{the epoch that } \{X_n\} \text{ first visits state} \\ &\qquad i \text{ after } j_s, \qquad s \geq 1. \end{aligned} \tag{20}$$

$\{j_s\}$ and $\{i_s\}$ are well defined on a sample path. Now define $L_s^{\{j\}}(i) = i_s - j_s$ and $R_s = \sum_{k=j_s}^{i_s-1} f(X_k)$. The Markov

property ensures that $L_s^{\{j\}}(i)$'s are independently identically distributed (i.i.d.), and each $L_s^{\{j\}}(i)$ is stochastically equivalent to the first passage time from state j to state i. By the same argument, R_s's are i.i.d., and each R_s is stochastically equivalent to the total cost accumulated within a first passage time from state j to state i. Since the chain is assumed to be positive recurrent, we have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{s=1}^{N}L_s^{\{j\}}(i)=E[L^{\{j\}}(i)],\qquad w.p.1 \tag{21}$$

and

$$\lim_{N\to\infty}\frac{1}{N}\sum_{s=1}^{N}R_s=E\left[\sum_{k=0}^{L^{\{j\}}(i)-1}f(X_k)\right],\qquad w.p.1 \tag{22}$$

based on which d_{ij}'s can be estimated from (15). When d_{ij}'s are known, the potentials can be obtained in the same fashion as (4) by using any row of D. To get a more accurate estimate, we may estimate the potential through all d_{ij}'s, that is, we get g from (6). The first algorithm estimates $\partial\eta/\partial Q$ based on (13) and the procedure outlined above.

Algorithm 1c:

1) Estimate $E\{\sum_{k=0}^{L^{\{j\}}(i)-1}f(X_k^{\{j\}})\}$, $E[L^{\{j\}}(i)]$, π, and η on a single sample path.
2) Calculate d_{ij}'s and the realization matrix D by using (15).
3) Set $g=D'\pi'$.
4) Calculate $\partial\eta/\partial Q$ by (13).

Example 1: We consider a Markov chain with ten states. The state transition matrix is shown in (22a) at the bottom of the page and the performance function is

$$f=(10\quad 5\quad 1\quad 15\quad 3\quad 0\quad 7\quad 20\quad 2\quad 18)'$$

and the matrix representing the direction of change in P is shown in (22b) at the bottom of the page. All the parameters are chosen arbitrarily. We did ten simulation runs, and each consists of 100 000 state transitions.

The theoretical values as well as the means and standard deviations of the estimated realization factors are listed in Tables I, II, and III, respectively. The estimated matrix D is indeed skew-symmetric and standard deviations are of the order 10^{-2}. The statistics of the potentials based on $g=D'\pi'$ are listed in Table IV. The performance derivative $\partial\eta/\partial Q$ is -0.1176; the estimated value is -0.1173 with a standard deviation 0.0013. □

The estimation of d_{ij} in the first algorithm is a bit involved. One needs to keep track of the cumulative sums of $\{L_s^{\{j\}}\{i\}\}$ and $\{R_s\}$ for *every* pair of i and j. At each transition, the number of variables that one needs to update is of the same order of the state space, which makes the updating computationally intensive.

The second algorithm estimates g from (17) and then uses (13) to calculate the performance derivative. The potential g can be estimated on sample path in a way similar to π. Based on the ergodicity of a positive recurrent Markov chain, (17) leads to

$$g_i(n)=\lim_{N\to\infty}\frac{\displaystyle\sum_{k=0}^{N-n+1}\left\{\epsilon^i(X_k)\left[\sum_{j=0}^{n-1}f(X_{k+j})\right]\right\}}{\displaystyle\sum_{k=0}^{N-n+1}\epsilon^i(X_k),}\qquad w.p.1. \tag{23}$$

The convergence in (23) is not obvious, since the items $\sum_{j=0}^{n-1}f(X_{k+j})$ for different k may not be independent.

$$P=\begin{pmatrix}
0.20 & 0.00 & 0.05 & 0.10 & 0.15 & 0.15 & 0.05 & 0.05 & 0.05 & 0.20\\
0.30 & 0.00 & 0.00 & 0.20 & 0.10 & 0.15 & 0.15 & 0.05 & 0.05 & 0.00\\
0.00 & 0.15 & 0.05 & 0.30 & 0.00 & 0.05 & 0.20 & 0.20 & 0.05 & 0.00\\
0.05 & 0.10 & 0.25 & 0.00 & 0.30 & 0.00 & 0.05 & 0.20 & 0.05 & 0.000\\
0.00 & 0.20 & 0.15 & 0.00 & 0.15 & 0.00 & 0.15 & 0.25 & 0.00 & 0.100\\
0.00 & 0.10 & 0.30 & 0.00 & 0.20 & 0.10 & 0.10 & 0.00 & 0.15 & 0.050\\
0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.100\\
0.00 & 0.20 & 0.00 & 0.20 & 0.00 & 0.20 & 0.00 & 0.20 & 0.00 & 0.200\\
0.05 & 0.15 & 0.25 & 0.00 & 0.15 & 0.15 & 0.15 & 0.00 & 0.00 & 0.100\\
0.15 & 0.05 & 0.00 & 0.20 & 0.15 & 0.10 & 0.20 & 0.10 & 0.05 & 0.000
\end{pmatrix} \tag{22a}$$

$$Q=\begin{pmatrix}
-0.010 & 0.000 & 0.005 & 0.005 & -0.010 & 0.010 & 0.010 & 0.005 & 0.005 & -0.020\\
-0.010 & 0.000 & 0.000 & 0.015 & 0.005 & 0.005 & -0.005 & -0.005 & -0.005 & 0.000\\
0.000 & 0.010 & 0.010 & 0.010 & 0.000 & -0.010 & 0.000 & -0.010 & -0.010 & 0.000\\
0.005 & -0.020 & 0.005 & 0.000 & 0.005 & 0.000 & 0.010 & -0.010 & 0.005 & 0.000\\
0.000 & 0.010 & -0.010 & 0.000 & 0.010 & 0.000 & -0.010 & 0.010 & 0.000 & -0.010\\
0.000 & 0.010 & -0.010 & 0.000 & -0.020 & 0.005 & 0.005 & 0.000 & 0.005 & 0.005\\
0.010 & -0.010 & 0.010 & -0.010 & 0.010 & -0.010 & 0.010 & -0.010 & 0.010 & -0.010\\
0.000 & 0.010 & 0.000 & -0.010 & 0.000 & 0.010 & 0.000 & -0.005 & 0.000 & -0.005\\
0.010 & -0.010 & -0.020 & 0.000 & 0.010 & 0.010 & 0.010 & 0.000 & 0.000 & -0.010\\
0.010 & -0.010 & 0.000 & 0.010 & -0.010 & -0.010 & 0.010 & -0.010 & 0.010 & 0.000
\end{pmatrix} \tag{22b}$$

