Structured Markov chain solver: MATLAB tool extensions

D. Bini, B. Meini, J.F. Perez and B. Van Houdt

The notations used in this document are in line with the papers *Structured Markov chains solver* [2, 3]. These notations are slightly different from the MATLAB help files. General references to the issues addressed in this document can be found in the books [5, 6, 4, 1].

1 Quasi-Birth-Death (QBD) Markov chains

Continuous time Markov chains (CTMC)

The focus in the original paper was on discrete-time Markov chains only. Therefore, QBD Markov chains were defined by a transition matrix P of the form

$$P = \begin{bmatrix} B_0 & B_1 & & & 0 \\ B_{-1} & A_0 & A_1 & & \\ & A_{-1} & A_0 & A_1 & \\ & & A_{-1} & A_0 & \ddots \\ 0 & & & \ddots & \ddots \end{bmatrix},$$

where $A_{-1}, A_0, A_1 \in \mathbb{R}^{m\times m}$, $B_0 \in \mathbb{R}^{m_b\times m_b}$, $B_{-1} \in \mathbb{R}^{m\times m_b}$ and $B_1 \in \mathbb{R}^{m_b\times m}$, are nonnegative matrices such that $A_{-1}+A_0+A_1$, $B_{-1}+A_0+A_1$ and B_0+B_1 are stochastic (where $B_1 = A_1$ in the default mode of operation).

The CTMC extension allows the user to solve continuous time QBD problems using the same set of MATLAB functions (i.e., QBD_CR, QBD_FI, QBD_IS, QBD_LR, QBD_NI and QBD_pi). In the continuous time case, the rate matrix Q characterizing the QBD has the same form as P except that the row sums of Q equal zero, while P is stochastic. Moreover, the only negative entries of Q appear on the main diagonal of A_0 and B_0.

In the continuous time case the R, G and U matrices solve the following set of equations:

$$\begin{aligned} 0 &= A_{-1} + A_0 G + A_1 G^2, \\ 0 &= A_1 + R A_0 + R^2 A_{-1}, \\ U &= A_0 + A_1(-U)^{-1} A_{-1}. \end{aligned} \tag{1}$$

Furthermore, one has $G = (-U)^{-1} A_{-1}$, $R = A_1(-U)^{-1}$. When the input is a continuous time QBD, it is automatically transformed to a discrete time problem (via a uniformization) such that the $\tilde{G}$, $\tilde{R}$ and $\tilde{U}$ solution of the discrete time problem obeys: $G = \tilde{G}$, $R = \tilde{R}$ and $U = \lambda(\tilde{U} - I)$ (with $\lambda = \max(-diag(A_0)))$. The G, R and U matrix of the continuous time problem are returned as output.

If the continuous time QBD is positive recurrent, the following set of equations hold

$$\begin{aligned} &\boldsymbol{\pi}_n = \boldsymbol{\pi}_1 R^{n-1}, \\ &\boldsymbol{\pi}_0 B_0 + \boldsymbol{\pi}_1 B_{-1} = 0, \\ &\boldsymbol{\pi}_0 B_1 + \boldsymbol{\pi}_1 (A_0 + R A_{-1}) = 0, \\ &\boldsymbol{\pi}_0 e + \boldsymbol{\pi}_1 (I - R)^{-1} \boldsymbol{e} = 1, \end{aligned}$$

for $n > 1$. The QBD_pi function automatically detects the continuous nature of the problem and will solve it by transforming it to a discrete time problem, having $\tilde{\boldsymbol{\pi}}$ as a solution, such that $\tilde{\boldsymbol{\pi}} = \boldsymbol{\pi}$.

