Decision diagrams for the approximate analysis of Markov models

Gianfranco Ciardo[1 *]

[1] Department of Computer Science and Engineering, University of California, Riverside, USA

Decision diagrams of various types can be used to encode the exact state space and transition rate matrix of large Markov models. However, the exact solution of such models still requires to store at least one real vector with one entry per reachable state, a formidable limitation to the practical use of these encodings. Thus, we discuss automatic techniques for the approximate computation of performance measures when the Markov model can be compactly encoded but not exactly solved.

1 Running example: a queuing network with fork-and-join behavior

We recall how the exact state space $\mathcal{S}$ and the transition rate matrix $\mathbf{R}$ of a discrete-state model with an underlying continuous-time Markov chain (CTMC) can be compactly encoded using decision diagrams through a running example. For details about the data structures and algorithms employed to build these encodings, see [1, 2].

Fig. 1 shows a stochastic Petri net (SPN) of a fork-join queueing network, including the transition firing rates, where $\#(x)$ means the number of tokens (customers) in place (queue) x in the current marking (state), and transition t_5 is immediate, meaning it fires as soon as it is enabled. Given the *local state spaces* shown in Fig. 2, the MDD encoding state space $\mathcal{S}$ of this model is shown in Fig. 3. Each path from node $root$ at level 5 to a node at level 1 corresponds to a state in $\mathcal{S}$ (e.g., path 21200 corresponds to state $f^1e^0d^1c^2b^0a^0$, that is, the state where there are two customers in c and one customer in f and d). Let $\mathcal{L}_k$ be the set of nodes at level k (e.g., $\mathcal{L}_4 = \{m, n, p, q, r\}$), let $nd(x, \gamma)$ be the node reached from node x following the sequence γ (e.g., $nd(p, 12) = s$), let $\mathcal{A}(x)$ be the set of substates "above" node x (e.g., $\mathcal{A}(p) = \{2\}$), and let $\mathcal{B}(x)$ be the set of substates "below" node x (e.g., $\mathcal{B}(p) = \{0101, 0110, 1200\}$ and $\mathcal{B}(root) = \mathcal{S}$).

Fig. 4 shows the CTMC underlying this SPN, while Fig. 5 shows the EV*MDD encoding the transition rate matrix of this CTMC, where each path from node $Root$ at level 5 to a node at level 1 corresponds to a nonzero entry in the matrix. Let $X[i_k, j_k]$ be the (i_k, j_k)-edge from node X (e.g., $N[1, 0] = (1/2, S)$, $N[1, 0].d = S$, $N[1, 0].v = 1/2$), let $\mathcal{M}_k$ be the set of nodes at level k (e.g., $\mathcal{M}_4 = \{M, N, O, P, Q\}$), let $nd(X,\gamma,\gamma')$ be the node reached from node X at level k following the interleaving of the two sequences γ and γ' (e.g., $nd(N, 100, 000) = T$), and let $vl(X,\gamma,\gamma')$ be the product of the values along the path (e.g., $vl(N, 100, 000) = 1/2 \cdot 1 \cdot 1 = 1/2$). Then, given any two states $\mathbf{i} = (i_L, ..., i_1)$ and $\mathbf{j} = (j_L, ..., j_1)$ in $\mathcal{S}$, $\mathbf{R}[\mathbf{i}, \mathbf{j}] = vl(Root, \mathbf{i}, \mathbf{j})$; also, if we write $\mathbf{i} = (\alpha, i_k, \beta)$ and $\mathbf{j} = (\alpha', j_k, \beta')$, then, if we let $X = nd(Root, \alpha, \alpha')$, we have that $vl(Root, (\alpha, i_k, \beta), (\alpha', j_k, \beta')) = vl(Root, \alpha, \alpha') \cdot X[i_k, j_k].v \cdot vl(X[i_k, j_k].d, \beta, \beta')$.