TABLE I
THE THEORETICAL VALUES OF REALIZATION ERRORS IN EXAMPLE 1

	1	2	3	4	5	6	7	8	9	10
1	0.000	-5.890	-6.987	4.403	-5.124	-15.418	-3.863	12.232	-11.899	7.749
2	5.890	0.000	-1.097	10.293	0.766	-9.528	2.028	18.122	-6.009	13.639
3	6.987	1.097	0.000	11.389	1.863	-8.430	3.124	19.219	-4.912	14.735
4	-4.403	-10.293	-11.389	0.000	-9.527	-19.820	-8.265	7.830	-16.302	3.346
5	5.124	-0.766	-1.863	9.527	0.000	-10.294	1.260	17.356	-6.775	12.873
6	15.418	9.528	8.430	19.820	10.294	0.000	11.556	27.650	3.519	23.167
7	3.863	-2.028	-3.124	8.265	-1.261	-11.556	0.000	16.095	-8.036	11.611
8	-12.232	-18.122	-19.219	-7.830	-17.356	-27.650	-16.095	0.000	-24.130	-4.484
9	11.899	6.009	4.912	16.302	6.775	-3.519	8.036	24.130	0.000	19.647
10	-7.749	-13.639	-14.735	-3.346	-12.873	-23.167	-11.610	4.484	-19.647	0.000

TABLE II
THE MEAN REALIZATION ERRORS IN EXAMPLE 1

	1	2	3	4	5	6	7	8	9	10
1	0.000	-5.801	-6.983	4.336	-5.106	-15.377	-3.827	12.286	-11.727	7.756
2	5.800	0.000	-1.012	10.294	0.780	-9.474	2.097	18.204	-5.898	13.682
3	6.983	1.012	0.000	11.381	1.838	-8.416	3.146	19.217	-4.912	14.769
4	-4.336	-10.294	-11.381	0.000	-9.492	-19.690	-8.143	7.876	-16.217	3.442
5	5.105	-0.782	-1.838	9.491	0.000	-10.223	1.390	17.408	-6.582	12.918
6	15.376	9.472	8.414	19.689	10.221	0.000	11.684	27.629	3.647	23.214
7	3.827	-2.098	-3.147	8.142	-1.391	-11.687	0.000	16.014	-7.999	11.629
8	-12.285	-18.204	-19.218	-7.875	-17.409	-27.630	-16.014	0.000	-24.069	-4.491
9	11.726	5.895	4.910	16.214	6.579	-3.653	7.997	24.067	0.000	19.709
10	-7.755	-13.683	-14.768	-3.440	-12.920	-23..216	-11.629	4.493	-19.713	0.000

The proof of (23) is based on a fundamental theorem on ergodicity [1]: Let $X = \{X_k, k \geq 0\}$ be an ergodic process on state space $\mathcal{E}$; $\phi(x_1, x_2, \cdots)$be a measurable function on $\mathcal{E}$. Then the process $Z = \{Z_k, k \geq 0\}$ with $Z_k = \phi(X_k, X_{k+1}, \cdots)$ is also ergodic. In our case, we define $Z_k = \epsilon^i(X_k)[\sum_{j=0}^{n-1} f(X_{k+j})]$; then $\{Z_k, k \geq 0\}$ is ergodic. Thus, (23) holds. The same theorem can be applied to prove many similar results.

Algorithm 2c:

1) Estimate the steady-state probabilities π and potentials g by using (18) and (23).
2) Calculate $\partial\eta/\partial Q$ by (13).

One problem remaining is how to choose n. According to the meanings of potentials, what matters is their differences, i.e., the realization factors $d_{ij} = g_j - g_i$. We use the expectation of the sum of the performance function over a period with a fixed length to approximate g_i, $i = 1, 2, \cdots$. Ideally, to estimate d_{ij}, the length should be the first passage time from state j to state i and hence the length of the period, n, should be comparable to the mean of the first passage time.

It is clear that the larger the n is, the smaller the bias of $g_j - g_i$ as an estimate of $d_{i,j}$ is. On the other hand, the larger the n is, the larger the variance of the estimate is. Therefore, there is a tradeoff in choosing n. One can prove that the distribution of the first passage times has an exponential tail, and we expect that n may be chosen as a small number. The following simulation example provides some empirical evidence.

Example 2: We consider the same Markov chain as in Example 1. We choose $n = 13, 5, 10, 15, 20$. For each value of n, we do two sets of simulation, each set of ten runs. Each simulation run contains of 100 000 state transitions in the first set, and 1 000 000 transitions in the second set. The means and standard deviations of the estimated performance derivatives as well as its theoretical value are listed in Tables V and VI.

TABLE III
THE STANDARD DEVIATIONS OF REALIZATION ERRORS IN EXAMPLE 1

	1	2	3	4	5	6	7	8	9	10
1	0.000	0.017	0.060	0.043	0.052	0.017	0.047	0.077	0.144	0.063
2	0.017	0.000	0.020	0.027	0.011	0.025	0.035	0.025	0.109	0.040
3	0.060	0.020	0.000	0.029	0.049	0.054	0.025	0.024	0.065	0.076
4	0.043	0.028	0.029	0.000	0.037	0.028	0.025	0.041	0.116	0.100
5	0.052	0.011	0.050	0.038	0.000	0.021	0.037	0.045	0.041	0.026
6	0.017	0.024	0.054	0.027	0.020	0.000	0.025	0.041	0.037	0.070
7	0.047	0.036	0.025	0.025	0.037	0.025	0.000	0.039	0.059	0.032
8	0.077	0.024	0.024	0.041	0.044	0.041	0.039	0.000	0.128	0.064
9	0.146	0.110	0.065	0.117	0.042	0.039	0.059	0.130	0.000	0.102
10	0.064	0.040	0.076	0.101	0.025	0.068	0.031	0.065	0.097	0.000

TABLE IV
THE POTENTIALS BASED ON REALIZATION FACTORS IN EXAMPLE 1

	1	2	3	4	5	6	7	8	9	10
Theoretic	1.865	-4.025	-5.121	6.268	-3.259	-13.553	-1.997	14.098	-10.033	9.614
Mean	1.859	-4.039	-5.092	6.237	-3.273	-13.517	-1.912	14.132	-9.932	9.671
SD	0.0122	0.0074	0.0105	0.0127	0.0111	0.0038	0.0148	0.0146	0.0405	0.0206

TABLE V
THE PERFORMANCE DERIVATIVES IN EXAMPLE 2 WITH 100 000 TRANSITIONS

n	1	2	3	5	10	15	20	Theoretic
Mean	-0.0979	-0.1224	-0.1162	-0.1172	-0.1180	-0.1183	-0.1176	-0.1176
SD	0.00045	0.00059	0.00070	0.00151	0.00186	0.00261	0.00216	-

TABLE VI
THE PERFORMANCE DERIVATIVES IN EXAMPLE 2 WITH 1 000 000 TRANSITIONS

n	1	2	3	5	10	15	20	Theoretic
Mean	-0.0989	-0.1229	-0.1167	-0.1176	-0.1178	-0.1176	-0.1174	-0.1176
SD	0.0009	0.00015	0.00016	0.00025	0.00026	0.00047	0.00059	-

These tables show that the estimate is quite accurate even when n is as small as two or three. The standard deviation is acceptable even if n is 20. Thus, the results are not so sensitive to the value of n. It is interesting to note that even if we choose $n = 1$ in this case, the error is only about 17%. $n = 1$ means using the performance function to approximate the potentials, i.e., assuming $g \approx f$. This corresponds to the "myopic" policy in optimization: when the system jumps to state i, we just use the one step performance $f(i)$ to represent the long-term performance.

