INFORMS Journal on Computing
Vol. 8, No. 1, Winter 1996

0899-1499/96/0801-0045 $01.25

Computation of Derivatives by Means of the Power-Series Algorithm

J. P. C. BLANC / *Tilburg University, Center for Economic Research, Tilburg, The Netherlands; Email: blanc@kub.nl*

R. D. VAN DER MEI / *Tilburg University, Center for Economic Research, Tilburg, The Netherlands, Email: mei@kub.nl*

(Received: October 1993; revised: July 26; accepted: February 1995)

The power-series algorithm (PSA) is a flexible tool for computing performance measures for moderately-sized queueing systems for which the underlying process has a multidimensional quasi birth-and-death structure. In the present paper the PSA is extended to the computation of derivatives of system performance measures with respect to a general class of system parameters. This extension is useful for analyzing the sensitivity of the system performance with respect to the system parameters and for solving a wide variety of optimization problems in queueing systems.

The rapid development of computer-communication systems has generated a variety of challenging problems to predict and to control the performance of systems with multiple queues. However, in most cases multiple-queue models are too complicated to be handled by means of mathematical techniques. Moreover, the computation time needed to achieve satisfactorily accurate results with simulation is often unacceptable. Therefore, numerical algorithms have been developed to compute system performance measures. In this paper we consider the so-called power-series algorithm (PSA), a method to compute the steady-state distribution of the broad class of multi-queue systems which can be modeled as a multi-dimensional quasi birth-and-death (QBD) process. The basic idea of the PSA is the transformation of the non-recursively solvable (infinite) set of balance equations into an, in principle, recursively solvable set of equations by adding one dimension to the state space. This transformation is realized by expressing the state probabilities as power series in some variable in light traffic. For moderately-sized systems, the PSA favourably compares with simulation and numerical methods based on truncation of the state space, mainly because it involves recursive schemes. We refer to [4] for a recent overview of the state of art of the PSA.

The ultimate goal of system modeling and analysis is efficient operation and system optimization. Optimization procedures involving real-valued decision variables generally require (partial) derivatives of a function, the cost function, to be optimized. When these partial derivatives are estimated on the basis of finite differences (cf., e.g., section 6.7 of [12], section 4.6 of [9]), one is confronted with a number of practical difficulties. Firstly, it is a priori unclear which step size should be chosen to calculate these finite differences, while it is known that the choice of an appropriate step size, say h, may have a large impact on the efficiency of the optimization procedure. More precisely, when the step size h is too large higher order derivatives of the cost function may predominate in the estimation of the derivatives. On the other hand, if h is too small the numerator and the denominator of the gradient estimator may both be close to zero, making the estimated derivative highly sensitive to inaccuracies in the computed values of the cost function. When the cost function values are not known exactly and numerical algorithms have to be applied to compute these values, inaccuracies are unavoidable. Numerical experience with the PSA has indicated that an inappropriate choice of the step size may have a dramatical effect on the computation time of an optimization procedure and in many cases the optimum is not even found at all. These observations make optimization procedures, in which derivatives are estimated by finite differences, unreliable. Secondly, when derivatives are estimated by finite differences the performance of neighbouring schedules has to be evaluated, which may be a rather time consuming task. Thirdly, for parameter values nearby or at the boundary of the feasible region neighbouring values may be infeasible, so that modifications of the finite difference estimator would have to be made. In practice, the latter goes at the expense of the transparency of the computer program of the optimization algorithm, because many "special cases" have to be dealt with.

To be exempted from these practical complications, we extend the use of the PSA to the computation, instead of estimation, of derivatives. In practice, this extension makes the PSA much more easily applicable for optimization purposes. A wide variety of numerical optimization problems related to queueing systems can be solved by combining the extension of the PSA with some classical gradient method for non-linear optimization (cf., e.g., chapter 6 of [12] for an overview). On the negative side, the extension of the PSA to the computation of derivatives increases the required amount of storage capacity (as elaborated upon in Section 4).

In this paper we present a general approach for the computation of derivatives of performance measures with respect to a broad class of system parameters. For parameters which are controllable (e.g., routing probabilities for servers and/or customers) the extension allows for the use of gradient methods to optimize the expected performance. For parameters which do not serve as decision variables (e.g., arrival rates) the extension allows for analyzing the sensitivity of performance measures with respect to these system parameters.

The main limitations of the PSA are the required amounts of storage capacity and computation time, restricting the use of the algorithm to moderately-sized systems. For those systems, the PSA may be used to gain insight into their performance. These insights may be used to derive sharp approximations for performance measures and optimal values for control variables. The accuracy of the approximations can then be tested by means of the PSA. Because moderately-sized systems usually cover the main characteristics of larger systems, these approximations are also applicable for systems with a large number of queues.

The general approach to the computation of derivatives will be demonstrated for a polling model, i.e., a multi-queue model in which the queues are attended to by a single server. Polling models are widely applicable for the modeling in the areas of computer-communication systems, maintenance and manufacturing (cf. [11] for an overview). Because in several applications the server has no global information of the buffer contents, often a cyclic server routing is chosen. One of the major control mechanisms is the use of service limits, which restrict the number of customers served during a visit of the server to a queue. These service limits can be used to give relative priorities to the customers at the different queues while keeping some degree of fairness between the customers. We consider a cyclic polling model controlled by so-called Bernoulli service disciplines for determining stochastic service limits.

Under a Bernoulli service discipline the maximal number of customers served during a visit of the server to queue i is geometrically distributed with parameter q_i, the Bernoulli parameter belonging to queue i. The Bernoulli service discipline may be viewed as a stochastic counterpart of the widely used limited service discipline, under which the number of customers served during a visit of the server to a queue has a fixed upper bound. Because the Bernoulli parameters serve as control variables, the problem of finding optimal combinations of Bernoulli parameters arises quite naturally here. Polling systems controlled by Bernoulli schedules are very hard to handle analytically, let alone the problem of determining optimal combinations of the control parameters q_i. The PSA provides a means to analyze the behavior of these systems. For the Bernoulli polling model we demonstrate how the PSA can be applied to compute performance measures, and their derivatives with respect to the Bernoulli parameters, the latter opening the possibility of studying optimal combinations of Bernoulli parameters given some objective function (cf. [6] for results on optimization of Bernoulli polling systems). Emphasis will be on the practical aspects of the implementation. We will discuss the practical issues of the required amounts of computation time and storage capacity.

The remainder of the paper is organized as follows. In Section 1 a detailed description of the general QBD model is given. Section 2 gives an outline of the use of the PSA for the general QBD model and serves as the starting point for Section 3, which contains the main contribution of the paper. In that section the use of the PSA for QBD processes is extended to the computation of derivatives with respect to a general class of parameters. In Section 4 the complexity of the extension is discussed extensively and practical notes on the implementation are given. Finally, in Section 5 the general approach to the computation of derivatives will be demonstrated for a polling model with so-called Bernoulli schedules (in which the service disciplines serve as control variables), with emphasis on practical aspects of the implementation.

