STUDIA INFORMATICA 2015
Volume 36 Number 2 (120)

Maciej Rafał BURAK
West Pomeranian University of Technology, Applied Informatics

INHOMOGENEOUS CTMC MODEL OF A CALL CENTER WITH BALKING AND ABANDONMENT

Summary. This paper considers a nonstationary multiserver queuing model with abandonment and balking for inbound call centers. We present a continuous time Markov chain (CTMC) model which captures the important characteristics of an inbound call center and obtain a numerical solution for its transient state probabilities using uniformization method with steady-state detection.

Keywords: call center, transient, Markov processes, numerical methods, uniformization, abandonment, balking

MODELOWANIE CALL CENTER JAKO SYSTEMU KOLEJKOWEGO Z WYKORZYSTANIEM ŁAŃCUCHÓW MARKOWA Z NIEJEDNORODNYM CZASEM CIĄGŁYM

Streszczenie. Artykuł opisuje zastosowanie CTMC do modelowania Call Center z klientami o ograniczonej cierpliwości.

Słowa kluczowe: *call center*, metody numeryczne, uniformizacja, łańcuchy Markowa

1. Introduction

The problem of managing operations of a telephone call center in an efficient way has a long history in the area of operational research and is a topic of current research in various disciplines (see e.g. [1] or [2] for extensive overviews). From the modeling point of view they can be viewed as queuing systems.

Such a queuing model can be described by a corresponding continuous time Markov chain (CTMC) whose steady-state distribution can be easily determined, either analytically –

with the Erlang-C formula for the simplest M/M/n model or with the Erlang-A formula for its version augmented with exponential patience time as proposed in [3] or numerically for more complicated models (as in [4] or recently [5]). However, as real call centers are time inhomogenous, with varying arrival rates and changing number of servers, scheduled to meet the forecasted demand and in order to provide break time, stationary models cannot be applied directly. It is, therefore, common to use approximations, assuming the system being pointwise stationary. Examples of such well established methods can be found e.g. in [1], [3] or in [6]. Unfortunately, stationary approximations are in many cases not adequate. For example, [4] compared them with simulations based on real inbound call center data, with the conclusion that due to the nonstationarity only some of the performance measures can be estimated with satisfactory accuracy. Ingolfsson in [7] compared them with an inherently transient model and found their results significantly inaccurate or even entirely unreliable. Despite this, their widespread use is commonly justified by simple implementation and low computational costs.

Many authors proposed to use simulation, which can achieve any desired accuracy. However, in order to achieve acceptable precision, very long computational times are needed, which makes it often impracticable for common applications like schedule planning.

An alternative approach, which is very effective in terms of the accuracy of the model, is to analyze transient CTMC using numerical methods, solving effectively their corresponding system of *ordinary differential equations* (ODE's) as proposed in [8], [9] or by the author in [10].

Other, less computationally intensive, analytical methods that can approximate such nonstationary systems more accurately than stationary models are *closure approximations* and *fluid and diffusion approximations*, discussed e.g. in [3], [6], [11] and [12] or, for the direct comparison of some examples of such methods with the numerical methods and stationary approximations, in [8].

Although there is a number of papers dealing with the phenomena of customer balking and abandonment in multiserver queues (e.g. [3], [13-15] or recently [5]), they concentrate on stationary models or approximations. To the best of our knowledge, an inherently transient CTMC model dealing with both balking and abandonment of a call center, has never been investigated.

The main objective of this work is to model such non-stationary systems, using transient analysis of corresponding CTMC, in a reliable and precise way, with computational efficiency enabling its use for practical applications – in particular, as a much more accurate replacement to the Erlang-C and Erlang-A formulas, used by practitioners for quantitative call center management.

In this paper we model an inbound telephone call center with balking and abandonment, i.e. the customer may not stay in the queue once realizing he is put on hold, or abandon the queue if the waiting time is too long, extending the nonstationary M/M/n queuing model analyzed by the author in [10].