2 M/G/1-type Markov chains

Alternate Shift: *ShiftType* option

The computation of the G matrix is accelerated (in the default mode) by the MG1_CR, MG1_NI and MG1_FI functions by applying the shift technique. Assume that $A_i = 0$ for $(i > M)$. The default shift technique (*ShiftType* = 'one') makes use of the following blocks $\widetilde{A}_i$, $i \geq -1$. If the chain is positive recurrent, set

$$\begin{aligned}\widetilde{A}_{-1} &= A_{-1}(I - Q),\\ \widetilde{A}_i &= A_i - (\textstyle\sum_{j=-1}^{i} A_j - I)Q, \quad 0 \le i \le M,\end{aligned}$$

where $Q = \boldsymbol{e}\boldsymbol{u}^T$ and $\boldsymbol{u}$ is any vector such that $\boldsymbol{e}^T\boldsymbol{u} = 1$. Otherwise, set

$$\begin{aligned}\widetilde{A}_{-1} &= A_{-1}\\ \widetilde{A}_0 &= A_0 + EA_{-1}\\ \widetilde{A}_i &= A_i - E(I - \textstyle\sum_{j=-1}^{i-1} A_j), \quad 1 \le i \le M,\end{aligned}$$

where $E = \boldsymbol{u}\boldsymbol{v}^T$, with $\boldsymbol{u}$ being any nonzero vector, and $\boldsymbol{v}$ such that $\boldsymbol{v}^T\boldsymbol{u} = 1$ and $\boldsymbol{v}^T(\sum_{i=-1}^{M} A_i) = \boldsymbol{v}^T$.

It has been proved that the roots of the polynomials $\widetilde{a}(z) = \det(\lambda I - \sum_{i=-1}^{M} z^{i+1}\widetilde{A}_i)$, and $a(z) = \det(\lambda I - \sum_{i=-1}^{M} z^{i+1} A_i)$, are the same except for the root $z = 1$ of $a(z)$ which is shifted to zero or to the infinity for $\widetilde{a}(z)$ according to the recurrent or transient nature of the chain under consideration.

We can also accelerate the convergence by shifting the largest root $0 \le \tau < 1, \tau \in \mathbb{R}$ to zero (in the transient case) or the smallest root $\hat{\tau} > 1, \hat{\tau} \in \mathbb{R}$ to infinity (in the positive recurrent case). Setting the *ShiftType* = 'tau', causes this type of shift operation. The τ and $\hat{\tau}$ roots are computed efficiently through a bisection algorithm. Whether the 'one' or 'tau' shift is the most effective depends on the numerical example at hand.

Finally, the double shift (activated by setting the *ShiftType* option to 'dbl') combines both the 'one' and the 'tau' shift and typically requires the least number of iterations. In the positive recurrent case, the smallest root $\hat{\tau} > 1, \hat{\tau} \in \mathbb{R}$ is first shifted to infinity, afterwards the zero in $z = 1$ is shifted to zero. In the transient case, we first shift the zero in $z = 1$ to infinity and next shift the the largest root $0 \le \tau < 1, \tau \in \mathbb{R}$ to zero.

Newton iteration

The computation of the matrix G via a Newton iteration is now also supported (via the function MG1_NI) and relies on a fast version of Newton's iteration for M/G/1-type Markov chains [7]. At each iteration a linear system of the following form needs to be solved

$$\sum_{j=1}^{M+1} B_j X A^{j-1} = C,$$

where M is the the largest integer such that $A_M > 0$ and m is the block size.

Such a system can be solved using a direct sum approach (which happens when the option *Mode* is set to *DirectSum*), but this results in a time complexity of $O(m^6 + m^4 M)$, as is the case for the Newton iteration discussed in [1]. By relying on a Schur decomposition of A in the linear system above, the time complexity per iteration can be reduced to $O(Mm^4)$. The modes *RealSchur* and *ComplexSchur* implement this approach using a real and complex Schur decomposition, respectively. Additionally, each of these modes can also be combined with a shift operation, the *RealSchurShift* mode is the default mode for the MG1_NI function.

Two more functions: MG1_NI_LRA0 and MG1_NI_LRAi further exploit potential low rank properties of the matrices A_i. The MG1_NI_LRA0 function reduces the time complexity per iteration to $O(Nm^2r^2 + m^3r)$ when A_{-1} can be decomposed as $A_{-1} = \hat{A}_{-1}\Gamma$, where $\hat{A}_{-1}$ is of size $m \times r$ and Γ of size $r \times m$, with $r < m$. The MG1_NI_LRAi function reduces the time complexity per iteration to $O(Nm^2r^2 + Nm^3)$ when A_i, for $i \geq 0$, can be decomposed as $A_i = \Gamma\hat{A}_i$, where $\hat{A}_i$ is of size $r \times m$ and Γ of size $m \times r$.