1. Model Description

Consider a multi-queue system consisting of s queues, $Q_1, \ldots, Q_s$. The queue-length process is described by an s-dimensional vector $\mathbf{N}(t) = (N_1(t), \ldots, N_s(t))$, which indicates the number of customers at each of the queues at time t, $t \geq 0$. In general, the process $\{\mathbf{N}(t), t \geq 0\}$ is not a Markov process. In order to transform the process $\{\mathbf{N}(t), t \geq 0\}$ into a Markov process, we add a vector of supplementary variables $\Phi(t)$. This variable may, e.g., be used to model phase-type distributions in the arrival or service processes, or to describe the status of the servers. For simplicity of the discussion, it is assumed that the supplementary space is the same for each $\mathbf{n} \in \mathbb{N}^s$, while it is possible that some states $(\mathbf{n}, \varphi)$ can not be entered. The supplementary space is assumed to be finite and is denoted by Θ. The joint process $\{(\mathbf{N}(t), \Phi(t)), t \geq 0\}$ is a Markov process (on the state space $\mathbb{N}^s \times \Theta$) with a QBD structure; that is, the time between an entrance at state $(\mathbf{n}, \varphi)$ and the successive departure from that state is exponentially distributed and transitions are only possible to states with at most one unit more or one unit less in one of the first s entries. The one-step transition rates are defined as follows: for $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $\psi \in \Theta$,

$\chi a^{(j)}(\mathbf{n}, \varphi, \psi)$: the arrival rate at Q_j at state $(\mathbf{n}, \varphi)$, leading to a transition to state $(\mathbf{n} + \mathbf{e}_j, \psi)$, $j = 1, \ldots, s$;

$d^{(j)}(\mathbf{n}, \varphi, \psi)$: the departure rate from Q_j at state $(\mathbf{n}, \varphi)$, leading to a transition to state $(\mathbf{n} - \mathbf{e}_j, \psi)$, with $d^{(j)}(\mathbf{n}, \varphi, \psi) = 0$ if $n_j = 0$, $j = 1, \ldots, s$;

$u(\mathbf{n}, \varphi, \psi)$: the phase-transition rate from state $(\mathbf{n}, \varphi)$ to $(\mathbf{n}, \psi)$;

here, $\mathbf{e}_j$ stands for the j-th unit vector in $\mathbb{N}^s$, $j = 1, \ldots, s$. For $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $\psi \in \Theta$, $j = 1, \ldots, s$, write

$$\lambda^{(j)}(\chi; \mathbf{n}, \varphi, \psi) := \chi a^{(j)}(\mathbf{n}, \varphi, \psi). \tag{1}$$

For a given set of relative arrival rates $a^{(j)}(\mathbf{n}, \varphi, \psi)$, $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $\psi \in \Theta$, $j = 1, \ldots, s$, the arrival rates $\lambda^{(j)}(\chi; \mathbf{n}, \varphi, \psi)$, $\chi \geq 0$, $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $\psi \in \Theta$, $j = 1, \ldots, s$, are functions of χ. In this way, the parametrization (1) actually characterizes a *family of systems*, each of which corresponds uniquely to one particular value of χ. The quantity χ will be used as a variable in the PSA.

Although necessary and sufficient conditions for the stability of the system are generally unknown, it is assumed here that the system is stable and that the system is in steady state. Denote by $(\mathbf{N}, \Phi)$ stochastic variables with as joint distribution the stationary distribution of the process $(\mathbf{N}(t), \Phi(t))$.

The transition rates $d^{(j)}(\mathbf{n}, \varphi, \psi)$, $u(\mathbf{n}, \varphi, \psi)$ and $a^{(j)}(\mathbf{n}, \varphi, \psi)$ are assumed to be functions of some vector of continuous control variables $\gamma = (\gamma_1, \ldots, \gamma_R)$, and it is assumed that

they are differentiable with respect to γ_r; the partial derivatives are denoted by $d_r^{(j)}(\mathbf{n}, \varphi, \psi)$, $u_r(\mathbf{n}, \varphi, \psi)$ and $a_r^{(j)}(\mathbf{n}, \varphi, \psi)$, respectively, $r = 1, \ldots, R$ (where some of these derivatives may vanish). The variable χ is assumed to be independent of γ. The components of γ may be, e.g., arrival rates, service rates, routing probabilities (for servers or customers) or parameters of a service discipline.

2. The Power-Series Algorithm for General Quasi-Birth-and-Death Processes

In this section we summarize how, under rather weak assumptions, the PSA can be applied to the general QBD model discussed in the previous section (cf. [3]). The state probabilities are defined and the global balance equations are formulated. The basic idea of the PSA is to transform this generally non-recursively solvable set of balance equations into a recursively solvable set of equations by expressing the state probabilities as power series and deriving a recursive computation scheme to determine the coefficients of these power series. Define the state probabilities as follows: for $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$,

$$p(\mathbf{n}, \varphi) = \Pr\{(\mathbf{N}, \Phi) = (\mathbf{n}, \varphi)\}. \tag{2}$$

The ergodicity assumption implies that for each state $(\mathbf{n}, \varphi)$ the total rate into that state is equal to the total rate out of that state. State transitions occur at either a customer arrival or a service completion of a customer or a phase transition. The rate-in-rate-out equations for the state probabilities (2) can be formulated as follows: for $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$,

$$\begin{aligned}
&\left(\sum_{j=1}^{s} \sum_{\psi \in \Theta} [\chi a^{(j)}(\mathbf{n}, \varphi, \psi) + d^{(j)}(\mathbf{n}, \varphi, \psi)] + \sum_{\psi \in \Theta} u(\mathbf{n}, \varphi, \psi) \right) p(\mathbf{n}, \varphi) \\
&= \chi \sum_{j=1}^{s} \sum_{\psi \in \Theta} a^{(j)}(\mathbf{n} - \mathbf{e}_j, \psi, \varphi)\, p(\mathbf{n} - \mathbf{e}_j, \psi) I\{n_j > 0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi \in \Theta} d^{(j)}(\mathbf{n} + \mathbf{e}_j, \psi, \varphi)\, p(\mathbf{n} + \mathbf{e}_j, \psi) \\
&\quad + \sum_{\psi \in \Theta} u(\mathbf{n}, \psi, \varphi)\, p(\mathbf{n}, \psi),
\end{aligned} \tag{3}$$

where $I\{E\}$ stands for the indicator function of the event E. Further, according to the law of total probability we have

$$\sum_{(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta} p(\mathbf{n}, \varphi) = 1. \tag{4}$$

In general, (3) and (4) form an (infinite) set of equations which can not be solved recursively. The PSA transforms this set of equations into a (mainly) recursively solvable set of equations by expressing the state probabilities as power series in the load offered to the system. The solution method for the set of equations (3), (4) relies on the following property: for $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$,

$$p(\mathbf{n}, \varphi) = O(\chi^{|\mathbf{n}|}), \quad \chi \downarrow 0, \tag{5}$$

where $|\mathbf{n}| := n_1 + \ldots + n_s$. This property (5) can be shown to be valid if for each reachable $\mathbf{n} \neq \mathbf{0}$, there is at least one positive departure rate, and for each reachable state $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $\mathbf{n} \neq \mathbf{0}$, the probability that a departure occurs before any arrival takes place, after the process has entered this state, is positive (cf. [14] for more details). The above condition is fulfilled in many practical cases, but it is *not*, for instance, if service only starts when the number of customers in a queue has reached some *threshold* larger than 1. Based on property (5), we introduce the following formal power-series expansions for the state probabilities: for $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $0 \leq |\chi| < \chi_0$,

$$p(\mathbf{n}, \varphi) = \chi^{|\mathbf{n}|} \sum_{k=0}^{\infty} \chi^k b_0(k; \mathbf{n}, \varphi), \tag{6}$$

for some positive real-valued radius of convergence χ_0. We refer to [13] for conditions upon which there exists a positive χ_0 such that the power-series expansions converge for all $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$. Substituting the power-series expansions (6) into the global balance equations (3) and equating the coefficients of corresponding powers of χ yields: for $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$,