The paper is structured as follows. In the next section the model and the basic notation are introduced. Section 3 reviews the proposed multi-step uniformization algorithm with steady-state detection and section 4 presents the results of numerical experiments. The paper ends with a summary of results, conclusions and proposals for future research.

2. Model

We propose a following model of a Call Centre: the analyzed period is finite (e.g. one working day) with the system starting empty. The state variable $X(t)$ represents total number of service requests (served/waiting calls) in the system at time t. The size $n(t)$ of the system, which represents the number of non empty possible states, is finite, equal to $s(t)$ = number of identical servers (agents) plus $q(t)$ equal capacity of the queue, with corresponding discrete state space $\varphi(t) = \{0,\ldots, n(t)\}$, $|\varphi(t)| = 1 + s(t) + q(t)$. Customers arrive according to an inhomogenous Poisson process with rate $\lambda(t)$, the service time is i.i.d. exponentially distributed with rate $\mu(t)$. The load $\rho(t) = \lambda(t) / s(t)\,\mu(t)$ can be bigger than 1.

Service requests that are not served immediately can leave the system (hang up or balk) with probability $1 - \gamma$, otherwise, after joining the queue, they abandon after reaching their *patience time*. The patience times are independent and identically exponentially distributed with mean $1 / \eta$. Queued requests are FCFS served. All of this is modeled via the state transition rates of a CTMC which is described by *infinitesimal generator matrix* $Q(t) : n(t) + 1 \; x \; n(t) + 1$, $Q(t) = (q_{i,j}(t))$ and the *initial state probability vector* $p(0)$, where the time dependent value $q_{i,j} (i \neq j)$ is the rate at which the state i changes to the state j and $q_{i,I} = -\sum_{i \neq j} q_{i,j}$ represents the rate for the event of staying in the same state.

Because $X(t) = k$ is a birth-and-death process, it can be described by following state dependent birth $q_{k,k+1} = \lambda_k(t)$ and death $q_{k,k-1} = \mu_k(t)$ rates:

$$\lambda_k(t) = \begin{cases} \lambda(t), & if \;\; 0 \leq k \leq s(t) - 1 \\ \gamma\lambda(t), & if \;\; s(t) \leq k \leq n - 1 \end{cases} \tag{1}$$

$$\mu_k(t) = \begin{cases} k\mu(t), & if \;\; 1 \leq k \leq s(t) - 1 \\ s(t)\mu(t) + (k - s(t))\eta, & if \;\; s(t) \leq k \leq n \end{cases} \tag{2}$$

The transient distribution at time t $p(t)$ for a given time dependent generator matrix $Q(t)$ can be calculated using Kolmogorov‘s forward equations:

$$p'(t) = p(t)Q(t) \tag{3}$$

where the vector $p(t) = [p_0(t) \dots p_n(t)]$ gives probabilities of the system being in any of the states at time t.

As we do not allow blocking or abandonment due to the overflow of the system, the capacity of the queue has to be big enough to be considered practically infinite, which is insofar realistic, as the cost of setting practically unlimited queue space in the telecommunications equipment is negligible nowadays. The system size must, in consequence, ensure that the probability of being in the state n (blocking or abandoning service requests) is insignificant compared to the required computational precision of the whole model.

3. Multi-Step Uniformization with Steady-State Detection

The infinitesimal generator matrix $Q(t)$ of an inhomogenous continuous-time Markov chain (ICTMC) is time dependent and the process is described by modified Kolmogorov‘s forward equations (3).

When the changes in generator matrix Q occur in a discrete way at finite points of time and all rates are constant during the intervals between them, we could also replace the analyzed ICTMC with a sequence of homogeneous systems computing the state probability vectors for consecutive time periods recursively using uniformization as proposed e.g. in [16] or in [17].

In case of a call center, time dependent changes in Q can occur either discretely due to the changing number of servers or due to changes in the arrival rate. Since the forecast and current traffic data in call center management applications are already aggregated with their average values by an arbitrary period (e.g. 5, 15 or 30 minutes), we will further assume, similarly to [7], $Q(t)$ being accordingly piecewise constant and refer to such consecutive time periods of length Δ with the coresponding homogenous continuous-time Markov chains (HCTMCs) as steps.

