Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 764819, 6 pages
http://dx.doi.org/10.1155/2014/764819

Research Article

A Multistep Extending Truncation Method towards Model Construction of Infinite-State Markov Chains

Kemin Wang, Yongbin Wang, Zhengtao Jiang, and Wenlong Fu

School of Computer Science, Communication University of China, Beijing 100024, China

Correspondence should be addressed to Kemin Wang; bbi-wkm@cuc.edu.cn

Received 9 October 2013; Accepted 14 December 2013; Published 12 January 2014

Academic Editor: Shuping He

The model checking of Infinite-State Continuous Time Markov Chains will inevitably encounter the state explosion problem when constructing the CTMCs model; our method is to get a truncated model of the infinite one; to get a sufficient truncated model to meet the model checking of Continuous Stochastic Logic based system properties, we propose a multistep extending advanced truncation method towards model construction of CTMCs and implement it in the INFAMY model checker; the experiment results show that our method is effective.

1. Introduction

Continuous Time Markov Chains (CTMCs) have been used in various areas of research as a formalism; so far, the model checking of CTMCs has been a hot research topic in computer science research communities. Some algorithms and implementations have been shown in several papers and tools [1–7]. However, our research aims to the model checking problem of the Infinite-State CTMCs, which means that the states of the CTMCs in our interest can be infinite. The papers [8–11] and the tool INFAMY [7] are with the same interest, which are also our research basis. Due to the explosion of states of CTMCs, our approach is based on a truncated CTMC model, which is determined by the exploration on the fly [9, 10], which means that the depth of model is computed dynamically by the exploration of states.

The truncation process is involved with reachability analysis [9, 10]; that is, the transient probability is computed with the exploration of the model. The transient probability is carried out by uniformization method [10]. Together with the transient analysis when constructing the model, computation is very heavy, so to get a sufficient truncated model to meet the requirement of related Continuous Stochastic Logic (CSL) property or certain precision as fast as possible is our destination. We introduce a multistep extending advanced truncation method to meet this end, and the experimental results show our method is effective.

The main contents of the paper are organized as follows. Section 2 introduces the truncation based reachability analysis; Section 3 introduces an advanced truncation algorithm and do experiments. Section 4 proposes some multistep extending solutions and experiments. Section 5 talks about our result and next job.

2. Truncation Based Reachability Analysis

2.1. Finite Truncations. Let $\mathscr{C} = (S, \mathbf{R}, L)$ be a CTMC, where S is a countable set of states, $\mathbf{R} : (S \times S \rightarrow \mathbb{R}_{\geq 0})$ is the rate matrix, and $L : S \rightarrow 2^{AP}$ is a labeling function.

First, we introduce some paths and probabilistic measures from [9]. A (timed) infinite path is an infinite sequence $\sigma = s_1 t_1 s_2 t_2 \ldots$ satisfying $\mathbf{R}(s_i, s_{i+1}) > 0$, and $t_i \in \mathbb{R}_{\geq 0}$ for all $i = 1, 2, \ldots$. For the path σ and $i \in N$, let $\sigma[i] = s_i$ denote the $(i+1)$th state, and let $\delta(\sigma, i) = t_i$ denote the time spent in s_i. For $t_i \in \mathbb{R}_{\geq 0}$, let $\sigma@t$ denote $\sigma[i]$ such that i is the smallest index with $t \leq \sum_{j=0}^{i} t_j$. For $\mathscr{C}$, let $\text{path}_{\infty}^{\mathscr{C}}$ denote the set of all paths, and let $\text{path}_{\infty}^{\mathscr{C}}(s)$ denote the set of all paths starting from s. For state $s \in S$, a probability measure, denoted by $\Pr_s^{\mathscr{C}}$, on the set $\Pr_s^{\mathscr{C}}(s)$ can be defined. A finite path is a finite sequence $\sigma = s_1 t_1 s_2 t_2 \ldots s_k$ for $k > 0$ satisfying $\mathbf{R}(s_i, s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{\geq 0}$ for $i = 1, 2, \ldots, k-1$. Let $\text{len}(\sigma) = k - 1$ denote the length of the path, $\text{first}(\sigma) = s_1$ denote the first state, and $\text{last}(\sigma) = s_k$ denote the last state of the path. Let $\text{path}_{f}^{\mathscr{C}}$ denote the set of all finite paths. We omit the superscript $\mathscr{C}$ if it is clear from context.