$$\begin{aligned}
&\sum_{\psi \in \Theta} \left\{ \sum_{j=1}^{s} d^{(j)}(\mathbf{n}, \varphi, \psi) + u(\mathbf{n}, \varphi, \psi) \right\} b_0(k; \mathbf{n}, \varphi) \\
&= \sum_{j=1}^{s} \sum_{\psi \in \Theta} a^{(j)}(\mathbf{n} - \mathbf{e}_j, \psi, \varphi) b_0(k; \mathbf{n} - \mathbf{e}_j, \psi) I\{n_j > 0\} \\
&\quad - \sum_{j=1}^{s} \sum_{\psi \in \Theta} a^{(j)}(\mathbf{n}, \varphi, \psi) b_0(k - 1; \mathbf{n}, \varphi) I\{k > 0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi \in \Theta} d^{(j)}(\mathbf{n} + \mathbf{e}_j, \psi, \varphi) b_0(k - 1; \mathbf{n} + \mathbf{e}_j, \psi) I\{k > 0\} \\
&\quad + \sum_{\psi \in \Theta} u(\mathbf{n}, \psi, \varphi) b_0(k; \mathbf{n}, \psi).
\end{aligned} \tag{7}$$

This set of equations (7) forms a recursive scheme with respect to the components $(k; \mathbf{n})$ under the following partial ordering $\prec$ of the vectors $(k; \mathbf{n}, \varphi), (\hat{k}; \hat{\mathbf{n}}, \hat{\varphi}) \in \mathbb{N}^{1+s} \times \Theta$:

$$(k; \mathbf{n}, \varphi) \prec (\hat{k}; \hat{\mathbf{n}}, \hat{\varphi}) \tag{8}$$

$$\text{if} \quad [k + |\mathbf{n}| < \hat{k} + |\hat{\mathbf{n}}|] \vee [k + |\mathbf{n}| = \hat{k} + |\hat{\mathbf{n}}| \wedge k < \hat{k}].$$

It is readily verified that (7) expresses the coefficients $b_0(k; \mathbf{n}, \varphi)$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$, in terms of coefficients of lower order than $(k; \mathbf{n}, \varphi)$ with respect to $\prec$, except for the coefficients $b_0(k; \mathbf{n}, \varphi)$, $\psi \in \Theta$. Hence, the coefficients can be calculated (mainly) recursively in increasing order with respect to $\prec$, where for each $(k; \mathbf{n})$ a set of at most $|\Theta|$ linear equations may have to be solved. The same conditions which guarantee that (5) holds also guarantee that these sets of equations possess a unique solution. The only exceptions are formed by empty states, i.e., states with $\mathbf{n} = \mathbf{0}$. In these

cases all departure rates vanish so that the sets of equations (7) reduce to: for $\varphi \in \Theta$, $k = 0, 1, \ldots,$

$$\begin{aligned}
&\sum_{\psi\in\Theta} u(\mathbf{0}, \varphi, \psi) b_0(k; \mathbf{0}, \varphi) \\
&= -\sum_{j=1}^{s} \sum_{\psi\in\Theta} a^{(j)}(\mathbf{0}, \varphi, \psi) b_0(k-1; \mathbf{0}, \varphi) I\{k>0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi\in\Theta} d^{(j)}(\mathbf{e}_j, \psi, \varphi) b_0(k-1; \mathbf{e}_j, \psi) I\{k>0\} \\
&\quad + \sum_{\psi\in\Theta} u(\mathbf{0}, \psi, \varphi) b_0(k; \mathbf{0}, \psi).
\end{aligned} \tag{9}$$

One may verify by summing the equations (9) over φ, $\varphi \in \Theta$, that these are dependent sets of equations for the coefficients $b_0(k; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, for each k, $k = 0, 1, \ldots$. However, an additional equation between the coefficients $b_0(k; \mathbf{n}, \varphi)$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$, follows from the law of total probability (4):

$$\sum_{\varphi\in\Theta} b_0(0; \mathbf{0}, \varphi) = 1; \tag{10}$$

$$\sum_{\varphi\in\Theta} b_0(k; \mathbf{0}, \varphi) = -\sum_{0<|\mathbf{n}|\le k} \sum_{\varphi\in\Theta} b_0(k-|\mathbf{n}|; \mathbf{n}, \psi), \tag{11}$$

$$k = 1, 2, \ldots.$$

Note that the right-hand side of (11) consists of terms of lower order with respect to $<$ than $b_0(k; \mathbf{0}, \varphi)$. In the sequel it is assumed that all but one of the equations (9) together with either (10) or (11) determine $b_0(k; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, $k = 0, 1, \ldots$. One may verify that the determinants of these sets of equations are the same for all k, $k = 0, 1, \ldots$, so that the sets of equations are solvable for all k, $k = 0, 1, \ldots$, if and only if it is solvable for $k = 0$. A sufficient condition for the solvability of the set of equations is that the Markov chain with transition probabilities $u(\mathbf{0}, \varphi, \psi)$, $\varphi, \psi \in \Theta$ is irreducible. In other words, the set of equations is uniquely solvable if the process, conditioned on the event that $\mathbf{N} = \mathbf{0}$ and no arrivals occur at all is irreducible on the subset of Θ of reachable states. This conditioned process will be referred to as the $\mathbf{0}$-process. When the $\mathbf{0}$-process has more than one recurrent class, the order in which the coefficients of the power-series expansions are computed has to be modified.

3. Extension to Derivatives

In this section a complete computation scheme is derived to calculate the derivatives of performance measures with respect to control variables γ_r, $r = 1, \ldots, R$.

Define the derivatives of the state probabilities: for $r = 1, \ldots, R$, $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$,

$$p_r(\mathbf{n}, \varphi) := \frac{\partial}{\partial \gamma_r} p(\mathbf{n}, \varphi). \tag{12}$$

These derivatives are expressed as power series in χ. For now, we assume that the variable χ satisfies the following two restrictions: (i) the system is stable for $0 \le \chi < 1$ and (ii) the variable χ does *not* depend on the value of the control parameter γ. At the end of this section we will show that the variable χ can indeed be chosen in such a way that these two properties are satisfied. Because χ is assumed to be independent of the control parameter γ, the power-series expansions of the derivatives of the state probabilities with respect to the components of γ can be obtained by termwise differentiation of the power-series expansions of the state probabilities (6): for $r = 1, \ldots, R$, $(\mathbf{n}, \varphi) \in \mathbb{N}^s \times \Theta$, $0 \le |\chi| < \chi_0$,

$$p_r(\mathbf{n}, \varphi) = \chi^{|\mathbf{n}|} \sum_{k=0}^{\infty} \chi^k b_r(k; \mathbf{n}, \varphi), \tag{13}$$

with

$$b_r(k; \mathbf{n}, \varphi) := \frac{\partial}{\partial \gamma_r} b_0(k; \mathbf{n}, \varphi). \tag{14}$$