Table VII lists the potentials g [in the form of (16), with $\pi g = 0$] estimated with $n = 5$ and simulation length 100 000. □

One disadvantage of Algorithms 1c and 2c is that they have to estimate the potential for every state. This is sometimes difficult for a number of reasons: the number of states may

TABLE VII
THE POTENTIALS IN EXAMPLE 2 WITH 100 000 TRANSITIONS AND $n = 5$

i	1	2	3	4	5	6	7	8	9	10
Theoretic	1.865	-4.025	-5.121	6.268	-3.259	-13.553	-1.997	14.098	-10.033	9.614
Mean	1.845	-4.056	-5.132	6.243	-3.266	-13.520	-1.893	14.162	-9.902	9.654
SD	0..098	0.088	0.163	0.140	0.140	0.187	0.116	0.110	0.185	0.160

be too large; some states may be visited very rarely; and for systems with special structures (e.g., queueing networks), it may not be convenient even to list out all the states. In the following, we develop an algorithm that can be used to estimate the performance derivatives directly without estimating each individual potential.

An analog is the estimation of the performance measure. There are two ways to get the estimation: we may estimate all π_i first and then use $\eta = \pi f$ to calculate the performance, or we may estimate η directly by

$$\eta = \lim_{N\to\infty} \frac{1}{N+1} \sum_{k=0}^{N} f(X_k), \qquad w.p.1. \tag{24}$$

We want to develop an algorithm similar to (24) for $\partial\eta/\partial Q$.

Each term in πQg takes the form $\pi_i q_{ij} g_j$. Because it does not seem straightforward to estimate this term directly on a sample path, we propose to use a standard technique in simulation, *importance sampling*. First, we observe that $\pi_i p_{ij} g_j$ can be estimated according to the following equation:

$$\pi_i p_{ij} g_j = \lim_{N\to\infty} \frac{1}{N-n+1} \cdot \left\{ \sum_{k=0}^{N-n} \epsilon^i(X_k)\epsilon^j(X_{k+1}) \left[\sum_{l=0}^{n-1} f(X_{k+l+1}) \right] \right\} \quad w.p.1. \tag{25}$$

To prove this equation, we use the same argument as for (23). Define a function $Z_k = \epsilon^i(X_k)\epsilon^j(X_{k+1})[\sum_{l=0}^{n-1} f(X_{k+l+1})]$, we get an ergodic process $Z = \{Z_k, k \geq 0\}$. Therefore, the right-hand side of (25) equals $E_{\pi_Z}\{\epsilon^i(X_k)\epsilon^j(X_{k+1})[\sum_{l=0}^{n-1} f(X_{k+l+1})]\}$, where E_{π_Z} is the expectation with respect to the steady-state probability measure of Z, which equals

$$E_{\pi_Z}\left\{ \sum_{l=0}^{n-1} f(X_{k+l+1}) | \epsilon^i(X_k)\epsilon^j(X_{k+1}) = 1 \right\} \cdot p^*[\epsilon^i(X_k)\epsilon^j(X_{k+1}) = 1]$$

where p^* is the steady-state probability of $X_k = i$ and $X_{k+1} = j$. By the Markov property, the first term equals g_j, and the second term equals $\pi_i p_{ij}$.

From (25), we have

$$\pi_i q_{ij} g_j = \lim_{N\to\infty} \frac{1}{N-n+1} \cdot \left\{ \sum_{k=0}^{N-n} \epsilon^i(X_k)\epsilon^j(X_{k+1}) \frac{q_{ij}}{p_{ij}} \left[\sum_{l=0}^{n-1} f(X_{k+l+1}) \right] \right\} \quad w.p.1.$$

In the above, the quantity involving q_{ij}, $\pi_i q_{ij} g_j$, is estimated by simulating a quantity involving p_{ij}, $\pi_i p_{ij} g_j$. This is a variant of the standard important sampling technique in simulation, which is widely applied to study the performance of a stochastic system with a probability distribution by simulating another stochastic system with a different probability distribution.

Finally, we have

$$\begin{aligned} \frac{\partial\eta}{\partial Q} &= \pi Q g \\ &= \lim_{N\to\infty} \frac{1}{N-n+1} \sum_i \sum_j \cdot \left\{ \sum_{k=0}^{N-n} \epsilon^i(X_k)\epsilon^j(X_{k+1}) \frac{q_{ij}}{p_{ij}} \left[\sum_{l=0}^{n-1} f(X_{k+l+1}) \right] \right\} \\ &= \lim_{N\to\infty} \frac{1}{N-n+1} \left\{ \sum_{k=0}^{N-n} \sum_i \sum_j \cdot \left\{ \epsilon^i(X_k)\epsilon^j(X_{k+1}) \frac{q_{ij}}{p_{ij}} \right\} \left[\sum_{l=0}^{n-1} f(X_{k+l+1}) \right] \right\} \\ &= \lim_{N\to\infty} \frac{1}{N-n+1} \left\{ \sum_{k=0}^{N-n} \left\{ \frac{q_{X_k, X_{k+1}}}{p_{X_k, X_{k+1}}} \right\} \cdot \left[\sum_{l=0}^{n-1} f(X_{k+l+1}) \right] \right\}, \qquad w.p.1. \end{aligned} \tag{26}$$

This equation suggests an algorithm, which is as simple as (23), for estimating the performance derivative. Compared with (23), the modification is: when the system jumps from state i to state j, a modifying factor q_{ij}/p_{ij} is multiplied to the term $\sum_{l=0}^{n-1} f(X_{k+l+1})$.

Algorithm 3c:

1) Estimate $\partial\eta/\partial Q$ by using (26).