Next, we introduce the notion of depth. Let S_0 be a finite subset of S with depth 0; that is, $d(s) = 0$ for $s \in S_0$. For now, one may think of S_0 as being equal to the support of the initial distribution μ. However, S_0 can be an arbitrary finite set. This will allow us not only to deal with the initial distribution, but also to compute truncation depths for nested CSL formulas. The depth of state s corresponds to the minimal distance from the set S_0.

Definition 1. For $\mathscr{C}$ and $S_0 \subset S$, the depth function $d : S \to \mathbb{N}$ is defined by $d_{S_0}(s) = \min\{\text{len}(\sigma) \mid \sigma \in \text{path}_f \wedge \text{first}(\sigma) \in S_0 \wedge \text{last}(\sigma) = s\}$.

Observe that $d_{S_0}(s) = 0$ for all $s \in S_0$. The subscript is omitted if S_0 is clear from the context. Intuitively, $d(s)$ corresponds to the minimal length of any finite path starting from S_0 and ending in s.

We consider a partition of the state space $S = \bigcup_{i\in\mathbb{N}} S_i$, where $S_i := \{s \in S \mid d(s) = i\}$ is the set of states with depth i. We say that the set S_i is the layer with depth i and call its elements layer-i states. Assume that $\perp\in S$ is a special state not in S and furthermore $\perp$ is also an atomic proposition not in AP. For $k \in \mathbb{N}$, let $S_{>k} \subset S$ denote the set of states with depth greater than k; that is, $S_{>k} = \{s \mid d(s) > k\}$.

Further, we write $S_{\leq k} = S \setminus S_{>k}$. We define the k-truncated CTMC as follows.

Definition 2. Let $\mathscr{C} = (S, \mathbf{R}, L)$ be a CTMC and let S_0 be the layer-zero states. For $k \in \mathbb{N}$, we define the k-truncated CTMC of $\mathscr{C}$ by $\mathscr{C}|_k = (S|_k, \mathbf{R}_k, L_k)$, where the states are $S|_k = S_{\leq k} \cup \{\perp\}$. The labeling function $L_k : S|_k \to AP \cup \{\perp\}$ is defined by $L_k(s) = L(s)$ if $s \in S_{\leq k}$, and $L_k(\perp) = \{\perp\}$. The rate matrix is defined by the following: $\mathbf{R}_k(s, s')$ equals $S(s, s')$ if $s, s' \in S_{\leq k}$, equals $\mathbf{R}(s, S_{>k}) = \sum_{t\in S_{>k}} \mathbf{R}(s, t)$ if $s \in S_k, s' = \perp$, and equals 0 otherwise.

The k-truncation of an infinite CTMC is illustrated in Figure 1. Intuitively, the transition matrix is restricted to the truncated state space $S|_k$, and $\perp$ is the distinguished absorbing state, which, by construction, is only reachable from states with depth k. In state $\perp$ only the atomic proposition $\perp$ holds, which indicates that the system is in state $\perp$. Since we consider finitely branching CTMCs, not surprisingly, the k-truncated CTMC $\mathscr{C}|_k$ is always finite. The absorbing state $\perp$ has been introduced to abstract $S_{>k}$. We assume that $\perp\notin \text{Sat}(\Phi)$ for any state formula Φ. We consider the probability of reaching the absorbing state $\perp$ in the k-truncated CTMC $\mathscr{C}|_k$, that is, $\vec{\pi}^{\mathscr{C}|_k}(s, t)(\perp)$. For mere notational convenience, we extend $\vec{\pi}^{\mathscr{C}|_k}$ to states of $\mathscr{C}$ with depth higher than $k : \vec{\pi}^{\mathscr{C}|_k}(s, t)(s') = 0$ for all $s' \in S$ with $d(s') > k$. For a fixed k, we define the forward rate $\text{fr}_k(s)$ of a state $s \in S_{\leq k}$ within $\mathscr{C}|_k$. For $s \in S_{\leq k} \setminus S_k$, it is the sum of the rates that go into the next layer $\text{fr}_k(s) = \mathbf{R}(s, S_{d(s)+1})$, and, for $s \in S_k$, it is the sum $\text{fr}_k(s) = \mathbf{R}(s, S_{>k})$ of the rates entering states in S_k.

