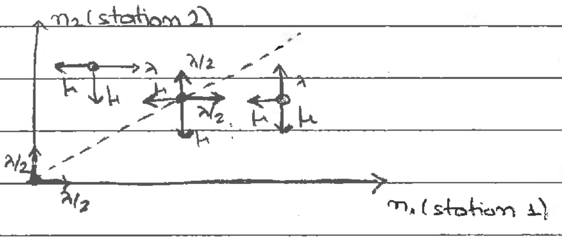
Power Series Algorithm (PSA)

The PSA is a numerical - analytic method for analyzing certain Markov processes. The main idea of the PSA is to consider the limiting distribution as a function of the system's parameter, say ϑ, e.g $\vartheta = \lambda, \rho, ..$ Then, assume that the limiting distribution is an analytic function of the system's parameter around some value, typically zero, and consider the corresponding power-series expansion $\sum_{k=0}^{\infty} \vartheta^k \alpha(k, \underline{n})$ for the limiting distribution at state $\underline{n} = (n_1, n_2)$. Substituting the power series expansion back into the balance equations, we can recursively retrieve the coefficient $\alpha(k, \underline{n})$ and build the limiting distribution.

The principal idea of PSA is originally due to Benes [10], and it was independently rediscovered by Hooghiemstra, Keane and Van de Ree [11]. In a series of papers, Blanc and co-authors have greatly extended the applicability of the PSA, see, eg [12]. The major advantages of the PSA are its wide applicability to any MC with a single recurrent class [13] and its flexibility since its main assumption lies on the Markovian assumption but does not require any further structure.

PSA example : Join the shortest queue (JSQ)

Consider two $\text{Exp}(\mu)$ parallel servers, each with their own queue. Customers arrive according to a Poisson process at rate λ. An arriving customer chooses the shortest queue and in case of a tie, the customer chooses either queue with probability $1/2$. Let $X_i(t)$ denote the number of customers at the i-th server at time t, $i = 1, 2$. The stochastic process $\{(X_1(t), X_2(t)), t \geq 0\}$, with state space $S = \mathbb{N}_0 \times \mathbb{N}_0$, is Markovian with rate transition diagram as depicted below



The balance equations for this model are

$$\left(\lambda + \mu \sum_{j=1}^{2} \mathbb{1}\{n_j > 0\}\right) p(n_1, n_2) = \mu\, p(n_1+1, n_2) + \mu\, p(n_1, n_2+1) + \lambda\, \mathbb{1}\{n_1 > 0\}\, p(n_1-1, n_2)\, \mathbb{1}\{n_1 < n_2\} + \lambda\, \mathbb{1}\{n_2 > 0\}\, p(n_1, n_2-1)\, \mathbb{1}\{n_1 > n_2\} + \frac{\lambda}{2} \mathbb{1}\{n_1 > 0\}\, p(n_1-1, n_2)\, \mathbb{1}\{n_1 - 1 = n_2\} + \frac{\lambda}{2} \mathbb{1}\{n_2 > 0\} \cdot p(n_1, n_2-1)\, \mathbb{1}\{n_1 = n_2 - 1\}$$

with $\mathbb{1}\{\cdot\}$ the indicator function taking value 1 if the event inside the curly brackets $\{\cdot\}$ is satisfied and 0, otherwise.

Equivalently the balance equations are written

$$\left(\rho + \frac{1}{2}\sum_{j=1}^{2} \mathbb{1}\{n_j>0\}\right) p(\underline{n}) = \frac{1}{2}\sum_{j=1}^{2} p(\underline{n}+e_j) + \rho \sum_{j=1}^{2} \mathbb{1}\{n_j>0\}\, p(\underline{n}-e_j)\, P(C=j\,|\,\underline{n}-e_j)$$

with $\rho = \frac{\lambda}{2\mu}$, $\underline{n}=(n_1,n_2)$, $e_1=(1,0)$, $e_2=(0,1)$ and $P(C=j\,|\,\underline{n}-e_j)$ is the probability an arriving customer finding the system in state $\underline{n}-e_j$ chooses the j-th queue/server.