Differentiating both sides of the equations (7) with respect to γ_r, $r = 1, \ldots, R$, yields the following set of equations for the coefficients of the power series (cf. (6)): for $r = 1, \ldots, R$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$,

$$\begin{aligned}
&\sum_{\psi\in\Theta} \left(\sum_{j=1}^{s} d^{(j)}(\mathbf{n}, \varphi, \psi) + u(\mathbf{n}, \varphi, \psi) \right) b_r(k; \mathbf{n}, \varphi) \\
&= \sum_{\psi\in\Theta} [u(\mathbf{n}, \psi, \varphi) b_r(k; \mathbf{n}, \psi) + u_r(\mathbf{n}, \psi, \varphi) b_0(k; \mathbf{n}, \psi)] \\
&\quad + \sum_{j=1}^{s} \sum_{\psi\in\Theta} a^{(j)}(\mathbf{n} - \mathbf{e}_j, \psi, \varphi) b_r(k; \mathbf{n} - \mathbf{e}_j, \psi) I\{n_j > 0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi\in\Theta} a_r^{(j)}(\mathbf{n} - \mathbf{e}_j, \psi, \varphi) b_0(k; \mathbf{n} - \mathbf{e}_j, \psi) I\{n_j > 0\} \\
&\quad - \sum_{j=1}^{s} \sum_{\psi\in\Theta} a^{(j)}(\mathbf{n}, \varphi, \psi) b_r(k-1; \mathbf{n}, \varphi) I\{k>0\} \\
&\quad - \sum_{j=1}^{s} \sum_{\psi\in\Theta} a_r^{(j)}(\mathbf{n}, \varphi, \psi) b_0(k-1; \mathbf{n}, \varphi) I\{k>0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi\in\Theta} d^{(j)}(\mathbf{n} + \mathbf{e}_j, \psi, \varphi) b_r(k-1; \mathbf{n} + \mathbf{e}_j, \psi) I\{k>0\} \\
&\quad + \sum_{j=1}^{s} \sum_{\psi\in\Theta} d_r^{(j)}(\mathbf{n} + \mathbf{e}_j, \psi, \varphi) b_0(k-1; \mathbf{n} + \mathbf{e}_j, \psi) I\{k>0\} \\
&\quad - \sum_{\psi\in\Theta} \left(\sum_{j=1}^{s} d_r^{(j)}(\mathbf{n}, \varphi, \psi) + u_r(\mathbf{n}, \varphi, \psi) \right) b_0(k; \mathbf{n}, \varphi).
\end{aligned} \tag{15}$$

To derive a computation scheme for the coefficients $b_r(k; \mathbf{n}, \varphi)$, we extend the partial ordering $<$ of the triples $(k; \mathbf{n}, \varphi)$ in (8) to the partial ordering $\prec$ of the quadruples

$(r, k; \mathbf{n}, \varphi)$ in the following way: for $(r, k; \mathbf{n}, \varphi), (\hat{r}, \hat{k}; \hat{\mathbf{n}}, \hat{\varphi}) \in \{0, 1, \ldots, R\} \times \mathbb{N}^{1+s} \times \Theta$,

$$(r, k; \mathbf{n}, \varphi) \tilde{\prec} (\hat{r}, \hat{k}; \hat{\mathbf{n}}, \hat{\varphi}) \quad \text{if} \quad [r = 0 \wedge \hat{r} > 0] \vee [r = \hat{r} \wedge (k, \mathbf{n}, \varphi) \prec (\hat{k}, \hat{\mathbf{n}}, \hat{\varphi})]. \tag{16}$$

The set of equations (15) expresses coefficients $b_r(k; \mathbf{n}, \varphi)$, $r = 0, 1, \ldots, R$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$, in terms of coefficients of lower order with respect to $\tilde{\prec}$, except for the terms $b_r(k; \mathbf{n}, \varphi)$, $\psi \in \Theta$. Hence, the coefficients $b_r(k; \mathbf{n}, \varphi)$ may be computed recursively in increasing order with respect to $\tilde{\prec}$, where for each $(r, k; \mathbf{n})$ a set of at most $|\Theta|$ linear equations, with unknowns $b_r(k; \mathbf{n}, \varphi)$, $\varphi \in \Theta$, may have to be solved. The only exceptions are formed by the states with $\mathbf{n} = \mathbf{0}$. In these cases the departure rates vanish, so that the equations (15) reduce to: for $r = 1, \ldots, R$, $k = 0, 1, \ldots$, $\varphi \in \Theta$,

$$\sum_{\psi \in \Theta} u(\mathbf{0}, \varphi, \psi) b_r(k; \mathbf{0}, \varphi) = \sum_{\psi \in \Theta} u(\mathbf{0}, \psi, \varphi) b_r(k; \mathbf{0}, \psi) + y_r(k; \varphi), \tag{17}$$

where the quantities $y_r(k; \varphi)$, $r = 1, \ldots, R$, $k = 0, 1, \ldots$, $\varphi \in \Theta$, are defined by $y_r(0; \varphi) := 0$, and for $k = 1, 2, \ldots$, by

$$\begin{aligned} y_r(k; \varphi) := & - \sum_{j=1}^{s} \sum_{\psi \in \Theta} \big(a^{(j)}(\mathbf{0}, \varphi, \psi) b_r(k-1; \mathbf{0}, \varphi) \\ & \qquad + a_r^{(j)}(\mathbf{0}, \varphi, \psi) b_0(k-1; \mathbf{0}, \varphi) \big) \\ & + \sum_{j=1}^{s} \sum_{\psi \in \Theta} \big(d^{(j)}(\mathbf{e}_j, \psi, \varphi) b_r(k-1; \mathbf{e}_j, \psi) \\ & \qquad + d_r^{(j)}(\mathbf{e}_j, \psi, \varphi) b_0(k-1; \mathbf{e}_j, \psi) \big) \\ & + \sum_{\psi \in \Theta} \big(u_r(\mathbf{0}, \psi, \varphi) b_0(k; \mathbf{0}, \psi) \\ & \qquad - u_r(\mathbf{0}, \varphi, \psi) b_0(k; \mathbf{0}, \varphi) \big). \end{aligned} \tag{18}$$

All coefficients at the right-hand side of (18) are of lower order with respect to $\tilde{\prec}$ than $b_r(k; \mathbf{0}, \varphi)$ and, hence, can be considered to be known in (17). Because of a necessary balance in transitions between the set of empty states and the set of states with one customer in the system, one may verify by summing over $\varphi \in \Theta$, that the set of equations (17) is a *dependent* set of equations for the coefficients $b_r(k; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, for $r = 1, \ldots, R$ and for $k = 0, 1, \ldots$. An additional equation is implied by the law of total probability (4): for $r = 1, \ldots, R$, it follows that for $k = 0$,

$$\sum_{\varphi \in \Theta} b_r(0; \mathbf{0}, \varphi) = 0, \tag{19}$$

and that for $k = 1, 2, \ldots$,

$$\sum_{\varphi \in \Theta} b_r(k; \mathbf{0}, \varphi) = - \sum_{0 < |\mathbf{n}| \le k} \sum_{\psi \in \Theta} b_r(k - |\mathbf{n}|; \mathbf{n}, \psi). \tag{20}$$

The right-hand side of (20) contains only coefficients of states of a lower order with respect to $\tilde{\prec}$ than $b_r(k; \mathbf{0}, \varphi)$. The determinants of the set of all but one of the equations (17) taken together with either (19) or (20) do not depend on k, $k = 0, 1, \ldots$, nor on r, $r \in \{0, 1, \ldots, R\}$ (for the case $r = 0$ see (9), (10), (11)), so that it suffices to consider the solvability of the set of equations for $k = 0$ and $r = 0$. Hence, the set of equations (17) for the coefficients $b_r(k; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, is uniquely solvable if and only if the set of equations (9), (10) between the coefficients $b_0(0; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, is uniquely solvable. The solvability of the latter set of equations has been discussed in Section 2.