2.2. The Logic CSL. The logic we consider is CSL without steady-state operator and unbounded until operator [10]. Let $I = [t, t']$ be an interval with $t, t' \in \mathbb{R}_{\geq 0}$ and $t \leq t'$. Let $p \in [0, 1]$ and $\trianglelefteq\in \{\leq, <, >, \geq\}$. The syntax of state formula Φ and path formulas ϕ is

$$\Phi = a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathscr{P}_{\trianglelefteq p}(\phi), \qquad \phi = \mathscr{X}^I\ \Phi \mid \Phi\ \mathscr{U}^I\ \Phi. \tag{1}$$

The semantics of the state formulas and path formula are precisely defined in [10]. In this paper, the state formula like the kind of $\mathscr{P}_{\trianglelefteq p}(\phi)$ will be more focused on; it means that starting some state, the probability of the system that satisfies a path formula ϕ would meet the rational relation $\trianglelefteq\ p$ or not, so the first solution to this problem is to compute this probability, so this is reduced to transient analysis, which is our main interest, to improve the efficiency of transient analysis for the whole model while extending the border states layer by layer.

2.3. Transient Analysis. Transient analysis means that, starting at state s, the transient probability vector at time t, it can be denoted as $\vec{\pi}(s, t)$. If $t = 0$, we have $\vec{\pi}(s, 0)(s') = 1$ if $s = s'$ and 0 otherwise. The uniformization method based solution of $\vec{\pi}(s, t)$ was given at [10]. Let $t > 0$ and $\vec{\pi}(s, t) = \sum_{i=0}^{\infty} \varphi(i, qt)\vec{\pi}(s, i)$, where q is the uniformization rate satisfying $q \geq \sup_{s\in S}\mathbf{R}(s, S)$ and $\varphi(i, qt) = e^{-qt}((qt)^i/i!)$ denotes the ith Poisson probability with parameter qt and vector $\vec{\pi}(s, i)$ is transient probability of the uniformized DTMC uni($\mathscr{C}$) at step i.

The truncated model in our method is given by dynamically exploring from the initial state(s); once we want to add a layer on it, we need to compute the reachability probability from the initial(s) to the current layer states which we want to add. As shown in Figure 1, s_0 is the initial states set, and each of $s_1, s_2, \ldots, s_k$ is explored dynamically, so we get a k-depth truncated model of the infinite one. $\perp$ is the absorbing state, which is the border states set, for which the sum of the reachability probabilities from the initial state(s) is less than ε; ε is the precision of the result, which can be 10^{-6}, 10^{-9}, or 10^{-12}, and so forth, which can be set under INFAMY model checker as a circumstance variable for some certain need.

The reachability probability is carried out by the uniformization method to the CTMC, which is a relatively fast