Example 3: We repeat the simulation for the same Markov system as in Example 2 by applying Algorithm 3c. We did ten simulation runs for each value of n and the results are listed in Table VIII. □

Before ending this section, we make some comparisons of the algorithms developed for Markov chains. Algorithm 1c is unbiased and Algorithms 2c and 3c are biased for any finite n. Among the three algorithms, Algorithm 1c has the least standard deviation. The standard deviation of Algorithm 2c is in general small, and that of Algorithm 3c is larger than the other two algorithms. These observations are evidenced by the simulation results in the examples and can be explained intuitively. From (15), in Algorithm 1c, d_{ij} is estimated as

TABLE VIII
THE PERFORMANCE DERIVATIVES IN EXAMPLE 3 WITH 1 000 000 TRANSITIONS

n	1	2	3	5	10	15	20	Theoretic
Mean	-0.0977	-0.1224	-0.1166	-0.1169	-0.1166	-0.1165	-0.1163	-0.1176
SD	0.0007	0.0014	0.0017	0.0030	0.0056	0.0079	0.0108	-

a mean of the sum of a number of random variables, and the potentials are obtained from d_{ij} by π_j with $\sum_j \pi_j = 1$. In Algorithm 2c, the values of $f(X_{k+l})$, with $l > L^{\{j\}}(i)$, contribute to a term with mean zero, and they just increase the variance. Algorithm 3c uses less information from the sample path than Algorithm 2c: in Algorithm 2c all the terms $\sum_{l=0}^{n-1} f(X_{k+l})$ starting from $X_k = j$ are used for estimating g_j, while in Algorithm 3c, only those terms starting from a state transition from i to j are used to estimate g_j in $\pi_i q_{ij} g_j$. In addition, applying importance sampling increases the variance: the exact quantity q_{ij} is replaced through a random variable p_{ij}.

Nevertheless, Algorithms 2c and 3c are more convenient to use than Algorithm 1c. Algorithm 3c estimates the derivative without recording any intermediate values of π_i and g_i, $i = 1, 2, \cdots, M$. This feature is particular important when the state space is large. Compared with Algorithm 2c, Algorithm 3c may save some computation when estimating the derivative with respect to a given Q. However, with algorithm 2c, the derivatives with respect to many Q's can be obtained almost simultaneously by simply calculating $\pi Q g$ after g is estimated; this is not true for Algorithm 3c.

Finally, (26) is equivalent to

$$\frac{\partial \eta}{\partial Q} = \lim_{N\to\infty} \frac{1}{N-n+1} \cdot \left\{ \sum_{k=0}^{N-n} \left[f(X_{k+n}) \sum_{l=0}^{n-1} \left(\frac{q_{X_{k+l}, X_{k+l+1}}}{p_{X_{k+l}, X_{k+l+1}}} \right) \right] \right\}, \quad w.p.1.$$

This expression is the same as the ensemble average importance sampling (EAIS) estimate of the performance sensitivity of a Markov process introduced in [5]. This is not surprising because the importance sampling technique is indeed used in Algorithm 3c to estimate $\pi_i q_{ij} g_j$. This also explains why Algorithm 3c has a relatively big variance. The variances of EAIS estimates, however, are in general smaller than the time average-based importance sampling estimates. A number of examples applying this method to queueing networks and a discussion about the variance of this technique can be found in [5].

IV. ALGORITHMS FOR MARKOV PROCESSES

In this section, we develop performance sensitivity algorithms for Markov processes. Algorithms 1c and 2c can be easily modified to handle Markov processes, but developing an algorithm similar to Algorithm 3c for Markov processes requires some special considerations.

First, the algorithm similar to Algorithms 1c is based on (2). The continuous-time analogue of (18), (19), (21), and (22) are obtained by changing the summations into corresponding integrations.

Algorithm 1p:

1) Estimate $E[\int_0^{S^{\{j\}}(i)} f(X_t^{\{j\}})\, dt]$, $E[S^{\{j\}}(i)]$, and η on a single sample path.

2(–4) Same as Steps 2) to 4) in Algorithm 1c.

Next, we let T_k be the kth transition epoch of $\{X_t\}$; S_k be its kth sojourn time, and X_k be its state after the kth transition. By definition, $S_k = T_{k+1} - T_k$ and $X_k = X_t|_{t=T_k^+}$. Algorithm 2 for Markov processes is based on

$$\pi_i = \lim_{N\to\infty} \frac{1}{T_N} \left\{ \sum_{k=0}^{N-1} \epsilon^i(X_k) S_k \right\}, \qquad w.p.1 \tag{27}$$

and

$$g_i(T) = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-n} \{\epsilon^i(X_k) \int_{T_k}^{T_k+T} f(X_t)\, dt\}}{\sum_{k=0}^{N-n} \epsilon^i(X_k)}, \qquad w.p.1 \tag{28}$$

where T is a properly chosen constant.

Algorithm 2p:

1) Estimate the steady-state probabilities π and potentials g by using (27) and (28).
2) Calculate $\partial\eta/\partial Q$ by (13).

The difficulty in developing the third algorithm comes from applying the importance sampling technique. Observe that Algorithm 3 is based on a weighted ratio q_{ij}/p_{ij}, which only accounts for the effect of changes in routing probabilities among states. For Markov processes, there can be changes in the total infinitesimal rates of states, in addition to the changes in routing probability. In general, the infinitesimal generator A takes the following form:

$$a_{ij} = \begin{cases} -\lambda(i), & \text{if } i = j \\ \lambda(i) p_{ij}, & \text{if } i \neq j \end{cases}$$

where $\lambda(i)$ is the infinitesimal rate at state i, and p_{ij} represents the routing probability when the process jumps out of state i. In the following, we will first consider the effect of changes in routing probabilities p_{ij}, and then we consider the effect of changes in state rates $\lambda(i)$. After that, we will consider the combined effect of both changes. Finally, we will discuss the case in which Q is specified instead of routing and rates changes.

First consider the sole effect of changes in routing. Let $P = [p_{ij}]$ be the transition probability matrix of the Markov chain embedded in jumping epochs. Suppose P changes to $P_\delta = P + \delta U$, where $Ue = 0$, and $u_{ij} \geq 0$ if $p_{ij} = 0$. Let $A = [a_{ij}]$ and $A^{(p)} = [a_{ij}^{(p)}]$ be the infinitesimal generators before and after the change in P, i.e., $a_{ii} = -\lambda(i)$, $a_{ij} = \lambda(i)p_{ij}$ for $i \neq j$; $a_{ii}^{(p)} = a_{ii}$, $a_{ij}^{(p)} = \lambda(i)(p_{ij} + \delta u_{ij})$ for $i \neq j$. Hence the elements of $Q^{(p)} = [q_{ij}^{(p)}]$, the change in A due to change in P, are

$$q_{ii}^{(p)} = 0,$$

and

$$q_{ij}^{(p)} = \lambda(i)u_{ij}, \qquad \text{for } i \neq j, \quad i, j \in \mathcal{E}. \tag{29}$$

By defining $Q^{(p)}$ in this way, the effect of the change in the routing probability matrix P is captured by $Q^{(p)}$, the directional change of the infinitesimal generator A; we will denote the derivative of η with respect to such a directional change by $\partial\eta/\partial Q^{(p)}$. In subsequent discussion, the effect of the change in the service rates of states is captured by $Q^{(\lambda)}$, the corresponding directional change of the infinitesimal generator A. We will denote the derivative of η with respect to such a directional change by $\partial\eta/\partial Q^{(\lambda)}$.