The PSA allows for the computation of all state probabilities and hence, of any real-valued function of the state-probabilities, as well as their derivatives with respect to the general control parameter γ. In practice, only a limited number of performance measures, say L, may have to be computed (e.g., mean queue lengths), rather than all individual state probabilities. Let $g^{(l)}(\mathbf{n}, \varphi)$ be an arbitrary real-valued function of the state space. Most performance measures of the system are of the form $E\{g^{(l)}(\mathbf{N}, \Phi)\}$, $l = 1, \ldots, L$. It is assumed throughout that the performance measures $E\{g^{(l)}(\mathbf{N}, \Phi)\}$, $l = 1, \ldots, L$, are differentiable with respect to all components of γ. These performance measures, and their derivatives with respect to γ_r, $r = 1, \ldots, R$, can be expressed in terms of the coefficients $b_r(k; \mathbf{n}, \varphi)$ as: for $l = 1, \ldots, L$,

$$\begin{aligned} E\{g^{(l)}(\mathbf{N}, \Phi)\} &= \sum_{k=0}^{\infty} \chi^k f_0^{(l)}(k), \\ \frac{\partial}{\partial \gamma_r} E\{g^{(l)}(\mathbf{N}, \Phi)\} &= \sum_{k=0}^{\infty} \chi^k f_r^{(l)}(k), \quad r = 1, \ldots, R, \end{aligned} \tag{21}$$

where for $r = 0, 1, \ldots, R$, $k = 0, 1, \ldots$,

$$f_r^{(l)}(k) := \sum_{0 \le |\mathbf{n}| \le k} \sum_{\psi \in \Theta} g^{(l)}(\mathbf{n}, \varphi) b_r(k - |\mathbf{n}|; \mathbf{n}, \varphi). \tag{22}$$

In practice, only a finite number of coefficients can be computed (caused by limitations on the available amounts of computation time and storage capacity). Let M be the number of terms that one wants or has to compute. Then the coefficients $b_r(k; \mathbf{n}, \varphi)$, $r = 0, 1, \ldots, R$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$, of the power series in (6) and (13), and the coefficients $f_r^{(l)}(k)$, $r = 0, 1, \ldots, R$, $l = 1, \ldots, L$, $k = 0, 1, \ldots$, of the power-series expansions in (21) and (22) can be computed according to the following computation scheme (up to the power M of χ):

Step 1: determine $b_0(0; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, by solving all but one of the equations (9) together with (10), and for $r = 1, \ldots, R$, determine $b_r(0; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, by solving all but one of the equations (17) together with (19); determine $f_r^{(l)}(0)$, $r = 0, 1, \ldots, R$, $l = 1, \ldots, L$, according to (22);

Step 2: $m := 1$;

Step 3: for all $r = 0, 1, \ldots, R$, $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$ with $\mathbf{n} \neq \mathbf{0}$ and with $k + |\mathbf{n}| = m$, determine $b_r(k; \mathbf{n}, \varphi)$, according to (7) for $r = 0$ and (15) for $r > 0$ (in increasing order of $(r, k; \mathbf{n}, \varphi)$ with respect to $\tilde{\prec}$) and update the value to $f_r^{(l)}(m)$, $l = 1, \ldots, L$;

Step 4: for $r = 0, 1, \ldots, R$, determine $b_r(m; \mathbf{0}, \varphi)$, $\varphi \in \Theta$, by solving the set of equations consisting of all but one of the equations (9) for $r = 0$ and (17) for $r > 0$,

together with (11) for $r = 0$ and (20) for $r > 0$, and update the value to $f_r^{(j)}(m)$, $l = 1, \ldots, L$;

Step 5: if $m < M$ then $m := m + 1$ and return to *Step 3*; otherwise STOP.

From the global balance equations (3) it follows directly that the state-probabilities do not depend on the normalization of the arrival rates. Moreover, by construction, the computed values (partial sums) of the state probabilities are also independent of the normalization of the arrival rates (cf. (1)).

The power series (6), (13), and (21) are usually not convergent for all values of χ for which the system is stable. To overcome this difficulty, there are techniques available to improve the convergence of the power series. The so-called conformal mapping technique may be applied to increase the radius of convergence of the power series (cf. [1]). In addition, the so-called ϵ-algorithm is very useful for increasing the speed of convergence and to enlarge the radius of convergence of the power series (cf. [15], [2]). For both techniques it is convenient that the arrival rates are normalized in such a way that the system is stable for $0 \leq \chi < 1$ and that (in the case of infinite buffer systems) the smallest value of χ for which the system becomes instable is equal to 1. However, for given relative arrival rates, the threshold value of χ for which the (infinite-buffer) system becomes instable generally depends on the value of the control parameter γ. We will now show that the arrival rates can be normalized in such a way that the variable χ is independent of the control parameter γ and the system becomes instable for $\chi \uparrow 1$. To show that there exists a variable χ with the desired properties, let $\hat{\chi}$ be a power-series variable which does not depend on the value of the control parameter γ. Then the smallest value of $\hat{\chi}$ for which the system explodes may depend on γ, say at $\hat{\chi} = f(\gamma)$, for some strictly positive real-valued function f. Because we consider the performance of the system for given values of γ, we fix the value of γ, say $\gamma = \gamma^{(0)}$. Then *for given* $\gamma = \gamma^{(0)}$, we define

$$\chi := \frac{\hat{\chi}}{f(\gamma^{(0)})}. \tag{23}$$

Then it is easily seen that the system becomes instable for $\chi \uparrow 1$. Moreover, from definition (23), χ is *not* a function of the variable γ, and differs from $\hat{\chi}$ only by a *constant* scale factor $f(\gamma^{(0)})$, which is known to have no influence on the computed performance measures and their derivatives. Note that the relative arrival rates $a^{(j)}(\mathbf{n}, \varphi, \psi)$ are normalized in such a way that the system explodes for $\chi \uparrow 1$ *only for* $\gamma = \gamma^{(0)}$. For neighbouring values of γ, $\gamma \neq \gamma^{(0)}$, the system does not necessarily become instable for $\chi \uparrow 1$. These considerations motivate why χ, defined in (23), can be considered as a variable which is independent of the control parameter γ. In this way, one avoids complications which would occur if χ were chosen to be a (possibly non-differentiable) function of the variable γ.

4. Complexity

The main restriction in applying the PSA is the required amount of memory space. The total number of coefficients that have to be computed to determine the coefficients of the power series of the state probabilities (6) up to the M-th power of χ is given by (cf. also [3])

$$\binom{M+s+1}{s+1} \times |\Theta|, \tag{24}$$

where the first factor stands for the number of couples $(k; \mathbf{n})$ for which $k + |\mathbf{n}| \leq M$. Thus, the memory requirements increase *exponentially* in the number of queues, s, and in the number of terms of the power-series, M, restricting the use of the PSA to small and moderately-sized systems.

The total number of coefficients that have to be calculated to compute the power series of the state probabilities *and* their derivatives with respect to γ_r, $r = 1, \ldots, R$, up to the M-th power of χ is given by

$$(R+1) \times \binom{M+s+1}{s+1} \times |\Theta|. \tag{25}$$

Thus, the memory (and time) requirements increase *linearly* in the number of derivatives, R, that have to be computed.