Another approach adopting uniformization for time-inhomogenous CTMCs introduced by [18] with subsequent improvements by [19], [20] and [21] could be used if continuous arrival rates were available, reducing the error of the approximation with the average rates.

Uniformization or Randomization, known since the publication of Jensen in 1953 and, therefore, often referenced to as Jensen method, is the method of choice for computing transient behavior of CTMCs. Many authors compared its performance in different applications with the conclusion that it usually outperforms known differential equation solvers (e.g. [20], [22], [23]). To use uniformization we first define the matrix:

$$P = I + \frac{Q}{\alpha} \tag{4}$$

which for $\alpha \geq max_i(|q_{i,i}|)$ is a stochastic matrix. The value of α is called uniformization rate. Further, let

$$\beta(\alpha t, k) = e^{-\alpha t} \frac{(\alpha t)^k}{k!} \tag{5}$$

be the probability of a Poisson process with rate α to generate k events in the interval $[0, t)$. One now finds for $p(t)$

$$p(t) = p(0) \sum_{k=0}^{\infty} \beta(\alpha t, k)(P)^k \tag{6}$$

The formula (6) can be interpreted as a discrete time Markov process (DTMC) embedded in a Poisson process generating events at rate α.

The implemented uniformization algorithm is based on [23] and computes transient state probabilities for a CTMC with the following modification of (6):

$$p(t) = \sum_{i=0}^{\infty} \Pi(i) e^{-\alpha t} \frac{(\alpha t)^i}{i!} \tag{7}$$

where α is uniformization rate, as described in (4), and $\Pi(i)$ is the state probability vector of the underlying DTMC after each step i computed iteratively by:

$$\Pi(0) = p(0),\ \Pi(i) = \Pi(i-1)P \tag{8}$$

To compute $p(i)$, within prespecified error tolerance, in finite time, the computation stops when the remaining value of cdf of Poisson distribution is less than the error bound ε:

$$1 - \sum_{i=0}^{k} e^{-\alpha t} \frac{(\alpha t)^i}{i!} \leq \epsilon \tag{9}$$

with k being the right truncation point. As αt increases, the corresponding probabilities of small number of i Poisson events occurring become less significant. This allows us to start the summation from the l'th iteration called left truncation point with the equation 7 reduced to:

$$p(t) = \sum_{i=l}^{k} \Pi(i) e^{-\alpha t} \frac{(\alpha t)^i}{i!} \tag{10}$$

In [23] it is suggested that the values of l and k be derived by:

$$\sum_{i=0}^{l-1} e^{-\alpha t} \frac{(\alpha t)^i}{i!} \leq \frac{\epsilon}{2},\ 1 - \sum_{i=0}^{k} e^{-\alpha t} \frac{(\alpha t)^i}{i!} \leq \frac{\epsilon}{2} \tag{11}$$

The main computational effort of the algorithm lies in consecutive k matrix vector multiplications (MVM), necessary for calculation of epochs of DTMC in (8), and is of $O(\eta k)$ where η is the number of nonzero elements of (sparse) P. For large αt, as the distribution converges to normal, both left and right truncation points l and k in (11) will tend to be symmetric to the mean. The number $(l+k) / 2$ is consequently of $O(\alpha t)$ and the number of additional $(k-l) / 2$ MVMs for the given error tolerance of $O(\sqrt{\alpha t})$ and proportional to inverse cdf for that given ε. Therefore, although we could solve the $p(t)$ with any accuracy $\varepsilon > 0$,

choosing a higher, acceptable for a respective practical application, value would bring some computational advantage.

The savings due to (tighter) left truncation are, however, rather insignificant, unless the computation of the first significant DTMC is performed in a more efficient way.