Choose $p(\underline{n}) := p(\rho;\underline{n}) := \rho^{n_1+n_2}\sum_{k=0}^{\infty}\rho^k \alpha(k;\underline{n})$, so that $\rho^{-(n_1+n_2)}p(\underline{n})$ converges as $\rho \to 0$. Subsituting the power-series expression into the balance equations yields.

$$\left(\rho + \frac{1}{2}\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\right)\rho^{n_1+n_2}\sum_{k=0}^{\infty}\rho^k\alpha(k;\underline{n}) = \frac{1}{2}\sum_{j=1}^{2}\rho^{n_1+n_2+1}\sum_{k=0}^{\infty}\rho^k\alpha(k;\underline{n}+e_j) +$$

$$+\rho\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\,\rho^{n_1+n_2-1}\sum_{k=0}^{\infty}\rho^k\alpha(k;\underline{n}-e_j)\,P(C=j\,|\,\underline{n}-e_j) \Rightarrow$$

$$\sum_{k=1}^{\infty}\rho^k\alpha(k-1;\underline{n}) + \frac{1}{2}\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\sum_{k=0}^{\infty}\rho^k\alpha(k;\underline{n}) = \frac{1}{2}\sum_{j=1}^{2}\sum_{k=1}^{\infty}\rho^k\alpha(k-1;\underline{n}+e_j) +$$

$$+\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\sum_{k=0}^{\infty}\rho^k\alpha(k;\underline{n}-e_j)\,P(C=j\,|\,\underline{n}-e_j)$$

Equating the coefficient of ρ^k (we can do this since the power-series are analytic) yields

$$\alpha(k-1;\underline{n}) + \frac{1}{2}\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\,\alpha(k;\underline{n}) = \frac{1}{2}\sum_{j=1}^{2}\alpha(k-1;\underline{n}+e_j)\,\mathbb{1}\{k>0\}$$

$$+\sum_{j=1}^{2}\mathbb{1}\{n_j>0\}\,\alpha(k;\underline{n}-e_j)\,P(C=j\,|\,\underline{n}-e_j) \qquad (18)$$

In addition, the normalising equation yields

$$\sum_{n_1,n_2} p(n_1,n_2) = 1 \Rightarrow \sum_{n_1,n_2}\sum_{k=0}^{\infty}\rho^{k+n_1+n_2}\alpha(k,\underline{n}) = 1$$

Equating the powers of ρ yields

$$\alpha(0,0)=1 \quad \text{and} \quad \sum_{\substack{n_1+n_2+k=K\\ n_1,n_2\geq 0}}\rho^{k+n_1+n_2}\alpha(k,\underline{n})=0$$

$$\Rightarrow \sum_{\substack{n_1+n_2+k=K\\ n_1,n_2=0}}\rho^{k+n_1+n_2}\alpha(k,\underline{n}) + \sum_{\substack{n_1+n_2+k=K\\ n_1,n_2\neq 0}}\rho^{k+n_1+n_2}\alpha(k,\underline{n}) = 0$$

$$\Rightarrow \alpha(K,\underline{0}) + \sum_{0<n_1+n_2\leq K}\alpha(K-(n_1+n_2),\underline{n}) = 0, \quad K=1,2,\ldots \qquad (19)$$

To obtain the coefficients of $p(\rho;\underline{n})$ up to power ρ^M proceed from $k=0$ to M; start from $\alpha(0,\underline{0})=1$; then, determine $\alpha(k,\underline{0})$ from (19); then, calculate $\alpha(k;\underline{n})$ recursively from (18) for increasing values of $0<\sum n_i \le M-k$.