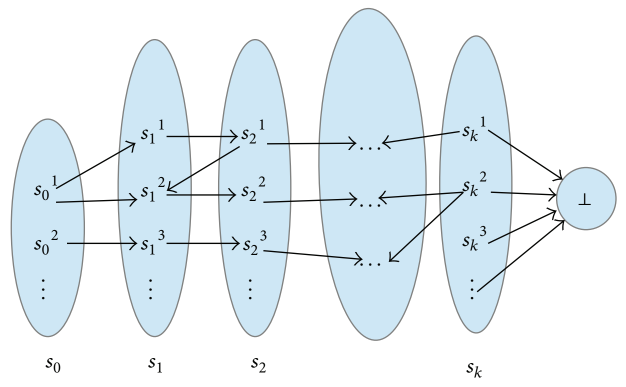

Figure 1: A k-depth truncated CTMC.

```
(1) procedure TRANSIENTTRUNCADVANCED($\mathscr{C}, S_0, t, \varepsilon$)
(2)   $I = S_0$
(3)   $B = S_0$
(4)   compute $\mathscr{P}_I\left(\Diamond^{[0,t]}B\right)$
(5)   while $\max\left(\mathscr{P}_I\left(\Diamond^{[0,t]}B\right)\right) \geq \varepsilon$ do
(6)     $s' = s \mid \mathscr{P}_s\left(\Diamond^{[0,t]}B\right) = \max\left(\mathscr{P}_I\left(\Diamond^{[0,t]}B\right)\right)$
(7)     compute $\mathscr{P}_{s'}\left(\Diamond^{[0,t]}s'' \mid s'' \in B\right)$
(8)     while $\sum_{s''\in B} \mathscr{P}_{s'}\left(\Diamond^{[0,t]}s'' \geq \varepsilon\right)$ do
(9)       $E = B \setminus \{s''_0, s''_1, s''_2, \ldots, s''_n\}$ such that
(10)      $\mathscr{P}_{s'}\left(\Diamond^{[0,t]}s''_0\right) \leq \mathscr{P}_{s'}\left(\Diamond^{[0,t]}s''_1\right) \leq \mathscr{P}_{s'}\left(\Diamond^{[0,t]}s''_2\right) \leq \cdots \leq \mathscr{P}_{s'}\left(\Diamond^{[0,t]}s''_n\right)$ and
          $\sum \mathscr{P}_{s'}\left(\Diamond^{[0,t]}s''_i\right) < \varepsilon$
(11)      $B = \{successors(s) \mid s \in E\}$
(12)      compute $\mathscr{P}_{s'}\left(\Diamond^{[0,t]}s'' \mid s'' \in B\right)$
(13)    end while
(14)    compute $\mathscr{P}_I\left(\Diamond^{[0,t]}B\right)$
(15)  end while
(16) end procedure
```

Algorithm 1: An advanced truncation algorithm towards construction of CTMC.

method to get the transient probability at a certain bound time t at some state, which is denoted as $\mathscr{P}(\Diamond^{[0,t]}s)$. However, if the state explosion situation is very serious, the time to construct the model layer by layer would be of much cost. So we introduce an advanced truncation algorithm to explore the states, this method can improve the efficiency of the model constructing and model checking, and then at Section 4, we further propose some multistep extending solutions, which are implemented based on INFAMY model checker, the experiment results show that these solutions can help to improve the efficiency.

3. An Advanced Truncation Algorithm and Experiments

The truncation process is implemented by extending the states layer by layer from the initial states, and all the new states need to be transient analyzed and then to be extended further no matter how small the probability is. For that the precision of the result is under some certain value, so some states with relatively nearly no contribution to the result can be omitted when extending; thus, an advanced truncation algorithm is introduced. It is different from the finite state projection (FSP) [9, 10] and layered chain and uniform chain method [9, 10]. The algorithm is shown in Algorithm 1.

The algorithm aims to stop the extending of less important (small probability) states and proceed to the extending of much important (large probability) states; the less or more is determined by the state reduction policy; as shown in Figure 2, line 7, $\mathscr{P}_{s'}(\Diamond^{[0,t]}s''_0) \leq \mathscr{P}_{s'}(\Diamond^{[0,t]}s''_1) \leq \mathscr{P}_{s'}(\Diamond^{[0,t]}s''_2) \leq \cdots \leq \mathscr{P}_{s'}(\Diamond^{[0,t]}s''_n)$ and $\sum \mathscr{P}_{s'}(\Diamond^{[0,t]}s''_i) < \varepsilon$. The policy means that the states in the border states set, which has been sorted upward, $s''_0, s''_1, s''_2, \ldots, s''_n$, will be excluded from the extending states set, for the sum of them is just exactly less than the precision. So with this policy, the number of states to be transient analyzed will be smaller to the FSP and forward-layered based model.

We consider the dependability of a fault-tolerant workstation cluster which is directly taken from case studies of [7]. Figure 2 depicts a dependable cluster of workstations. The cluster consists of two subclusters, which, in turn, contain N workstations connected via a central switch. The two switches are connected via a backbone. Each component of the system can break down and is then fixed by a single repair unit responsible for the entire system. Hereby, the quality of service (QoS) constraint minimum requires at least k ($k < N$) workstations to be operational, where $k = \lfloor (3/4)N \rfloor$. Workstations have to be connected via switches. If in each subcluster the number of operational workstations is smaller than k, the backbone is required to be operational to provide the required service. We consider the property.

$\mathscr{P} =?\Diamond[0,t]\neg$Minimum. This probability means that the QoS drops below minimum quality within t time unit.

For the property, we compare PRISM [6], FSP method and layered method of INFAMY. The results are given in Tables 1, 2, and 3. Because the resulting probabilities are very small in some cases, we use a precision of 10^{-6} here, for the computation of the truncation point. Results for INFAMY are given for the layered chain, FSP, and advanced configurations, respectively. The uniform chain configuration is omitted, as it is always dominated by the layered chain configuration. PRISM implements three different engines: a sparse-matrix and two symbolic engines. We used the sparse-matrix engine

Table 1: Experiments of forward-layered method of INFAMY.