An example of this, presented first in [24], is based on recognizing the steady-state of the underlying DTMC. If convergence of the probability vector in (8) is guaranteed then we can stop the MVM after arriving at the steady-state, i.e. let us assume that DTMC has the steady state solution $\Pi(\infty)$ and that after the S iteration of (8) $||\Pi(S)- \Pi(\infty)||_v = \delta(S)$, is smaller than some predefined threshold, where $||.||_v$ is an arbitrary vector norm. Then (10) changes to:

$$\hat{p}(t) = \begin{cases} \Pi(S) & if\ S \le l, \\ \sum_{i=l}^{S} \Pi(i) e^{-\alpha t} \frac{(\alpha t)^i}{i!} + \Pi(S)(1 - \sum_{i=0}^{S} e^{-\alpha t} \frac{(\alpha t)^i}{i!}) & if\ l < S \le k, \\ same\ as\ p(t)\ in\ (10) & if\ S > k \end{cases} \quad (12)$$

with $\hat{p}(t)$ used instead $p(t)$ denoting transient state probability vector computed using approximate steady state DTMC vector $\Pi(S)$. According to [25] for a predefined error bound ε (as in (9),(11)) the following inequality holds:

$$\|p(t) - \hat{p}(t)\| < \frac{\epsilon}{2} + 2\delta(S) \quad (13)$$

The computing of consecutive epochs of the DTMC is equivalent to the power method of finding stationary probability vector of a finite Markov chain. According to [26] if the stochastic matrix P is aperiodic convergence of the power method is guaranteed and the number of iterations k needed to satisfy a tolerance criterion ξ may be obtained approximately from the relationship

$$\rho^k = \xi,\ i.e.\ , k = \frac{\log \xi}{\log \rho} \quad (14)$$

where ρ is the magnitude of subdominant eigenvalue λ_2 of matrix P

$$1 = \|\lambda_1\| > \|\lambda_2\| \ge \|\lambda_3\| \dots \ge \|\lambda_N\| \quad (15)$$

reducing, consequently, the computational complexity to $O(\eta \log \xi / \log|\lambda_2|)$.

Since in most cases the size of the subdominant eigenvalue is not known in advance, the usual method of testing for convergence is to examine some norm of the difference of successive iterates:

$$\|\Pi_i(k) - \Pi_i(k-m)\| < \xi \quad (16)$$

In [26] it is recommended to use the relative convergence test of iterates spaced apart by m being function of the rate of convergence:

$$max_i \left(\frac{|\Pi_i(k) - \Pi_i(k-m)|}{|\Pi_i(k)|} \right) < \xi \quad (17)$$

and suggests envisaging a "battery" of different convergence tests in order to accept the approximation $\Pi(S)$ as being sufficiently accurate. The main risk in this approach is that in

order to ensure, with the above proposed methods, that the $\Pi(S)$ is steady, an additional computational effort for both the convergence tests and the required additional number of iterations can easily obliterate the potential savings.

However, in case of our model, we can easily calculate precise stationary distribution $\Pi(\infty)$ in advance, using global balance equations (e.g. as in [26]) with birth and death rates as in (1) and (2). Therefore, we can consequently, as proposed in [10] instead of iterating the DTMC vector in (8) up to a point S where it would probably satisfy required convergence tests, simply use the $\Pi(\infty)$ (instead of $\Pi(S)$, as proposed in the original algorithm by [24]) as the $\hat{p}(t+\Delta)$ approximation of $p(t+\Delta)$.

This can be decided after relatively few iterations due to convergence properties of the power method as described e.g. in [27] or in standard books on numerical analysis, using numerically estimated convergence function of $\Pi(i)$ (as proposed in [10]), as it allows for precise calculation of the error of such an approximation:

$$\varepsilon_{t+\Delta} = \frac{\|\Pi(l)-\Pi(\infty)\|_\infty}{\|\Pi(\infty)\|_\infty} \tag{18}$$

in order to decide if it is acceptable (smaller than the steady-detection threshold δ_t).