PSA algorithm for one dimensional state space

for $k=0$ to M do
    if $k=0$ then $\alpha(0,0) \leftarrow 1$
        else $\alpha(k,0) \leftarrow -\sum_{0<n\le x} \alpha(k-n,n)$
    endif
    for $n=1$ to $M-k$ do

$$\alpha(k,n) \leftarrow \Big(\sum_{0\le m<n} q_{m,n}\,\alpha(k,m) + \sum_{n<m\le k+n} q_{m,n}\,\alpha(k-m+n,m) - \sum_{n<m\le k+n} q_{n,m}\,\alpha(k-m+n,m)\Big) \Big/ \sum_{0\le m<n} q_{n,m}$$

    endfor
endfor

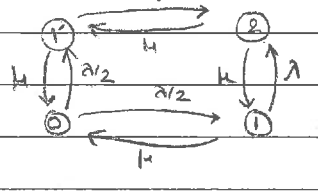
Convergence of PSA

The algorithmic approach derived for the JSQ example through the power-series of $p(\rho;\underline{n})$ as a function of ρ is not convergent on the whole interval $\rho\in(0,1)$ for this model. To enlarge the radius of convergence of the power series we use the conformal mapping

$$\vartheta = \frac{1+G}{1+G\rho}\rho \quad (\text{with } G\ge 0)$$

that maps $(0,1)$ into itself and introduce $p(\rho(\vartheta);\underline{n}) = \vartheta^{n_1+n_2}\sum_{k=0}^{\infty}\vartheta^k b(k;\underline{n})$.
Substituting the above power-series expression into the balance equations and equating the coefficients of the corresponding power of ϑ yields a set of equations from which the coefficients can be obtained recursively. In this case, the radius of convergence is larger than the original one.

PSA example: truncated JSQ



This yields $p_0 = \frac{2\mu^2}{\lambda^2+2\lambda\mu+2\mu^2}$, $p_1 = p_{1'} = \frac{\lambda\mu}{\lambda^2+2\mu\lambda+2\mu^2}$, $p_2 = \frac{\lambda^2}{\lambda^2+2\lambda\mu+2\mu^2}$ $\Longrightarrow$

$$p_0 = \frac{1}{2\rho^2+2\rho+1}, \quad p_1 = p_{1'} = \frac{\rho}{2\rho^2+2\rho+1}, \quad p_2 = \frac{2\rho^2}{2\rho^2+2\rho+1}$$

Now consider p_n as a function of ρ. Note that all p_n are rational, thus are analytic in ρ around 0, because $2\rho^2+2\rho+1$ has no zero in $\rho=0$. Therefore, the power series expansions around zero exist.

Because $p_0 = \frac{1}{2\rho^2+2\rho+1}$ and the zeros of the denominator are $(-1 \pm i)/_2$, the power-series converge in the interval $\rho \in [0, \frac{1}{\sqrt{2}}) \approx [0, 0.707)$. Since all p_n have the same denominator the radius of convergence is $[0, 1/\sqrt{2})$, making the partial sums found by the PSA to converge up to $\rho = 1/\sqrt{2}$, but beyond this value the series diverge.

Considering the mapping discussed before the radius would change.

Instead, we consider

$$p_0 = 1, \quad p_1 = p_{1'} = 2\rho \quad \text{and} \quad p_2 = 2\rho^2$$

Notice that now the radius of convergence covers $\mathbb{R}+$. Thus, by normalising afterwards we increase the region of convergence. This is the PSA|N variant, see [11].

Intermezzo: the ε-algorithm

The ε-algorithm was introduced by Wynn see [14] to accelerate the convergence of power series. Given $S_m = \sum_{k=0}^{m} c_k \rho^k$, a 2-dimensional array with entries $\varepsilon_r^{(m)}$ is computed using the formula

$$\varepsilon_{r+1}^{(m)} = \varepsilon_{r-1}^{(m+1)} + \left(\varepsilon_r^{(m+1)} - \varepsilon_r^{(m)}\right)^{-1},$$

with initial conditions

$$\varepsilon_{-1}^{(m)} = 0, \quad m = 1, 2, \dots, \quad \text{and}$$

$$\varepsilon_0^{(m)} = S_m, \quad m = 0, 1, \dots$$

Now $\varepsilon_{2r}^{(m)}$ is used to approximate S_∞, while the odd sequences $\varepsilon_{2r+1}^{(m)}$ are just intermediate steps in the calculation scheme. Note that the ε-algorithm transforms the sequence of polynomials to sequences of quotients of two polynomials. More precisely, $\varepsilon_{2r}^{(m-2r)}$ is a quotient of a polynomial of degree $m-r$ over a polynomial of degree r, and

$$|S_m - \varepsilon_{2r}^{(m-2r)}| = O(\rho^{m+1}) \quad \rho \to 0, \quad r = 0, 1, \dots, \quad m = 2r, 2r+1, \dots$$