3 GI/M/1-type Markov chains

New parameter 'Dual'

In the first edition of this tool, the R matrix of a GI/M/1-type Markov chain was computed by computing the G matrix of its Ramaswami dual, from which R can be obtained easily. In the current version of the tool, the user can select one of two duals: the Ramaswami or Bright dual, by setting the 'Dual' parameter to 'R' or 'B', respectively. Setting the 'Dual' parameter to 'A' (Automatic) causes the software to select the Bright dual for positive recurrent chains and the Ramaswami dual for transient ones as this choice is typically the most efficient (for null recurrent chains, both duals are identical).

The Ramaswami dual of a GI/M/1-type Markov chain characterized by $(A_i)_{i\geq -1}$ is an M/G/1-type Markov chain characterized by the series of matrices $(A_i^{(r)})_{i\geq -1}$ with

$$A_i^{(r)} = (\Delta^{(r)})^{-1} A_i' (\Delta^{(r)}),$$

for $i \geq -1$ and $\Delta^{(r)} = diag(\pi)$, where π is the stochastic left-invariant vector of $A = \sum_{i\geq -1} A_i$.

The Bright dual of a GI/M/1-type Markov chain characterized by $(A_i)_{i\geq -1}$ is an M/G/1-type Markov chain characterized by the series of matrices $(A_i^{(b)})_{i\geq 0}$ with

$$A_i^{(b)} = (\Delta^{(b)})^{-1} A_i' (\Delta^{(b)}) \tau^i,$$

for $i \geq -1$ and $\Delta^{(b)} = diag(w)$, where w is the positive stochastic left-eigenvector of $A(\tau)$ with eigenvalue τ, where $A(z) = \sum_{i=0}^{\infty} A_{i-1} z^i$. The scalar τ depends on whether the chain is positive recurrent or transient. In the positive recurrent case we have: $\tau = sp(R) < 1$ the spectral radius of R, while in the transient case $\tau > 1$ is the smallest zero of $\det(zI - A(z))$ outside the unit circle.

The G matrix of the Ramaswami or Bright dual, denoted as $G^{(r)}$ and $G^{(b)}$ respectively, satisfy the following relation with the R matrix of the original GI/M/1-type Markov chain:

$$R = (\Delta^{(r)})^{-1} (G^{(r)})' (\Delta^{(r)}) = (\Delta^{(b)})^{-1} (G^{(b)})' (\Delta^{(b)}) \tau,$$

allowing us to obtain R from the G matrix of its dual.

Newton iteration

As for the M/G/1-type Markov chains, a fast Newton iteration is also incorporated into the tool. It can be used to compute R by calling the GIM1_R function with the *Algor* parameter set to *NI*.

The two new functions GIM1_NI_LRA0 and GIM1_NI_LRAi exploit potential low rank properties of the matrices A_i. These functions rely on their M/G/1-type counterparts using either the Bright or Ramaswami dual. Notice, in order to use the GIM1_NI_LRAi function the matrices A_i must be expressed as $\hat{A}_i\Gamma$, for $i \geq 0$ (as opposed to $\Gamma\hat{A}_i$ as in the M/G/1-type case).

References

[1] D. A. Bini, G. Latouche, and B. Meini. *Numerical methods for structured Markov chains*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005. Oxford Science Publications.

[2] D. A. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: algorithms. In *SMCtools Workshop*, Pisa, Italy, 2006. ACM Press.

[3] D. A. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: software tools. In *SMCtools Workshop*, Pisa, Italy, 2006. ACM Press.

[4] G. Latouche and V. Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[5] M. F. Neuts. *Matrix-geometric solutions in stochastic models*, volume 2 of *Johns Hopkins Series in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, Md., 1981. An algorithmic approach.

[6] M. F. Neuts. *Structured stochastic matrices of $M/G/1$ type and their applications*, volume 5 of *Probability: Pure and Applied*. Marcel Dekker Inc., New York, 1989.

[7] J. Perez, M. Telek, and B. Van Houdt. A fast Newton's iteration for M/G/1-type and GI/M/1-type Markov chains. *Submitted*, 2011.