One of the biggest advantages of the uniformization is its strict error bounding for one step independently of its length. It is not difficult to show (e.g. [28]) that the total error for a number of uniformization steps is the sum of truncation errors (error bounds) for each step.

Assume for a time period T with a known initial distribution $p(0)$ that for any $p(t)$, t=$(0, T]$ the value of each its state has to be computed with an error less than ε_T . Let us further assume $\varepsilon_t < \varepsilon_T$ being the error after computing some $p(t)$, $t < T$. Then:

$$\varepsilon_t + \sum_i \epsilon_{\Delta_i} \leq \varepsilon_T, \ \sum_i \Delta_i = T - t, \tag{19}$$

According to e.g. [20] for $\varepsilon_R = \varepsilon_T - \varepsilon_t$ being the remaining error in a step of length $\Delta \leq (T-t)$ starting with $p(t)$ the error should be:

$$\epsilon_\Delta \leq \varepsilon_R \frac{\Delta}{T-t} \tag{20}$$

to not exceed the error ε_T. This implies distribution of the error bound proportional to the length of the respective single interval. Although it is very intuitive, one could also consider, according to the already mentioned computational complexity of higher right truncation values which is asymptotically of $O(\sqrt{\alpha t})$, to set rather higher error bounds for the steps with smaller αt (shorter size or lower activity) or, in our case, trade them for higher steady-state detection thresholds.

In particular, as the error bound of steady state approximation is, in case the steady state is reached, absolute and independent of the error of the previous steps, we can set the convergence threshold δ dependent rather on the actual total error bound than the error for the single step (as proposed e.g. by [25]). It allows, consequently, to trade the error bounds of

steps for higher convergence thresholds while still within the global error bound for the whole solution. Then, assuming the system at time m, $0 \le m < T$ – to satisfy $\varepsilon_t < \varepsilon_T$ for each $p(t)$, $t=(m, T]$ we have to:

$$\delta_m \le \varepsilon_T - \varepsilon_m - \sum_m^T \epsilon_\Delta \tag{21}$$

4. Computational Examples

To test the implementation the following model has been used: a service system (call center) working for time T = 24h and starting empty. The arrival rate changes sinusoidal with two peaks and is divided into 288 (5min) periods with constant averaged rates, same as the first example in [10]. The service rate and number of servers are constant ($\mu(t) = \mu$, $s(t) = s$), the arrival rate varies in time - $\lambda(t) = s\mu(0.85 + 0.2\sin(3\pi t / T)$, $0 \le t < T$ (the load varying between 0.65 and 1.05 as shown in Figure 1. The probability 1 - γ of a customer immediately leaving when not served immediately is 0.03. The mean value of patience time 1 / η is equal to 4 minutes.

The capacity of the queue is constant and chosen so that for all times the probability $p_n(t)$ of the system being in the state n is less than 1×10^{-5} for all tested system sizes.

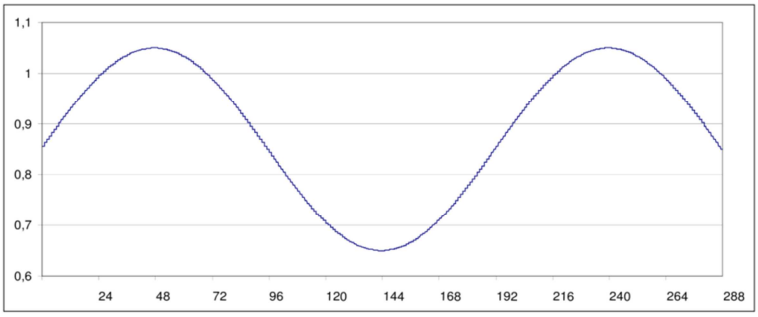

Fig. 1. System load
Rys. 1. Obciążenie systemu

To evaluate the impact of the proposed steady-state detection algorithm, models of 5 different sizes have been at first calculated using unmodified uniformization algorithm with an error step $\varepsilon = 1.5\times10^{-5}$ corresponding to the total error bound $\varepsilon_T = 2.88\times10^{-3}$.