In practice, the performance of the PSA can be strongly improved by efficient memory management and by techniques for improving the convergence of the power series. A strong reduction of the storage requirement can be achieved when only a limited number of performance measures (e.g., mean queue lengths) has to be evaluated, rather than all individual state probabilities. Then, the coefficients of the power-series expansions of the important performance measures can be *aggregated* during the execution of the PSA, and stored in small separate arrays (cf. (21), (22)), while the coefficients of the state probabilities can be *removed* as soon as they are not needed anymore in further computations. This storage procedure strongly reduces the storage requirement in (24) for the calculation of the power-series expansions and hence, increases of the maximum number of terms M that can be computed for given amount of memory space (cf. Table 2 of [2] for an illustration). When coefficients are removed as soon as they are not needed anymore, the required amount of memory space given in (25) is reduced to

$$(R+1) \times \binom{M+s}{s} \times |\Theta| \tag{26}$$

coefficients. As an illustration, we have computed the maximal number of terms of the power series that can be computed (according to (26)) for given amount of storage capacity of 10^7 coefficients and for various values of the number of queues, s, the size of the supplementary space, $|\Theta|$, and the number of derivatives, R. Table I shows that the number of terms of the power series that can be computed for a given amount of storage capacity may decrease considerably when the number of derivatives is increased. It should be noted that in the case $R = 0$ no derivatives are computed.

It is not easy to give general rules of thumb for the number of terms that have to be computed to achieve some desired degree of accuracy. This number generally depends on a number of factors such as the load offered to the system and on the "degree of symmetry" of the system. If the system is "fairly symmetrical" then 10 terms may suffice to

Table I. Maximum Number of Terms at a Storage Capacity of 10^7 Coefficients

	$s = 2$				$s = 4$				$s = 6$			
$\|\Theta\|$	4	12	24	48	4	12	24	48	4	12	24	48
$R = 0$	2234	1289	911	643	85	64	53	44	31	25	22	19
$R = 1$	1579	911	643	454	71	53	44	37	27	22	19	17
$R = 2$	1289	743	525	371	64	48	40	33	25	20	18	15
$R = 5$	911	525	371	262	53	40	33	27	22	18	15	13
$R = 10$	672	387	273	193	45	34	28	23	19	16	13	12

give rather accurate results for lightly loaded systems; if the system is heavily loaded then 10 to 15 terms may still do well (by applying extrapolation techniques, cf. [3]). If the system is rather asymmetrical the algorithm may converge rather slowly. If the system is lightly loaded then 10 to 15 terms may still do well, but if the system is more heavily loaded then typically 30 or 40 terms (or even more) may be needed to achieve accurate results.

When the system under consideration is a polling system (see Section 5 for an extensive discussion of a specific polling system), so-called pseudo-conservation laws (PCLs) (cf. [7]) may give an indication of the accuracy of the computations. A PCL is an exact expression for a specific weighted sum of the mean waiting times at the queues.

We will now discuss some ideas about the implementation of the PSA which are specific for the extension to the computation of derivatives as elaborated upon in Section 3. Firstly, it follows from the equations (15) that for given $(k; \mathbf{n}, \varphi)$ the coefficients $b_r(k; \mathbf{n}, \varphi)$, $r = 1, \ldots, R$, cannot be computed when $b_0(k; \mathbf{n}, \varphi)$ is unknown, but that the partial derivatives can be computed *separately*. It is most appropriate to compute for each $(k; \mathbf{n}, \varphi) \in \mathbb{N}^{1+s} \times \Theta$ the coefficients $b_r(k; \mathbf{n}, \varphi)$ $r \in \{1, \ldots, R\}$, *directly after* the computation of the coefficients $b_0(k; \mathbf{n}, \varphi)$. In this way the coefficients $b_0(k; \mathbf{n}, \varphi)$ need not be kept in memory.

Secondly, the fact that the coefficients $b_r(k; \mathbf{n}, \varphi)$, $r = 1, \ldots, R$, can be computed separately also implies that one may *partition* the set $T := \{1, \ldots, R\}$ into proper subsets, say $T_1, \ldots, T_m$ $(1 < m \leq R)$, and compute for each $(k; \mathbf{n}, \varphi)$ the coefficients $b_r(k; \mathbf{n}, \varphi)$ for $r \in \{0\} \cup T_j$ *successively*, $j = 1, \ldots, m$. Note that this partitioning may (partly) *compensate* for the loss of the maximal number of terms of the power series that can be computed (cf. Table I). As an illustration of the partitioning, consider a model with $s = 6$, $|\Theta| = 12$ and $R = 5$ (so that derivatives with respect to 5 different systems parameters have to be computed) and suppose the available amount of memory space is 10^7 coefficients. Then, according to Table I the maximal number of coefficients of the power series that can be computed is equal to 18. Suppose, however, that numerical experience with the PSA has taught that in order to achieve an "acceptable degree of accuracy" one needs to compute 20 terms of the power series. Then from Table I it follows that in order to have enough memory space to compute 20 terms of the power series the set $T := \{1, \ldots, 5\}$ should be partitioned into sets of at most 2 elements. For example, T could be partitioned into $T_1 = \{1, 2\}$, $T_2 = \{3, 4\}$, and $T_3 = \{5\}$.

It should be noted that this partitioning leads to an increase of the required amount of computation time, because in this case the terms $b_0(k; \mathbf{n}, \varphi)$ have to be computed m times instead of once. The latter illustrates that in order to determine the gradient for given amount of memory space one has to deal with a *trade-off* between the required amount of computation time on the one hand and the accuracy of the computed performance measures on the other hand.

The PSA is applicable to systems with a quasi birth-and-death structure. Therefore, general arrival processes such as the Markovian Arrival Processes (MAPs) can be handled. Service times and switch-over times may be of phase-type.

The PSA is most efficient when the coefficients can be determined recursively for each $(k; \mathbf{n})$-combination. For this reason, in the modeling of probability distributions, e.g., for service times, interarrival times or switch-over times, Coxian distribution are generally preferred to more general phase-type distributions. However, in some cases the number of phases in a general phase-type distribution needed to approximate some probability distribution is considerably smaller than in the case of Coxian distributions. In those cases, phase-type distributions may be preferred to Coxian distributions.

Higher order derivatives can also be determined by means of the PSA, following similar lines as discussed above. Clearly, this further extension adds to the complexity of the PSA, limiting the number of terms in power-series expansions that can be computed even more.

5. Application to a Polling Model

To demonstrate the general approach presented in Section 3, consider a polling system consisting of s infinite-buffer queues, $Q_1, \ldots, Q_s$, with Poisson arrival rates $\lambda_i = a_i\chi$, $i = 1, \ldots, s$. The service times at Q_i are exponentially distributed with rate μ_i^1, $i = 1, \ldots, s$. The server is routed along the queues in the cyclic order $1, 2, \ldots, s, 1, 2 \ldots$. The switch-over times needed by the server to move from Q_{i-1} to Q_i are assumed to be exponentially distributed with rate μ_i^0, $i = 1, \ldots, s$. The number of customers during a visit of the server to Q_i is determined as follows. When the server finds Q_i empty upon arrival, the queue is skipped and the server starts to move to the next queue. If after a service at Q_i there are still customers present at that queue, with probability q_i another customer at that queue is served; otherwise, the server proceeds to the next queue, $i = 1, \ldots, s$. Note that the cases $q_i = 0$ and $q_i = 1$ correspond to the classical 1-limited

and the exhaustive service discipline, respectively. The vector of parameters $\mathbf{q} = (q_1, \ldots, q_s)$ is referred to as a Bernoulli schedule. The arrival process, the service process and the switch-over process are assumed to be mutually independent and independent of the state of the system. Necessary and sufficient conditions for the stability of the systems read (cf. [8]) $\rho + \sigma_1\lambda_i(1 - q_i) < 1$, $i = 1, \ldots, s$, where $\sigma_1 := \sum_{i=1}^{s} 1/\mu_i^0$, the mean switch-over time per cycle, and $\rho := \sum_{i=1}^{s} \lambda_i/\mu_i^1$, the offered load to the system. In the sequel it is assumed that these conditions are satisfied and that the system is in steady state.