Similar to (25), for $i \neq j$, we have

$$\pi_i p_{ij} g_j = \lim_{N\to\infty} \frac{1}{T_{N-n+1}} \left\{ \sum_{k=0}^{N-n} \epsilon^i(X_k)\epsilon^j(X_{k+1}) \cdot S_k \left[\int_{T_{k+1}}^{T_{k+1}+T} f(X_t)\, dt \right] \right\}, \qquad w.p.1.$$

Equivalently

$$\pi_i q_{ij}^{(p)} g_j = \lim_{N\to\infty} \frac{1}{T_{N-n+1}} \left\{ \sum_{k=0}^{N-n} \epsilon^i(X_k)\epsilon^j(X_{k+1}) \cdot S_k \frac{\lambda(i)u_{ij}}{p_{ij}} \left[\int_{T_{k+1}}^{T_{k+1}+T} f(X_t)\, dt \right] \right\}, \qquad w.p.1$$

and

$$\frac{\partial\eta}{\partial Q^{(p)}} = \lim_{N\to\infty} \frac{1}{T_{N-n+1}} \sum_{k=0}^{N-n} \cdot \left\{ S_k \frac{\lambda(X_k)u_{X_kX_{k+1}}}{p_{X_kX_{k+1}}} \left[\int_{T_{k+1}}^{T_{k+1}+T} f(X_t)\, dt \right] \right\}, \quad w.p.1. \tag{30}$$

An alternative to (30) is to use a uniformized embedded Markov chain so that all integrations can be replaced by summations. This approach applies when we simulate the Markov process. Since the uniformized embedded chain contains transitions from a state to itself, which are not observable from a given sample path, the uniformized chain approach cannot be used for a real system.

Now consider the sole effect of changes in infinitesimal rates of states. Suppose $\lambda(i)$ changes to $\lambda(i) + \delta\rho(i)$, where $\rho(i) \geq 0$ if $\lambda(i) = 0$. By writing out elements of A before and after these changes as in the previous case, we can show that the elements of $Q^{(\lambda)} = [q_{ij}^{(\lambda)}]$, the change in A due to changes in rates, are given by

$$q_{ii}^{(\lambda)} = -\rho(i)$$

and

$$q_{ij}^{(\lambda)} = \rho(i)p_{ij}, \qquad \text{for } i \neq j, \quad i, j \in \mathcal{E}. \tag{31}$$

We may proceed as in Markov chain, though there is a shorter path through (11).

Observe that the kth row of $Q^{(\lambda)}$ in this case equals $\rho(k)/\lambda(k)$ times the kth row of A. Let $\nu_k = [\rho(k)/\lambda(k)]\pi_k$ and $\nu = (\nu_1, \cdots, \nu_M)$. We have $\pi Q^{(\lambda)} = \nu A$. Using (11)

$$\begin{aligned} \frac{\partial\eta}{\partial Q^{(\lambda)}} &= -\nu A A^{\#} f = -\nu(I - e\pi)f = \nu(\eta e - f) \\ &= \sum_{k=1}^{M} \left\{ \frac{\rho(k)}{\lambda(k)} \pi_k [\eta - f_k] \right\} \\ &= \eta \left\{ \sum_{k=1}^{M} \left[\frac{\rho(k)}{\lambda(k)} \pi_k \right] \right\} - \sum_{k=1}^{M} \left\{ \frac{\rho(k)}{\lambda(k)} \pi_k f_k \right\}. \end{aligned} \tag{32}$$

The two terms in the summations of (32) can be easily estimated directly from a single sample path. We have

$$\sum_{k=1}^{M} \left\{ \frac{\rho(k)}{\lambda(k)} \pi_k \right\} = \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \left\{ \frac{\rho(X_k)}{\lambda(X_k)} S_k \right\}, \qquad w.p.1. \tag{33}$$

and

$$\sum_{k=1}^{M} \left\{ \frac{\rho(k)}{\lambda(k)} \pi_k f_k \right\} = \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \left\{ \frac{\rho(X_k)}{\lambda(X_k)} S_k f(X_k) \right\}, \qquad w.p.1. \tag{34}$$

It is interesting to note that the derivative (32) is expressed independent of g; thus, the estimate based on (34) and (33) does not depend on T, which is chosen in estimating g.

In general, if both $\lambda(i)$'s and P change, we can ignore the (higher order) composition effect of routing and rate changes, and decompose Q into $Q^{(p)}$ and $Q^{(\lambda)}$, where elements of $Q^{(p)}$ are from (29) and elements of $Q^{(\lambda)}$ are from (31). $\partial\eta/\partial Q$ can be obtained by (30), (32), and

$$\frac{\partial\eta}{\partial Q} = \pi Q g = \pi Q^{(p)} g + \pi Q^{(\lambda)} g = \frac{\partial\eta}{\partial Q^{(p)}} + \frac{\partial\eta}{\partial Q^{(\lambda)}}. \tag{35}$$

Sometimes, we are given a Q without explicitly specifying the routing and rate changes. In such a case, we can still decompose Q into $Q^{(p)}$ and $Q^{(\lambda)}$. Let $\rho(i) = -q_{ii}$. Then $Q^{(\lambda)}$ is found from these $\rho(i)$'s through (31), and $Q^{(p)}$ is found back from

$$Q^{(p)} = Q - Q^{(\lambda)}. \tag{36}$$

Since the original infinitesimal rate out of state i is $\lambda(i)$, the change in routing from state i to state j is given by

$$u_{ij} = \frac{q_{ij}^{(p)}}{\lambda(i)}. \tag{37}$$

TABLE IX
THE PERFORMANCE DERIVATIVES IN EXAMPLE 4 WITH 1 000 000 TRANSITIONS

T		1	2	3	5	10	15	20	Theoretic
$\frac{\partial \eta}{\partial Q^{(\lambda)}}$	Mean	-0.0336							-
	SD	0.0000 (> 0)							-
$\frac{\partial \eta}{\partial Q^{(p)}}$	Mean	-0.0390	-0.0554	-0.0663	-0.0784	-0.0860	-0.0868	-0.0882	-
	SD	0.0026	0.0037	0.0046	0.0050	0.0070	0.0105	0.0116	-
$\frac{\partial \eta}{\partial Q}$	Mean	-0.0727	-0.0890	-0.0999	-0.1120	-0.1196	-0.1203	-0.1218	-0.1176
	SD	0.0026	0.0037	0.0046	0.0050	0.0070	0.0105	0.0116	-

One can verify that $Q^{(p)}$ and $Q^{(\lambda)}$ found above indeed bear the intended physical meaning and $\partial\eta/\partial Q$ can then follow from (30), (32), and (35).