All experiments were performed on a 1.7GHz PC under 64bit Linux OS with a processor supporting vector operations (an Intel i5-3317U with CPU throttling disabled via kernel scaling governor using avx instuction set with 256bit vectors – 4 double or 8 float operations simultaneously) , compiled with GNU GCC compiler. All measurements use standard Unix *time.h*/*clock*() function – returning CPU time. All times are in milliseconds. The detailed results of computation times are in Table 1.

Table 1

Computation times, steady-state detection, load $0.65 \leq \rho \leq 1.05$

ε_Δ=1e-7	δ=0(ε_Δ=1e-5)		ε_T= 5e-03		ε_T= 1.5e-02		ε_T= 3e-02		ε_T= 5e-02	
System size	time (ms)	t/n^2	time (ms)	t/n^2	time (ms)	t/n^2	time (ms)	t/n^2	time (ms)	t/n^2
54.....(30+24)	4.25	1.46	4.27	1.46	3.25	1.11	2.13	0.731	2.48	0.850
150....(100+50)	15.8	0.70	15.3	0.68	11.5	0.51	4.46	0.198	3.73	0.166
390 ...(300+90)	90.1	0.59	86.2	0.57	62.0	0.41	43.6	0.286	15.3	0.101
1200(1000+200)	709	0.49	715	0.50	534	0.37	468	0.325	378	0.262
3300(3000+300)	5996	0.55	5547	0.51	4350	0.40	4053	0.372	3915	0.359

The impact of reduced computational effort due to steady-state detection for some chosen total error bounds (between 0 and 5×10^{-2}), with corresponding steady-state detection thresholds, is illustrated for the system of size 1200 in Figure 2.

Figure 3 shows the expected state of the system, derived from the calculated probability vector as:

$$ES(t) = \sum_i i\pi_i(t),\ p(t) = [\pi_0 \dots \pi_n]$$

Figure 4 shows its relative error for different steady-state detection thresholds. The reference for the error estimate has been calculated with $\varepsilon_\Delta = 1\times10^{-13}$.

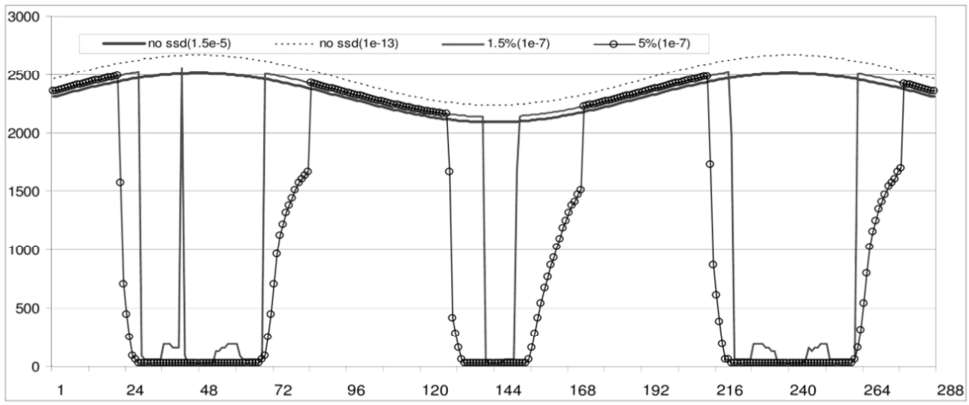

Fig. 2. Number of iterations (mvm) per step (load $0.65 \leq \rho \leq 1.05$, s = 1000, q = 200)
Rys. 2. Liczba operacji (mnożenia wektora matrycy) na interwał

Figure 5 shows the probability for an incoming service request to be served immediately (with no waiting time).

5. Conclusion

In this paper we showed that the uniformization with steady-state detection can be used in a very effective way to evaluate transient behavior of multiserver queues. Applied to the modeling of the call center schedules, it allows calculation of transient system states for systems of any, possible in practical applications, size in a very short time, in a numerically

stable way, with very high precision, using relatively common and inexpensive CPU. It can, therefore, be used for schedule planning based on available forecasts, as described in [7].