2 MDD-based aggregation for CTMCs

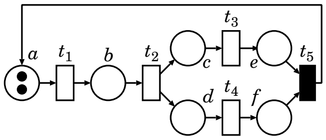
Given an *ergodic* CTMC with state space $\mathcal{S}$, transition rate matrix $\mathbf{R}$, and stationary probability vector $\boldsymbol{\pi}$, we can partition $\mathcal{S}$ into n disjoint classes $\{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ and define an *aggregated* CTMC with state space $\mathcal{S}_{agg} = \{1, \ldots, n\}$ and transition rate matrix $\mathbf{R}_{agg}$ given by $\mathbf{R}_{agg}[i, j] = \sum_{\mathbf{i}\in\mathcal{C}_i} \boldsymbol{\pi}[\mathbf{i}|\mathcal{C}_i] \cdot \sum_{\mathbf{j}\in\mathcal{C}_j} \mathbf{R}[\mathbf{i}, \mathbf{j}]$, where $\boldsymbol{\pi}[\mathbf{i}|\mathcal{C}_i]$ is the stationary conditional probability of state $\mathbf{i}$ given that the CTMC is in class $\mathcal{C}_i$. It is known that this aggregated CTMC is ergodic and its stationary probability vector $\boldsymbol{\pi}_{agg}$ satisfies $\boldsymbol{\pi}[\mathbf{i}|\mathcal{C}_i] = \boldsymbol{\pi}[\mathbf{i}]/\boldsymbol{\pi}_{agg}[i]$, thus $\boldsymbol{\pi}_{agg}[i] = \sum_{\mathbf{i}\in\mathcal{C}_i} \boldsymbol{\pi}[\mathbf{i}]$. However, in general, computing $\mathbf{R}_{agg}$ exactly is as hard as as computing the entire $\boldsymbol{\pi}$.

We then extend [3], defining a general approximate fixpoint iteration that uses L aggregated CTMCs, one for each level of the MDD encoding $\mathcal{S}$. The level–k aggregated CTMC has state space $\mathcal{S}_{agg,k} = \{\langle p, i_k\rangle : p \in \mathcal{L}_k, i_k \in \mathcal{S}_k, \mathcal{B}(p[i_k]) \neq \emptyset\}$, where aggregated state $\langle p, i_k\rangle$ corresponds to the set of states $\mathcal{C}_{\langle p,i_k\rangle} = \mathcal{A}(p) \times \{i_k\} \times \mathcal{B}(p[i_k])$ of the original CTMC. Fig. 6 shows the level-4 state space and the CTMC for our running example.

Let $\Pr\{\mathcal{X}\} = \sum_{\mathbf{i}\in\mathcal{X}} \boldsymbol{\pi}[\mathbf{i}]$ be the probability that the original CTMC is in a state of $\mathcal{X} \subseteq \mathcal{S}$, $\Pr\{p\} = \Pr\{\mathcal{A}(p)\times\mathcal{B}(p)\}$ be the probability of an MDD node p at level k, and $\Pr\{p, \gamma\} = \Pr\{\mathcal{A}(p) \times \{\gamma\} \times \mathcal{B}(nd(p, \gamma))\}$ for a sequence $\gamma = i_k, i_{k-1}, ..., i_h$, where $\Pr\{p, i_k\} = \Pr\left\{\mathcal{C}_{\langle p,i_k\rangle}\right\} = \boldsymbol{\pi}_{agg,k}[\langle p, i_k\rangle]$ is the special case when $|\gamma| = 1$.

Given a node $p \in \mathcal{L}_k$, consider two sequences $\alpha \in \mathcal{A}(p)$ and $\beta \in \mathcal{B}(p[i_k])$, and a state $\mathbf{i} = (\alpha, i_k, \beta) \in \mathcal{C}_{\langle p,i_k\rangle}$. Since $\Pr\{\gamma|p\} = \Pr\{p, \gamma\}/\Pr\{p\}$, we have $\boldsymbol{\pi}[\mathbf{i}|\mathcal{C}_{\langle p,i_k\rangle}] = \Pr\{\alpha, i_k, \beta|p, i_k\} = \Pr\{\alpha|p, i_k\} \Pr\{\beta|\alpha, p, i_k\} = \Pr\{\alpha|p, i_k\} \Pr\{\beta|\alpha, i_k\}$.