Algorithm 3p:

1) Decompose Q into $Q^{(p)}$ and $Q^{(\lambda)}$ either from (29) and (31) when $\rho(i)$ and u_{ij} are given, or by setting $\rho(i) = -q_{ii}$ and using (31), (36), and (37) when Q is given instead.
2) Estimate $\partial\eta/\partial Q^{(p)}$ by (30), and $\partial\eta/\partial Q^{(\lambda)}$ by (32)–(34).
3) Calculate $\partial\eta/\partial Q$ by (35).

We have carried out the simulation of Algorithms 1p and 2p for a ten-state Markov process with the embedded Markov chain given by the transition probability matrix P in Example 1. The results are not presented here, since, with the same number or runs, results from the Markov process behave like those from the chain. The standard deviation of d_{ij}'s from the Markov process tends to be larger. Such a result is expected, since the random sojourn times in the Markov process induce variability. We can reduce this variability by increasing the number of runs. In the following, we give the results of the Algorithm 3p on the Markov process mentioned at the beginning of this paragraph.

Example 4: We carry out 1 000 000 transitions. The result of $\partial\eta/\partial Q^{(p)}$, $\partial\eta/\partial Q^{(\lambda)}$ and of $\partial\eta/\partial Q$ are shown in Table IX. Since the estimates in (33) and (34) do not depend on T, neither does $\partial\eta/\partial Q^{(\lambda)}$.

V. CLOSED QUEUEING NETWORKS

On first sight the sensitivity estimate of closed queueing networks (CQN's) is not worthwhile to discuss in detail. With exponential service times, the CQN can be modeled as a Markov process, and its sensitivity analysis follows from Section IV. On second thought, the real issue lies in practicality, not mathematical equivalence. In real life, it is impossible or impractical to specify A, the infinitesimal generator of a CQN. More importantly, in controlling a queueing system, we usually only refer to characteristics of individual stations, but not the state of the system, which usually involves all stations. As a result, we seldom specify Q, but, instead, specify changes in routing probabilities among stations and in service (infinitesimal) rates of stations. We would like to know the effect of changing these quantities to the performance measure. To apply algorithms in Section IV, we need to transform quantities specified on stations to states of the underlying Markov process. In the following, we demonstrate our procedure on Algorithm 3p, which, as discussed before, only needs to collect information on a single sample path.

Consider an exponential CQN with M stations. Let $\mu_i, 1 \le i \le M$, be the service rate of the ith station; $\mathbf{n}_t = [\mathbf{n}_t(1), \cdots, \mathbf{n}_t(M)]$ be the state of the system, where $n_t(i)$ is the number of customers at station i at epoch t. Between the $(k-1)$th and the kth transitions, the state of the system is denoted by $\mathbf{n}_k = [\mathbf{n}_k(1), \cdots, \mathbf{n}_k(M)]$. The station which has a service completion at the kth transition is denoted by c_k, while the station which has an arrival right after the kth transition is denoted by a_k. The state $\mathbf{n}_{(\cdot)}(i,j)$ is reached when a customer moves from station i to station j for state $\mathbf{n}_{(\cdot)}$. Based on our notation, $\mathbf{n}_k(c_k) = \mathbf{n}_{k+1}(c_k) + 1$, $\mathbf{n}_k(a_k) + 1 = \mathbf{n}_{k+1}(a_k)$, and $\mathbf{n}_k(c_k, a_k) = \mathbf{n}_{k+1}$. The performance function at state $\mathbf{n}$ is denoted by $f(\mathbf{n})$, which is assumed to be equal to $\sum_{i=1}^{M} f_i[n(i)]$, where f_i is the performance function for the ith station. This assumption does not constrain our procedure, which can handle various practical problems by adopting different f_i's. The routing probability matrix of customers among stations is denoted by $R = [r_{ij}]$ under which the Markov process is assumed to be irreducible. Other notations defined in earlier sections are used here with necessary modification in interpretation; e.g., $\lambda(\mathbf{n})$ becomes the infinitesimal rate of state $\mathbf{n}$, while $\lambda(i)$ is the service rate of the ith station.

When the context is clear, or when we aim at a generic expression, we will suppress the subscript k, which indicates the dependence on transitions. For example, r_{ca} is a short form of $r_{c_k a_k}$ when the dependence on the kth transition is not the main issue; similarly, $\mathbf{n}(c, a)$ stands for $\mathbf{n}_k(c_k, a_k)$.

With the above notation, the infinitesimal rate out of state $\mathbf{n}$ is $\lambda(\mathbf{n}) = \sum_{\{l|\mathbf{n}(l)>0\}} \lambda(l)$; the sojourn time for the system to stay in state $\mathbf{n}$ is $S(\mathbf{n})$, which distributes as $\exp[\lambda(\mathbf{n})]$, and $p_{\mathbf{n},\mathbf{n}(c,a)} = \lambda(c) r_{ca}/\lambda(\mathbf{n}) = \lambda(c)/\sum_{\{l|\mathbf{n}(l)>0\}\lambda(l)} r_{ca}$. Suppose $\lambda(i)$ changes to $\lambda(i) + \delta\rho(i)$ and r_{ij} to $r_{ij} + \delta u_{ij}, 1 \le i, j \le M$. The change of rate in state $\mathbf{n}$ is $\rho(\mathbf{n}) = \sum_{\{l|\mathbf{n}(l)>0\}} \rho(l)$. As in (31)

$$q^{(\lambda)}_{\mathbf{n},\mathbf{n}} = -\rho(\mathbf{n}) = -\sum_{\{l|\mathbf{n}(l)>0\}} \rho(l)$$

and

$$q^{(\lambda)}_{\mathbf{n},\mathbf{n}(c,a)} = \rho(\mathbf{n}) p_{\mathbf{n},\mathbf{n}(c,a)} = \frac{\rho(\mathbf{n})}{\lambda(\mathbf{n})}\lambda(c) r_{ca} = \frac{\sum_{\{l|\mathbf{n}(l)>0\}} \rho(l)}{\sum_{\{l|\mathbf{n}(l)>0\}} \lambda(l)} \lambda(c) r_{ca}. \tag{38}$$

It is clear that

$$q^{(p)}_{\mathbf{n},\mathbf{n}} = 0.$$

By ignoring higher order terms, (36) leads to

$$q^{(p)}_{\mathbf{n},\mathbf{n}(c,a)} = \{\rho(c) r_{ca} + \lambda(c) u_{ca}\} - q^{(\lambda)}_{\mathbf{n},\mathbf{n}(c,a)}. \tag{39}$$