Let $\mathbf{N} = (N_1, \ldots, N_s)$ be the joint queue length vector and let (H, Z) be a couple of the supplementary variables, where h denotes that the server is either switching to or serving at Q_h and where Z indicates whether the server is switching ($Z = 0$) or serving ($Z = 1$). The supplementary space is given by $\Theta = \{1, \ldots, s\} \times \{0, 1\}$, while the states $(\mathbf{n}, h, 1)$ with $n_h = 0$ can not be entered. It is readily verified that the global balance equations read as follows. For the states in which the server is switching: for $\mathbf{n} \in \mathbb{N}^s$, $h = 1, \ldots, s$,

$$\left[\chi \sum_{i=1}^{s} a_i + \mu_h^0\right] p(\mathbf{n}, h, 0)$$

$$= \chi \sum_{i=1}^{s} a_i p(\mathbf{n} - \mathbf{e}_i, h, 0) I\{n_i > 0\} \tag{27}$$

$$+ \mu_{h-1}^{0,1} p(\mathbf{n}, h-1, 0) I\{n_{h-1} = 0\}$$

$$+ \mu_{h-1}^{1} p(\mathbf{n} + \mathbf{e}_{h-1}, h-1, 1)[1 - q_{h-1} I\{n_{h-1} > 0\}],$$

where the indices "$h - 1$" should be read as "s" if $h = 1$. The first term at the right-hand side of (27) indicates an arrival of a customer while the server is switching from Q_{h-1} to Q_h. The second term indicates that the server skips Q_{h-1} if this queue is empty and proceeds to Q_h immediately, and the third term indicates a service completion at Q_{h-1} followed by a departure of the server from that queue.

For the states in which the server is serving: for $\mathbf{n} \in \mathbb{N}^s$, $h = 1, \ldots, s$, $n_h > 0$,

$$\left[\chi \sum_{i=1}^{s} a_i + \mu_h^1\right] p(\mathbf{n}, h, 1)$$

$$= \chi \sum_{i=1}^{s} a_i p(\mathbf{n} - \mathbf{e}_i, h, 1) I\{n_i > 0\} \tag{28}$$

$$+ \mu_h^0 p(\mathbf{n}, h, 0) + q_h \mu_h^1 p(\mathbf{n} + \mathbf{e}_h, h, 1).$$

The first term at the right-hand side of (28) indicates a customer arrival during the service of a customer at Q_h. The second term indicates an arrival of the server at Q_h, followed by an immediate service initiation at that queue. The third term corresponds to a service completion at Q_h and a subsequent service initiation of another customer at that queue.

Table II. Maximal Number of Terms for Given Amount of Storage Capacity

	Memory = 10^6					Memory = 10^7						
s	2	3	4	5	6	2	3	4	5	6	7	8
$R = 0$	705	98	39	23	16	2234	213	71	38	25	19	15
$R = 1$	498	77	32	19	14	1579	168	59	32	22	17	13
$R = 2$	406	67	29	17	13	1289	147	53	30	20	15	13
$R = s$	406	61	25	15	10	1289	133	47	25	17	13	10

In addition, according to the law of the total probability,

$$\sum_{(\mathbf{n},h,\zeta) \in \mathbb{N}^s \times \Theta} p(\mathbf{n}, h, \zeta) = 1. \tag{29}$$

It is readily verified that the conditions for application of the PSA are satisfied (cf. Section 2). Based on the property $p(\mathbf{n}, h, \zeta) = O(\chi^{|\mathbf{n}|})$, $\chi \downarrow 0$, cf. (5), the state probabilities and their derivatives with respect to the Bernoulli parameters q_r, $r = 1, \ldots, s$, can be expressed as power series in χ as follows: for $(\mathbf{n}, h, \zeta) \in \mathbb{N}^s \times \Theta$,

$$p(\mathbf{n}, h, \zeta) = \chi^{|\mathbf{n}|} \sum_{k=0}^{\infty} \chi^k b_0(k; \mathbf{n}, h, \zeta), \tag{30}$$

and for $r = 1, \ldots, s$,

$$p_r(\mathbf{n}, h, \zeta) = \chi^{|\mathbf{n}|} \sum_{k=0}^{\infty} \chi^k b_r(k; \mathbf{n}, h, \zeta),$$

$$b_r(k; \mathbf{n}, h, \zeta) = \frac{\partial}{\partial q_r} b_0(k; \mathbf{n}, h, \zeta), \tag{31}$$

where the latter equality follows from the assumption that χ does not depend on the control parameter $\gamma = \mathbf{q}$. We refer to [5] for a complete derivation of a fully recursive computation scheme to determine the coefficients $b_r(k; \mathbf{n}, h, \zeta)$ (for the case of Coxian distributed service times and switch-over times).

For the present model the size of the supplementary space is $|\Theta| = 2s$. Table II shows the maximal number of terms of the power series that can be computed for varying number of queues, s, number of derivatives, R, and available amount of storage capacity. To illustrate that the maximal number of terms, M, may have a considerable impact on the accuracy of the computations, consider the model with the following set of system parameters: $s = 3$; $(a_1, a_2, a_3) = (1/6, 1/3, 1/2)$; $1/\mu_h^0 = 1.0$, $1/\mu_h^1 = 0.05$, $h = 1, 2, 3$. The performance measure is defined by

$$C(\mathbf{q}) := \sum_{i=1}^{s} c_i EW_i, \tag{32}$$

a weighted sum of the mean waiting times, where the weights corresponding to the different queues are set differently according to the relative importance of the queues. In the present example the weights are given by

Table III. Computed Values of $C(\mathbf{q})$ and $\partial C(\mathbf{q})/\partial \mathbf{q}$ as Function of M

	$\rho = 0.5$		$\rho = 0.8$		
M	$C(\mathbf{q})$	$\partial C(\mathbf{q})/\partial \mathbf{q}$	$C(\mathbf{q})$	$\partial C(\mathbf{q})/\partial \mathbf{q}$	CPU
5	0.881	(−0.044,0.003,0.046)	2.938	(−0.089,0.010,0.101)	0:00:00
10	0.878	(−0.093,−0.004,0.102)	2.975	(−0.333,−1.394,−0.414)	0:00:01
15	0.878	(−0.094,−0.003,0.103)	2.962	(−0.451,−0.345,0.268)	0:00:03
20	0.878	(−0.094,−0.003,0.103)	2.958	(−0.451,−0.365,0.448)	0:00:10
25	0.878	(−0.094,−0.003,0.103)	2.957	(−0.452,−0.370,0.391)	0:00:23
30	0.878	(−0.094,−0.003,0.103)	2.957	(−0.452,−0.369,0.392)	0:00:44
40	0.878	(−0.094,−0.003,0.103)	2.957	(−0.452,−0.369,0.392)	0:02:13