* Corresponding author E-mail: ciardo@cs.ucr.edu, Phone: +1 951 827 2281, Fax: +1 851 827 4643



t_1 rate = $2.0 \cdot \#(a)$ t_2 rate = $4.0 \cdot \#(b)$
t_3 rate = $1.0 \cdot \#(c)$ t_4 rate = $3.0 \cdot \#(d)$

Fig. 1 A fork-join queueing network.

$$\mathcal{S}_5=\{f^0e^0,f^0e^1,f^1e^0,f^2e^0,f^0e^2\}=\{0,1,2,3,4\}$$
$$\mathcal{S}_4=\{d^0,d^1,d^2\}=\{0,1,2\}$$
$$\mathcal{S}_3=\{c^0,c^1,c^2\}=\{0,1,2\}$$
$$\mathcal{S}_2=\{b^0,b^1,b^2\}=\{0,1,2\}$$
$$\mathcal{S}_1=\{a^0,a^1,a^2\}=\{0,1,2\}$$

Fig. 2 The local state spaces.

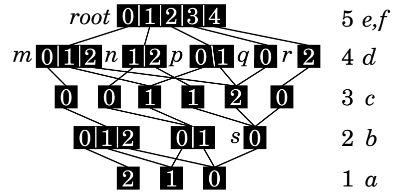


Fig. 3 MDD encoding $\mathcal{S}$.

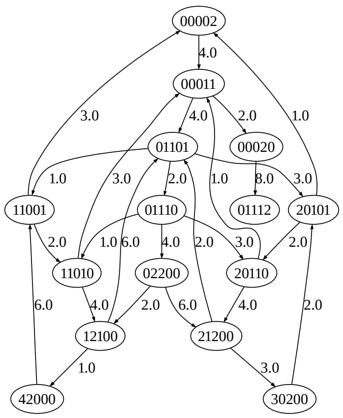


Fig. 4 The CTMC underlying the model.

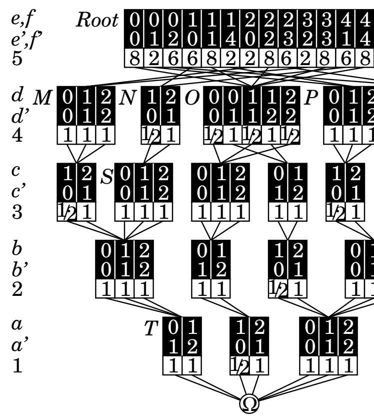
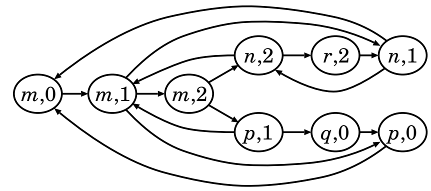


Fig. 5 EV*MDD encoding $\mathbf{R}$.



$$\begin{aligned}\mathcal{S}_{agg,4} &= \{\langle m,0\rangle,\langle m,1\rangle,\langle m,2\rangle,\langle n,1\rangle,\\ &\quad \langle n,2\rangle,\langle p,0\rangle,\langle p,1\rangle,\langle q,0\rangle,\langle r,2\rangle\}\\ \mathcal{A}(p) &= \{2\}\\ \mathcal{B}(p[0]) &= \{101,110\}\\ \mathcal{C}_{\langle p,0\rangle} &= \{2\}\times\{0\}\times\{101,110\}\\ &= \{20101,20110\}\subseteq\mathcal{S}\end{aligned}$$

Fig. 6 Example: level-4 aggregation.