Time bound t	Forward-layered			
	Depth	Time (s)	n	Prob
5.0	316	450	1668363	$[0.9E-06, 1.50E-06]$
10.0	517	661.5	2373652	$[3.2E-06, 3.8E-06]$
20.0	517	1293	2373652	$[8.8E-06, 9.4E-06]$
30.0	517	1453	2373652	$[1.45E-05, 1.51E-05]$
50.0	517	2096	2373652	$[2.61E-05, 2.67E-05]$

Table 2: Experiments of advanced method of INFAMY.

Time bound t	Advanced			
	Depth	Time (s)	n	Prob
5.0	18	6.6	4071	$[0.9E-06, 1.50E-06]$
10.0	25	23.3	7118	$[3.2E-06, 3.9E-06]$
20.0	38	105.2	14434	$[8.8E-06, 9.5E-06]$
30.0	50	271.6	22426	$[1.45E-05, 1.52E-05]$
50.0	69	1491.5	38448	$[2.61E-05, 2.68E-05]$

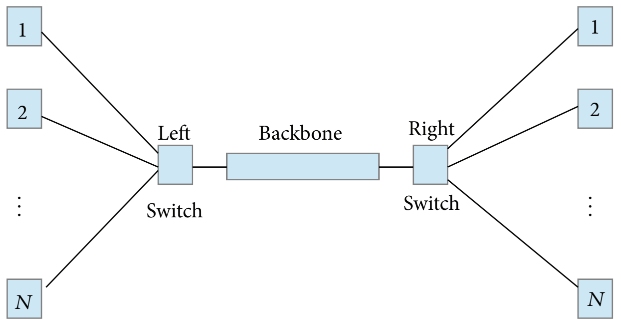

Figure 2: Dependable clusters of workstations.

as it was the fastest one. The results are shown in Tables 1, 2, and 3.

The experiment conditions are shown as follows.

Host Machine: it includes Mac Book, OS: Mac OS X 10.6.8, Processor: 2.2 GHz Intel Core 2 Duo T7500, and Storage: 2 GB 667 MHz DDR3 SDRAM.

Guest Machine: it includes Virtual Machine Software: VirtualBox 4.1.14 r77440 for mac; Virtual OS: Linux ubuntu12.04 LTS 32Bit; Processor: 2.2 GHz Intel Core 2 Duo T7500; Storage: 512 MB

From Tables 1, 2, and 3, we can see that, for $t \leq 20$, FSP based INFAMY is faster, but for $t \leq 30$, $t \leq 50$, INFAMY model checker needs more time; this is because the transient analysis when constructing the model needs more computing. This is also the result of [11]. For the advanced method of the context, as in Tables 1, 2, and 3, comparing advanced with FSP, we can get that, under advanced based method, the depth of model is much deeper, and the states number is smaller; for the time costing, for $t \leq 20$, $t \leq 30$, the costing is less reduced, but for $t \leq 50$, the costing is greater; this result is reasonable, for the exploration policy is essentially undeterministically efficient for different models; for the current case, when $t \leq 50$, the current states number is very large, even with the reduction policy, with no contribution for it any more. We need to take other techniques to tackle this situation. Thus, we propose a multistep extending solution. See Section 4.

4. Multistep Extending Solutions and Experiments

The mutistep extending solution aims to reduce the extending of less important (small probability) states and enhance the extending of more important (large probability) states; for the latter states, we can, for example, extend two or more steps per extending, and for the former states, we can, for example, extend one step per extending, as we know that transient probabilities will be computed once again before the states were added to the border states, so, if we extend two or more steps, transient analysis at the intermediate states will be omitted; thus, time on the model construction will be reduced; then we can make the model much faster to converge to the absorbing state.