From (30), see (40) shown at the bottom of the page. The estimate of (40) on sample path is a simple extension of (30), since $\mathbf{n}_k$, c_k, and a_k are known quantities on each transition. Similarly, (32) gives

$$\begin{aligned}\frac{\partial \eta}{\partial Q^{(\lambda)}} =& \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} S_k f(\mathbf{n}_k) \right\} \\ & \cdot \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \frac{\rho(\mathbf{n}_k)}{\lambda(\mathbf{n}_k)} S_k \right\} \\ & - \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \frac{\rho(\mathbf{n}_k)}{\lambda(\mathbf{n}_k)} S_k f(\mathbf{n}_k) \right\} \\ =& \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \sum_{i=1}^{M} S_k f_i[\mathbf{n}_k(i)] \right\} \\ & \cdot \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \frac{\sum_{\{l|\mathbf{n}_k(l)>0\}} \rho(l)}{\sum_{\{l|\mathbf{n}_k(l)>0\}} \lambda(l)} S_k \right\} \\ & - \left\{ \lim_{N\to\infty} \frac{1}{T_N} \sum_{k=0}^{N-1} \frac{\sum_{\{l|\mathbf{n}_k(l)>0\}} \rho(l)}{\sum_{\{l|\mathbf{n}_k(l)>0\}} \lambda(l)} \right. \\ & \left. \cdot S_k \sum_{i=1}^{M} f_i[\mathbf{n}_k(i)] \right\}, \quad w.p.1 \end{aligned} \tag{41}$$

where all terms can be easily estimated on a sample path.

TABLE X
THE PERFORMANCE DERIVATIVES IN EXAMPLE 5

T=20		number of transitions			Theoretical
		100,000	1,000,000	2,000,000	
station 1	mean	1.5633	1.5760	1.5778	1.6019
	SD	0.0693	0.0111	0.0068	
station 2	mean	-4.1886	-4.2331	-4.228	-4.4376
	SD	0.1108	0.0317	0.0090	
station 3	mean	2.7310	2.7332	2.7344	2.8358
	SD	0.0862	0.0206	0.0058	

TABLE XI
THE PERFORMANCE DERIVATIVES IN EXAMPLE 6

T=20		number of transitions			Theoretical
		100,000	1,000,000	2,000,000	
station 1	mean	0.3071	0.3092	0.3095	0.3155
	SD	0.0136	0.0024	0.0011	

Example 5: We work with a small exponential CQN whose actual performance derivatives can easily be found. Consider a three-server four-customer CQN such that $r_{ii} = 0$, $r_{12} = r_{13} = r_{21} = r_{23} = 0.5$, $r_{31} = 0.625$, and $r_{32} = 0.375$. The mean service times of the three stations are 7, 2.5, and 2 time units, respectively. We simulate for three different numbers of transitions, each with ten replications, aiming to determine the change in expected number of customers in the three stations with respect to the change in rate of the second station. With such an objective, $f_i(j) = j$ for all i and j. See Table X for simulation settings and results.

Since the computation for the CQN is actually based on Algorithm 3p of Markov processes, the results behave similar to those of Example 4. The means of the performance derivatives are close to the theoretical values for a small number of runs (e.g., 100 000) and a small T (e.g., 20). The standard derivations of our estimates are reasonably small. □

The next example shows that our algorithms can be applied to problems that IPA cannot solve.

Example 6: In this example, we study the probability that the number of customers in a station exceeds a threshold. Consider the same CQN as in Example 5. We take $\eta = P_\pi\{\mathbf{n}_1 \geq 3\}$ and we want to determine $\partial\eta/\partial\lambda(2)$. As in

$$\begin{aligned}\frac{\partial \eta}{\partial Q^{(p)}} &= \lim_{N\to\infty} \frac{1}{T_{N-n+1}} \sum_{k=0}^{N-n} \left\{ S_k(\mathbf{n}_k) \frac{q^{(p)}_{\mathbf{n}_k,\mathbf{n}_{k+1}}}{p_{\mathbf{n}_k,\mathbf{n}_{k+1}}} \int_{T_{k+1}}^{T_{k+1}+T} f(\mathbf{n}_t)\, dt \right\} \\ &= \lim_{N\to\infty} \frac{1}{T_{N-n+1}} \sum_{k=0}^{N-n} \left\{ S_k(\mathbf{n}_k) \frac{\lambda(\mathbf{n}_k)[\rho(c_k) r_{c_k a_k} + \lambda(c_k) u_{c_k a_k}] - \rho(\mathbf{n}_k)\lambda(c_k) r_{c_k a_k}}{\lambda(c_k) r_{c_k a_k}} \int_{T_{k+1}}^{T_{k+1}+T} \sum_{i=1}^{M} f_i[\mathbf{n}_t(i)]\, dt \right\} \\ & \quad w.p.1. \end{aligned} \tag{40}$$

previous examples, we translate the change in $\lambda(2)$ into $Q^{(\lambda)}$, the directional change in A. Define $f_1(0) = f_1(1) = f_1(2) = 0$, $f_1(3) = f_1(4) = 1$, and $f_i(j) = 0$ for $i \neq 1$. It is clear that $\eta = E_\pi(f)$; hence (40) and (41) can be directly applied.

We repeat the simulation of the Markov process shown in Example 5 for the new f. The results are shown in Table XI. □

VI. Conclusion

In this paper we give several algorithms to compute performance derivatives for Markov chains and Markov processes. The algorithms are based on a single sample path of the Markov chain or Markov process. Compared with the existing approaches, our approach has its own advantages. Unlike IPA, which does not apply for systems where the performance functions are discontinuous, our approach applies to a wide range of systems and performance measures. Compared with the likelihood ratio method (see, e.g., [15] and [22]) or the score function method (e.g., [23]), which usually resorts to regenerative structure to reduce the variance and does not apply when the changes in the transition matrix involves open arcs (i.e., perturbing the zero entries in a transition matrix), our approach does not have the same restrictions. In addition, the computation involved is mainly adding up the performance values in some intervals, which is similar to IPA. After the estimates of potentials are obtained, the performance derivatives with respect to many Q's can be obtained by simple calculation using πQg. Thus, compared with the brute force simulation method, our approach is more efficient when one wants to estimate many performance derivatives; this is the case for performance optimization problems. Furthermore, compared with smoothed perturbation analysis (SPA) our approach does not need complicated analysis that is problem dependent; this is important from a practical point of view, since engineers may not need special knowledge to implement the approach in their real world problems.

Our empirical results show that the standard deviations are generally small, and that in Algorithms 1c, 2c, 1p, and 2p, it suffices to use small values of n and T (≈ 5) to estimate the potential and hence the derivatives. While optimal values of n and T are problem dependent and may increase with the size of the state space, the variability of Q, etc., the typical geometric tails of stationary distributions make us believe that our algorithms, especially Algorithms 3c and 3p, can be applied to problems of practical sizes. Our work also creates some new research topics, such as state aggregation for systems with large sizes.

Finally, as demonstrated by the examples, to get accurate estimates we need to run the system for a large number of transitions. This is common in estimating mean values in stochastic systems. However, in many modern systems such as high-speed communication networks, millions of transitions may happen within seconds, and therefore, the application of our approach to real-world systems is feasible.