In [16], Blanc proposes to modify the ε-algorithm, based on the observation that for many queueing models the ν-th moments are of the order $(1-\rho)^{-\nu}$, as $\rho \to 1$, $\nu = 1, 2, \dots$

So for the calculation of moments of first order take

$$\varepsilon_0^{(m)} = S_m + c_m \frac{\rho^{m+1}}{1-\rho}, \quad m = 1, 2, \dots$$

and for second order moments take

$$\varepsilon_0^{(m)} = S_m + \left[c_m + \frac{c_m - c_{m-1}}{1-\rho}\right] \frac{\rho^{m+1}}{1-\rho}, \quad m = 1, 2, \dots$$

instead of $\varepsilon_0^{(m)} = S_m$. This leads to considerably faster convergence. E.g. notice that if $\varepsilon_0^{(m)} = S_m$, then $\varepsilon_2^{(m-2)} = S_m + \dfrac{c_m^2}{c_{m-1}} \dfrac{\rho^{m+1}}{\left(1 - \frac{c_m}{c_{m-1}}\rho\right)}$

while if $\varepsilon_0^{(m)} = S_m + c_m \dfrac{\rho^{m+1}}{1-\rho}$, then

$$\varepsilon_2^{(m-2)} = S_m + c_m \frac{\rho^{m+1}}{1-\rho} + \frac{(c_m - c_{m-1})^2}{c_{m-1} - c_{m-2}} \frac{\rho^{m+1}}{(1-\rho)\left(1 - \frac{c_m - c_{m-1}}{c_{m-1} - c_{m-2}}\rho\right)}$$

It is immediately evident that $\rho = 1$ is a pole in the second expression and it can be numerically validated that the second expression is a better approximation of S_∞. So if we have prior knowledge about poles, then it makes sense to use the above described variation of the ε-algorithm, and perhaps even combine it with the mapping so as to extend the radii of convergence. Because the ε-algorithm transforms polynomials into rational functions, it is not necessary to choose the value of G, in the transformation $\vartheta = \dfrac{1+G}{1+G\rho}\rho$, very large, e.g. choose for the JSQ $G = \dfrac{1}{2}$ for a convergent result.

Remark: The ε-algorithm is a numerical procedure that converts partial sums of power-series into the quotients of polynomials and a remainder term

$$\varepsilon_{2r}^{(m-2r)} = \frac{\text{pol of degree } m-r}{\text{pol. of degree } r} + \text{remainder}$$

In case, the series has poles, the zeros of the denominator will converge to the poles. In case of prior knowledge of a pole, this effect can be accelerated by varying appropriately $\varepsilon_0^{(m)}$.

In case of finite HCs the PSA together with the ε-algorithm, if applicable, give exact results.

References

[10] VE Beneš (1965) Mathematical Theory of Connecting Networks and Telephone Traffic. Academic Press.

[11] G. Hooghiemstra, M. Keane, and S. van de Ree (1988) Power series for stationary distributions of coupled processor models. SIAM Journal of Applied Mathematics, 48: 1159-1166.

[12] JPC Blanc (1993) Performance analysis and optimization with the power-series algorithm. In L. Donatiello and R.D. Nelson, editors, Performance Evaluation of Computer and Communication Systems, pages 53-80.

[13] GM Koole (1994). On the power series algorithm. In OJ Boxma and GM Koole, editors, Performance Evaluation of Parallel and Distributed Systems,

pages 139–155.

[14] P Wynn (1996) On the convergence and stability of the epsilon algorithm. SIAM Journal of Numerical Analysis 3: 91–122.

[15] JPC Blanc (1990) A numerical approach to cyclic service queueing models. Queueing Systems 6(1): 173–188.