Table IV. Optimal Bernoulli Schedules $\mathbf{q}^*(M)$ as Function of M

	$\rho = 0.5$			$\rho = 0.8$		
M	$\mathbf{q}^*(M)$	$C(\mathbf{q}^*(M))$	CPU	$\mathbf{q}^*(M)$	$C(\mathbf{q}^*(M))$	CPU
5	(0.50,0.50,0.50)	0.927	0:00:00	(0.50,0.50,0.50)	3.249	0:00:00
10	(1.00,0.60,0.00)	0.844	0:00:10	(1.00,0.57,0.63)	3.010	0:00:03
15	(1.00,0.58,0.00)	0.844	0:00:33	(1.00,0.79,0.43)	2.888	0:00:39
20	(1.00,0.58,0.00)	0.844	0:01:35	(1.00,0.82,0.42)	2.887	0:04:09
25	(1.00,0.58,0.00)	0.844	0:02:46	(1.00,0.88,0.45)	2.885	0:05:39
30	(1.00,0.58,0.00)	0.844	0:05:27	(1.00,0.88,0.45)	2.885	0:09:52
40	(1.00,0.58,0.00)	0.844	0:16:25	(1.00,0.88,0.45)	2.885	0:28:45

$(c_1, c_2, c_3) = (0.4, 0.2, 0.2)$. For Bernoulli schedule $\mathbf{q} = (1.0, 0.5, 0.5)$, Table III shows the computed performance measures, $C(\mathbf{q})$, their derivarives with respect to the Bernoulli parameters, $\partial C(\mathbf{q})/\partial \mathbf{q}$, and the required amount of computation time, CPU (in seconds on a SUN SPARC IPX), as function of M for various values of the offered load to the system ρ. Note that once the coefficients of the power series have been computed, the value of $C(\mathbf{q})$ can be determine almost instantaneously for different values of ρ. Table III illustrates that the required number of terms of the power series that have to be computed increases with increasing system load. In the specific model considered in Table III for $\rho = 0.5$ only 10 or 15 terms suffice to give rather accurate computations, while the required computation time is only a few seconds. For $\rho = 0.8$ about 25 to 30 terms will be needed to achieve good results, requiring extra computation time.

To illustrate the performance of optimization procedures based on the use of the PSA, consider the problem of finding a Bernoulli vector $\mathbf{q}^* = (q_1, \ldots, q_s)$ which minimizes the cost function $C(\mathbf{q})$ (cf. (32)) over all Bernoulli schedules $\mathbf{q} \in [0, 1]^s$ for which the system is stable. We have computed the optimal Bernoulli vector $\mathbf{q}^*(M)$ for various values of M, the number of terms of the power series, with initial Bernoulli schedule (0.5, 0.5, 0.5). Table IV shows the computed optimal Bernoulli vectors $\mathbf{q}^*(M)$, the cost corresponding to these optima (computed with a large number of terms) and the computation times, CPU, as function of the maximal number of terms, M.

An alternative approach which is generally applicable, is to use the PSA with a small number of terms to find the neighbourhood of the optimal schedule with reduced computational effort, and then proceed with the PSA with more terms to locally improve the optimal schedule. In the above-mentioned example with $\rho = 0.8$ one could, e.g., use the PSA with $M = 15$ to find a solution in the neighbourhood of the optimum rather quickly (cf. Table IV) and then use the PSA with $M = 25$ or $M = 30$ to locally improve this solution to find an optimum. We emphasize that this procedure is generally applicable for optimization by means of the PSA, and goes far beyond the specific optimization problem considered in this section.

The extension of PSA demonstrated here is also applicable to optimization of polling systems with Bernoulli schedules with other types of (non-cyclic) server routing mechanisms such as random (Markovian) server routing, or dynamic server routing (e.g., priority for the longest queue).

In this section the use of exponential distributions for the service times and switch-over times mainly served the ease of the discussion. When Coxian distributions are used to approximate service times and switch-over times, the size of the supplementary space is increased. More precisely, if service times at Q_i and times needed by the server to move towards Q_i are approximated by Coxian distributions with Ψ_i^1 and Ψ_i^0 phases, respectively, the size of the supplementary space is equal to

$$|\Theta| := \sum_{i=1}^{s} \{\Psi_i^0 + \Psi_i^1\} \geq 2s. \tag{33}$$

6. Concluding Remarks

In this paper we have extended the use of the PSA for a broad class of QBD processes to the computation of derivatives of system performance measures with respect to continuous system parameters. It is illustrated that the extension is useful for solving optimization problems for moderately-sized queueing systems for which no exact solutions exist. This is important for deriving sharp approximations for non-exactly solvable optimization problems for large systems, as illustrated in [6].

Recent developments have indicated that the PSA is also applicable to systems for which the QBD structure is violated. In Van den Hout and Blanc[13] the use of the PSA is extended to systems with Batch Markovian Arrival Processes (BMAPs). In [14] the PSA is further extended to the general class of so-called Markovian queueing networks, in which the arrival process is a Multi-queue Markovian Arrival Process (MMAP), which is a multi-queue generalization of the BMAP. Recently, Koole[10] has shown that the PSA is, in principle, applicable to general Markov processes. The computation of derivatives presented in this paper is readily applicable in the context of these generalization of the PSA.

References

1. J.P.C. BLANC, 1987. On a Numerical Method for Calculating State-Probabilities for Queueing Systems with More Than One Waiting Line, *J. Comp. Appl. Math. 20*, 119–125.
2. J.P.C. BLANC, 1990. A Numerical Approach to Cyclic-Service Queueing Models, *Queueing Systems 6*, 173–188.
3. J.P.C. BLANC, 1992. Performance Evaluation of Polling Systems by Means of the Power-Series Algorithm, *Ann. Oper. Res. 35*, 155–186.
4. J.P.C. BLANC, 1993. Performance Analysis and Optimization with the Power-Series Algorithm, in *Performance Evaluation of Computer and Communication Systems*, L. Donatiello and R. Nelson (eds.), North-Holland, Amsterdam, pp. 53–80.
5. J.P.C. BLANC and R.D. VAN DER MEI, 1992. Optimization of Polling Systems by Means of Gradient Methods and the Power-Series Algorithm, Technical Report FEW 575, Tilburg University.
6. J.P.C. BLANC and R.D. VAN DER MEI, 1995. Optimization of Polling Systems with Bernoulli Schedules, *Perf. Eval. 22*, 139–158.
7. O.J. BOXMA and W.P. GROENENDIJK, 1987. Pseudo-Conservation Laws in Cyclic-Service Systems, *J. Appl. Prob. 24*, 949–964.
8. C. FRICKER and M.R. JAIBI, 1994. Monotonicity and Stability of Polling Models, *Queueing Systems 15*, 211–238.
9. P.E. GILL, W. MURRAY, and M.H. WRIGHT, 1981. *Practical Optimization*, Academic Press, New York.
10. G. KOOLE, 1994. On the Power-Series Algorithm, CWI Technical Report BS-R9404, Amsterdam.
11. H. LEVY and M. SIDI, 1990. Polling Systems: Applications, Modeling and Optimization, *IEEE Trans. Commun. 38*, 1750–1760.
12. S.S. RAO, 1978. *Optimization Theory and Applications*, Wiley Eastern Limited, New Delhi.
13. W.B. VAN DEN HOUT and J.P.C. BLANC, 1995. Development and Justification of the Power-Series Algorithm for BMAP-Systems, *Stochastic Models 11*, 471–496.
14. W.B. VAN DEN HOUT and J.P.C. BLANC, 1995. The Power-Series Algorithm for Markovian Queueing Networks, in *Computations with Markov Chains*, W. J. Stewart (ed.), Kluwer, pp. 321–338.
15. P. WYNN, 1966. On the Convergence and Stability of the Epsilon Algorithm, *SIAM J. Num. Anal. 3*, 91–122.