3 EV*MDD–based approximation algorithm for structured CTMCs

Given two aggregated states $\langle p,i_k\rangle$ and $\langle q,j_k\rangle$, and letting $M = nd(Root,\alpha,\alpha')$, we can rewrite $\mathbf{R}_{agg,k}[\langle p,i_k\rangle,\langle q,j_k\rangle]$ as $\sum_{\alpha\in\mathcal{A}(p)} \Pr\{\alpha|p,i_k\}\cdot\sum_{\alpha'\in\mathcal{A}(q)} vl(Root,\alpha,\alpha')\cdot M[i_k,j_k].v\cdot\sum_{\beta\in\mathcal{B}(p[i_k])}\Pr\{\beta|\alpha,i_k\}\cdot\sum_{\beta'\in\mathcal{B}(q[j_k])} vl(M[i_k,j_k].d,\beta,\beta')$, but computing $\Pr\{\alpha|p,i_k\}$ and $\Pr\{\beta|\alpha,i_k\}$ exactly is hard, so we estimate $\Pr\{\alpha|p,i_k\}$ and $\Pr\{\beta|\alpha,i_k\}$ with an assumption (true if product-form holds): $\Pr\{i_k|\alpha\} = \Pr\{i_k|p\}$, for all $L \le k \le 1$, $p\in\mathcal{L}_k$, $i_k\in\mathcal{S}_k$, and $\alpha\in\mathcal{A}(p)$. If this holds, $\Pr\{\alpha|p,i_k\} = \Pr\{\alpha|p\} = \Pr\{p',i_{k+1}|p\}\Pr\{\gamma|p'\}$ and $\Pr\{\beta|\alpha,i_k\} = \Pr\{\beta|p[i_k]\} = \Pr\{i_{k-1}|p[i_k]\}\Pr\{\delta|p[i_k][i_{k-1}]\}$, where $p'\in\mathcal{L}_{k+1}$, $p'[i_{k+1}] = p$, $\gamma\in\mathcal{A}(p')$, $\alpha = (\gamma,i_{k+1})$, and $\beta = (i_{k-1},\delta)\in\mathcal{B}(p[i_k])$ and $\boldsymbol{\pi}[\mathbf{i}] = \Pr\{i_1\,|\,i_L,\ldots,i_2\}\cdot\Pr\{i_L,\ldots,i_2\} = \ldots = \prod_{k=1}^{L}\Pr\{i_k\,|\,p_k\} = \prod_{k=1}^{L}\boldsymbol{\pi}_{agg,k}[\langle p_k,i_k\rangle]/\Pr\{p_k\}$, letting $p_L = root$ and $p_k = nd(i_L,\ldots,i_{k+1})$.

We rewrite $\mathbf{R}_{agg,k}[\langle p,i_k\rangle,\langle q,j_k\rangle]\approx\sum_{M\in\mathcal{M}_k}\mathbf{A}_M[p,q]\cdot M[i_k,j_k].v\cdot\mathbf{B}_{(M[i_k,j_k].d)}[p[i_k],q[j_k]]$, where $\mathbf{A}_M[p,q]$ and $\mathbf{B}_M[p,q]$ describe the contribution to the transition rate from node p to q due to "above" and "below" of node M. For $M\in\mathcal{M}_{k-1}$ and $p,q\in\mathcal{L}_{k-1}$, $\mathbf{A}_M[p,q] = \sum_{\alpha\in\mathcal{A}(p),\alpha'\in\mathcal{A}(q):nd(root,\alpha,\alpha')=M}\Pr\{\alpha\,|\,p\}\cdot vl(Root,\alpha,\alpha')$ is computed with the top–down recurrence $\mathbf{A}_M[p,q] = \sum_{M'\in\mathcal{M}_k,p',q'\in\mathcal{L}_k}\mathbf{A}_{M'}[p',q']\cdot\sum_{i_k,j_k:p=p'[i_k],q=q'[j_k],M=M'[i_k,j_k].d}\Pr\{p',i_k\,|\,p\}\cdot M'[i_k,j_k].v$, with base case $\mathbf{A}_{Root.d}[root,root] = Root.v$. Analogously, $\mathbf{B}_M[p,q] = \sum_{\beta\in\mathcal{B}(p),\beta'\in\mathcal{B}(q)}\Pr\{\beta\,|\,p\}\cdot vl(M,\beta,\beta')$ is computed with the bottom–up recurrence $\mathbf{B}_M[p,q] = \sum_{i_k,j_k}\Pr\{i_k\,|\,p\}\cdot M[i_k,j_k].v\cdot\mathbf{B}_{M[i_k,j_k].d}\,[p[i_k],q[j_k]]$, with base case $\mathbf{B}_\Omega[\mathbf{1},\mathbf{1}] = 1$.