Another strong support to our approach is the recently developed concept of *ordinal optimization* [18]. The main idea is that by softening the goal of optimization and by comparing different schemes ordinally instead of obtaining the exact performance values, we can dramatically reduce our demand in the accuracy of the estimates. We believe that using ordinal optimization, relatively short sample paths can be used to estimate the performance derivatives to achieve the goal of performance optimization.

References

[1] L. Breiman, *Probability.* Reading, MA: Addison-Wesley, 1968.

[2] P. Brémaud, "Maximal coupling and rare perturbation sensitivity analysis," *Queueing Syst.: Theory and Applicat.,* vol. 10, pp. 249–270, 1992.

[3] C. A. Brooks and P. Varaiya, "Using perturbation analysis to solve the capacity and flow assignment problem for general and ATM networks," *IEEE Globcom,* 1994.

[4] Xi-Ren Cao and Han-Fu Chen, "Potentials, perturbation realization, and sensitivity analysis of Markov processes," *IEEE Trans. Automat. Contr.,* vol. 42, pp. 1382–1393, 1997.

[5] X. R. Cao, "Uniformization and performance sensitivity estimation in closed queueing networks," *Math. Computer Modeling,* vol. 23, pp. 77–92, 1996.

[6] M. Caramanis and G. Liberopoulos, "Perturbation analysis for the design of flexible manufacturing system flow controllers," *Operations Res.,* vol. 40, pp. 1107–1125, 1992.

[7] C. G. Cassandras, *Discrete-Event Systems: Modeling and Performance Analysis.* Homewood, IL: Aksen Associates, Inc., 1993.

[8] C. G. Cassandras and S. G. Strickland, "On-line sensitivity analysis of Markov chains," *IEEE Trans. Automat. Contr.,* vol. 34, pp. 76–86, 1989.

[9] E. K. P. Chong and P. J. Ramadge, "Optimization of queues using infinitesimal perturbation analysis-based algorithms with general update times," *SIAM J. Contr. Optimiz.,* vol. 31, pp. 698–732, 1993.

[10] L. Y. Dai, "Sensitivity analysis of stationary performance measures for general Markov chains; Special issue on discrete-event dynamic systems," *Math. Computer Modeling,* vol. 23, no. 11/12, 1996.

[11] ———, "Derivative estimation of discrete-time Markov chains via finite-difference approximation with common random numbers," to appear in *SIAM J. Appl. Math.,* 1997.

[12] L. Y. Dai and Y. C. Ho, "Structural infinitesimal perturbation analysis (SIPA) for derivative estimation of discrete-event dynamic systems," *IEEE Trans. Automat. Contr.,* vol. 40, pp. 1154–1166, 1995.

[13] M. C. Fu and J. Q. Hu, "Extensions and generalizations of smoothed perturbation analysis in a generalized semi-Markov process framework," *IEEE Trans. Automat. Contr.,* vol. 37, pp. 1483–1500, 1992.

[14] P. Glasserman, *Gradient Estimation Via Perturbation Analysis.* Boston, MA: Kluwer, 1991.

[15] P. W. Glynn, "Likelihood ratio gradient estimation: An overview," in *Proc. WSC'87,* 1897, pp. 366–375.

[16] W. B. Gong and Y. C. Ho, "Smoothed perturbation analysis for discrete-event dynamic systems," *IEEE Trans. Automat. Contr.,* vol. 32, pp. 858–866, 1987.

[17] A. Haurie, P. L'Ecuyer, and C. van Delft, "Convergence of stochastic approximation coupled with perturbation analysis in a class of manufacturing flow control models," *Discrete-Event Dynamic Syst.: Theory Applicat.,* vol. 4, pp. 87–111, 1994.

[18] Y. C. Ho, "Soft optimization for hard problems," computerized lecture via private communication, 1996.

[19] Y. C. Ho and X. R. Cao, *Perturbation Analysis of Discrete-Event Dynamic Systems.* Boston, MA: Kluwer, 1991.

[20] Y. C. Ho, X. R. Cao, and C. G. Cassandras, "Infinitesimal and finite perturbation analysis for queueing networks," *Automatica,* vol. 19, pp. 439–445, 1983.

[21] J. Keilson, *Markov Chain Models—Rarity and Exponentiality.* New York: Springer-Verlag, 1979.

[22] M. I. Reiman and A. Weiss, "Sensitivity analysis for simulation via likelihood ratios," *Operations Res.,* vol. 37, pp. 830–844, 1989.

[23] R. Y. Rubinstein and A. Shapiro, *Discrete-Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.* New York: Wiley, 1993.

[24] R. Suri and M. A. Zazanis, "Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue," *Management Sci.,* vol. 34, pp. 39–64, 1988.

[25] P. Vakili, "A standard clock technique for efficient simulation," *Operations Res. Lett.,* vol. 10, pp. 445–452, 1991.

[26] N. Xiao, F. F. Wu, and S. M. Lun, "Dynamic bandwidth allocation using infinitesimal perturbation analysis," in *IEEE Infocom'94,* 1994, pp. 383–389.

[27] H. Yan and X. Y. Zhou, "Finding optimal number of kanbans in a manufacturing system via perturbation analysis," in *Lecture Notes in Control and Information Sciences.* New York: Springer-Verlag, vol. 199, pp. 572–578, 1994.

Xi-Ren Cao (S'82–M'84–SM'89–F'96) received the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1981 and 1984, respectively.

He was a Research Fellow at Harvard from 1984 to 1986. He then worked as a Principal and Consultant Engineer at Digital Equipment Corporation, until October 1993. Since then, he has been a Professor of the Hong Kong University of Science and Technology. He has published two books: *Realization Probabilities—the Dynamics of Queueing Systems* (Berlin, Germany: Springer-Verlag, 1994) and *Perturbation Analysis of Discrete-Event Dynamic Systems* (Dordrecht, The Netherlands: Kluwer, 1991). His current research interests include discrete-event systems, computer and communication networks, stochastic processes, and optimization techniques.

Dr. Cao received the Outstanding Transactions Paper Award from the IEEE Control Systems Society in 1987 and the Outstanding Publication Award from the Institution of Management Science in 1990. He served as Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY; he is currently Associate Editor of the *Journal of Discrete-Event Dynamic Systems* and area editor of the *International Journal of Operations and Quantitative Management*. He is Chairman of the Technical Committee of Networks and Communications of IEEE Control Systems Society.

Yat-wah Wan received the B.Sc. degree in mechanical engineering from the Hong Kong University in 1981, the M.S. degree in industrial engineering from the Texas A&M University in 1986, College Station, and the Ph.D. degree from the University of California at Berkeley in 1991.

From 1991 to 1993, he worked in the Manufacturing Engineering Department, City Polytechnic of Hong Kong. Since 1993, he has been working in the Industrial Engineering and Engineering Management, Hong Kong University of Science and Technology. His research interests are applied stochastic processes, queueing theory, and scheduling.