We continue with the upper case study. As shown in Algorithm 1, Line 11, $B = \{successors(s) \mid s \in E\}$ is to extend the model from set E and get the border states set B. This means that the advanced method in the upper section is extending one step per extending. Now, we design some multistep extending solutions:

(1) two-step extending solution, that is, $B = \{successors(successors(s)) \mid s \in E\}$;

(2) three-step extending solution, that is, $B = \{successors(successors(successors(s))) \mid s \in E\}$;

(3) synthesis solution 1: a synthesis extending solution separates set E to three parts, as set E is a sorted set, which is sorted upward by the probabilities, so we can separate E as E_1, E_2, and E_3, and the sizes of each part are the same. Then we can get

Table 3: Experiments of PRISM and FSP method of INFAMY.

Time bound t	PRISM (sparse engine)		FSP			
	Time (s) (construct/M.C.ing)	Prob	Depth	Time (s)	n	Prob
5.0	1.868/46.5	$1.16E-06$	16	5.7	4405	$[0.9E-06, 1.70E-06]$
10.0	1.8/77.265	$3.51E-06$	22	20.7	8425	$[3.2E-06, 4.0E-06]$
20.0	1.8/127.072	$9.06E-06$	33	113.9	19161	$[8.8E-06, 9.7E-06]$
30.0	1.8/179.675	$1.48E-05$	42	353.6	31185	$[1.45E-05, 1.56E-05]$
50.0	1.8/274.593	$2.64E-05$	57	1219.7	57705	$[2.61E-05, 2.73E-05]$

Table 4: Experiment data under different extending solutions.

Time bound t	Advanced (one step per extending)			Two-step per extending			Three-step per extending		
	Depth	Time (s)	n	Depth	Time (s)	n	Depth	Time (s)	n
5.0	18	6.6	4071	20	8.9	4756	33	15.0	7895
10.0	25	23.3	7118	29	26.5	8027	38	41.3	11849
20.0	38	105.2	14434	42	115.3	16315	52	174.0	20816
30.0	50	271.6	22426	54	327.6	25308	64	456.0	30785
50.0	69	1491.5	38448	72	1006.5	43824	82	1572.6	50942
t	Synthesis solution 1			Synthesis solution 2			Solution D		
5.0	35	11.9	6539	20	9.6	4564	20	8.6	4459
10.0	42	35.4	9772	26	29.7	7646	26	24.9	7383
20.0	53	131.2	17777	41	111.7	15621	39	91.1	15170
30.0	61	367.4	26362	53	279.7	24239	50	250.9	23063
50.0	74	1349.6	45170	71	989.1	42703	71	791	40403

(i) $B_1 = \{successors(s) \mid s \in E_1\}$;

(ii) $B_2 = \{successors(successors(s)) \mid s \in E_2\}$;

(iii) $B_3 = \{successors(successors(successors(s))) \mid s \in E_3\}$;

(iv) $B = B_1 \cup B_2 \cup B_3$;

(4) synthesis solution 2 separates set E to two parts, so we can separate E to E_1 and E_2, and the sizes of the two parts are the same. Then we can get

(i) $B_1 = \{successors(s) \mid s \in E_1\}$;

(ii) $B_2 = \{successors(successors(s)) \mid s \in E_2\}$;

(iii) $B = B_1 \cup B_2$.

The experiments data are shown in Table 4. From the data in Table 4, we can see that under different extending solutions, the times needed are different, synthesis solution 2 performs better, when $t \leq 50$, and the time (including model constructing and model checking) has been reduced to 1000 s. This means that for this case, the step per extending should be relatively small; for synthesis solution 2, half small states extend one step per-extending, and other half large states extend two steps per extending. And for this, we continue to propose a solution to revise synthesis solution 2, named Solution D the policy is as follows: For the states $e_0, e_1, \ldots, e_n$ in set E, the gap between the largest and smallest probabilities: $\mathrm{ProbGap} = \mathrm{Prob}(E_n) - \mathrm{Prob}(E_0)$. We can separate set E to two parts E_1 and E_2, such that.

(i) $E_1 = \{e_0, e_1, \ldots, e_i \mid e_i \leq \mathrm{Prob}(e_0 + \mathrm{ProbGap}/2)$ and $e_{i+1} > \mathrm{Prob}(e_0 + \mathrm{ProbGap}/2)\}$;

(ii) $E_2 = E \setminus E_1$.

And then we can get that

(i) $B_1 = \{successors(s) \mid s \in E_1\}$;

(ii) $B_2 = \{successors(successors(s)) \mid s \in E_2\}$;

(iii) $B = B_1 \cup B_2$.