To build the $\mathbf{A}$ matrices, we need $\Pr\{p',i_k\,|\,p\} = \boldsymbol{\pi}_{agg,k}[\langle p',i_k\rangle]/\Pr\{p\}$, where $p = p'[i_k]$ and the probability of node $p\in\mathcal{L}_l$ is $\Pr\{p\} = \sum_{i_l:p[i_l]\neq\mathbf{0}}\boldsymbol{\pi}_{agg,l}[\langle p,i_l\rangle]$. Similarly, to build the $\mathbf{B}$ matrices, we need $\Pr\{i_k\,|\,p\} = \boldsymbol{\pi}_{agg,k}[\langle p,i_k\rangle]/\Pr\{p\}$.

Then, for $M\in\mathcal{M}_k$, $\mathbf{A}_M$ is built based on $\boldsymbol{\pi}_{agg,l}$ for levels $L\ge l\ge k$ and, for $M\in\mathcal{M}_k$, $\mathbf{B}_M$ is built based on $\boldsymbol{\pi}_{agg,l}$ for levels $k\ge l\ge 1$. Thus, the level–k aggregated CTMC depends on the solutions of all other aggregated CTMCs and, to break this cyclic dependency, we use a fixpoint iteration starting from an initial guess.

Once $\boldsymbol{\pi}$ is known, we may seek the expectation of a a reward rate function $r:\mathcal{S}\to\mathbb{R}$, $m = \sum_{\mathbf{i}\in\mathcal{S}} r(\mathbf{i})\cdot\boldsymbol{\pi}[\mathbf{i}]$. If we approximate each $\boldsymbol{\pi}[\mathbf{i}]$ with $\boldsymbol{\pi}[\mathbf{i}]\approx\prod_{k=1}^{L}\boldsymbol{\pi}_{agg,k}[\langle p_k,i_k\rangle]/\Pr\{p_k\}$, where $p_L,p_{L-1},\ldots,p_2,p_1$ are the MDD nodes along the path corresponding to $\mathbf{i}$ (thus $p_L = root$), computing m requires $O(L\cdot|\mathcal{S}|)$ time. However, if r is the product of L "local" functions, $r(\mathbf{i}) = \prod_{k=1}^{L} r_k(i_k)$, we can compute m recursively, by letting $m(p) = \sum_{i_k:p[i_k]\neq 0} r_k(i_k)\cdot\Pr\{i_k\,|\,p\}\cdot m(p[i_k])$, in time proportional to the number of MDD edges.

References

[1] G. Ciardo and A. J. Yu, Saturation-based symbolic reachability analysis using conjunctive and disjunctive partitioning, in: Proc. CHARME, (Springer-Verlag, Saarbrücken, Germany, October 2005), pp. 146–161.

[2] A. S. Miner, Efficient solution of GSPNs using canonical matrix diagrams, in: Proc. PNPM'01 (IEEE Comp. Soc. Press, Aachen, Germany, September 2001), pp. 101–110.

[3] A. S. Miner, G. Ciardo, and S. Donatelli, Using the exact state space of a Markov model to compute approximate stationary measures, in: Proc. ACM SIGMETRICS (ACM Press, Santa Clara, CA, USA, June 2000), pp. 207–216.