The experiment results are as shown in Table 4. From Table 4, we can see that current solution D performs better on this case.

5. Conclusion

The multistep extending advanced truncation method can improve the efficiency of model construction of Infinite-State CTMCs; this is because the transient probabilities of states which have been jumped have not been computed, so to some extent this method is effective; however, which solution performs better needs to be experimented; there is no general solution that fits well for all cases. The efficiency is determined by the iterations when computing the transient probability. Less iteration is more efficient. However, this approach is essentially a linear approach to improve the efficiency; when the outsider state gets explosion, this approach will be less effective; just as in our case study, this approach can be used effectively to improve the model checking efficiency at a relatively small time bound t. For future work, we need to consider other techniques to tackle the state explosion problem on model checking of CTMCs. And other works like [12–15] can also be considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Lijun Zhang and Ernst Moritz Hahn for their original work on INFAMY model checker and the free contribution of the tool codes; Definitions 1 and 2 are directly taken from [10]. This paper is partially funded by the Communication University of China, 2013 Planned Research Program, and Research Funding Foundation of School of Computer Science, and the scholarship from China Scholarship Council for the Excellent University Teachers Study Abroad Program and supported by the National Natural Science Foundation of China (61103199), Beijing Municipal Natural Science Foundation (4112052); Supported by Engineering Program Project of CUC (3132013XNG1326, 3132013XNG1321), and the National Key Science and Technology Pillar Programs of China (2012BAH51F02, 2012BAH38F05, and 2013BAH66F02).

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, *Verifying Continuous Time Markov Chains*, pp. 269–276, Springer, 1996.

[2] C. Baier, J. P. Katoen, and H. Hermanns, "Approximative symbolic model checking of continuous-time markov chains," in *CONCUR '99 Concurrency Theory*, vol. 1664 of *Lecture Notes in Computer Science*, pp. 146–161, Springer, 1999.

[3] H. Hermanns, J. Meyer-Kayser, and M. Siegle, "Multi terminal binary decision diagrams to represent and analyse continuous time markov chains," in *Proceedings of the 3rd International Workshop on the Numerical Solution of Markov Chains*, pp. 188–207, Citeseer, 1999.

[4] J. P. Katoen, M. Kwiatkowska, G. Norman, and D. Parker, *Faster and Symbolic CTMC Model Checking*, Springer, 2001.

[5] C. Baier, B. Haverkort, H. Hermanns, and J. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 524–541, 2003.

[6] Prism model checker, http://www.prismmodelchecker.org/.

[7] Infamy model checker, http://depend.cs.uni-sb.de/tools/infamy/.

[8] A. Remke, B. R. Haverkort, and L. Cloth, "Model checking infinite-state Markov chains," in *Proceedings of the 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '05)*, vol. 3440 of *Lecture Notes in Computer Science*, pp. 237–252, Springer, April 2005.

[9] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang, "Time-bounded model checking of infinite-state continuous-time markov chains," *Fundamenta Informaticae*, vol. 95, no. 1, pp. 129–155, 2009.

[10] L. Zhang, H. Hermanns, E. M. Hahn, and B. Wachter, "Time-bounded model checking of infinite-state continuous-time markov chains," in *Proceedings of the 8th International Conference on Application of Concurrency to System Design (ACSD '08)*, pp. 98–107, June 2008.

[11] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang, "Infamy: an infinite-state markov model checker," in *Computer Aided Verification*, Springer, pp. 641–647, 2009.

[12] S. He and F. Liu, "Robust stabilization of stochastic Markovian jumping systems via proportional-integral control," *Signal Processing*, vol. 91, no. 11, pp. 2478–2486, 2011.

[13] S. He and F. Liu, "Robust $l_2 - l_\infty$ filtering of time-delay jump systems with respect to the finite-time interval," *Mathematical Problems in Engineering*, vol. 2011, Article ID 839648, 17 pages, 2011.

[14] S. He, Z. Ding, and F. Liu, "Output regulation of a class of continuous-time markovian jumping systems," *Signal Processing*, vol. 93, no. 2, pp. 411–419, 2013.

[15] H. Shen, S. Xu, J. Lu, and J. Zhou, "Passivity-based control for uncertain stochastic jumping systems with mode-dependent round-trip time delays," *Journal of the Franklin Institute*, vol. 349, no. 5, pp. 1665–1680, 2012.