The presented method can be extended in several directions. One could be, in regard to call center modeling, to automatically optimize the model size (queue length) with significant impact on the computational efficiency. Another could be to use known periodicity of traffic forecasts to divide total error bound in between known times of the day, bounded by the points of time when the system will reach a steady state, than for the whole modeled period.

BIBLIOGRAPHY

1. Aksin Z., Armony M., Mehrotra V.: The modern call center: A multi-disciplinary perspective on operations management research. Production and Operations Management 16(6) (Nov 2007), 665-688.
2. Gans N., Koole G., Mandelbaum A.: Telephone call centers: Tutorial, review, and research prospects. Manufacturing & Service Operations Management 5(2) (Apr 2003), 79-141.
3. Brown L., Gans N., Mandelbaum A., Sakov A., Shen H., Zeltyn S., Zhao L.: Statistical analysis of a telephone call center. Journal of the American Statistical Association 100(469) (Mar 2005), 36-50.
4. Deslauriers A., L'Ecuyer P., Pichitlamken J., Ingolfsson A., Avramidis A. N.: Markov chain models of a telephone call center with call blending. Computers & Operations Research 34(6) (Jun 2007), 1616-1645.
5. Phung-Duc T., Kawanishi K.: Performance analysis of call centers with abandonment, retrial and after-call work. Performance Evaluation 80 (Oct 2014), 43-62.
6. Green L.V., Kolesar P.J., Whitt W.: Coping with time-varying demand when setting staffing requirements for a service system. Production and Operations Management 16(1) (Jan 2007), 13-39.
7. Ingolfsson A., Campello F., Wu X., Cabral E.: Combining integer programming and the randomization method to schedule employees. European Journal of Operational Research 202(1) (Apr 2010), 153-163.
8. Ingolfsson A., Akhmetshina E., Budge S., Li Y., Wu X.: A survey and experimental comparison of service-level-approximation methods for nonstationary m(t)/m/s(t) queueing systems with exhaustive discipline. INFORMS Journal on Computing 19(2) (May 2007), 201-214.
9. Bylina J., Bylina B., Zoła A., Skaraczyński T.: A markovian model of a call center with time varying arrival rate and skill based routing. In: Computer Networks. Springer Science Business Media (2009), 26-33.

10. Burak M.: Multi-step uniformization with steady-state detection in nonstationary m/m/s queuing systems. arXiv preprint arXiv:1410.0804 (2014).
11. Czachórski T., Fourneau J. M., Nycz T., Pekergin F.: Diffusion approximation model of multiserver stations with losses. Electronic Notes in Theoretical Computer Science 232 (Mar 2009), 125-143.
12. Czachórski T., Nycz T., Nycz M., Pekergin F.: Traffic engineering: Erlang and engset models revisited with diffusion approximation. In: Information Sciences and Systems 2014. Springer Science – Business Media (2014), 249-256.
13. Mandelbaum A., Zeltyn S.: Staffing many-server queues with impatient customers: Constraint satisfaction in call centers. Operations Research 57(5) (Oct 2009),1189-1205.
14. Whitt W.: Sensitivity of performance in the Erlang-a queueing model to changes in the model parameters. Operations Research 54(2) (Apr 2006), 247-260.
15. Artalejo J., Pla V.: On the impact of customer balking, impatience and retrials in telecommunication systems. Computers & Mathematics with Applications 57(2) (Jan 2009), 217-229.
16. Rindos A., Woolet S., Viniotis I., Trivedi K.: Exact methods for the transient analysis of nonhomogeneous continuous time Markov chains. In: Computations with Markov Chains. Springer US (1995), 121-133.
17. Gross D., Miller D. R.: The randomization technique as a modeling tool and solution procedure for transient Markov processes. Operations Research 32(2) (Apr 1984), 343-361.
18. Van Dijk N. M.: Uniformization for nonhomogeneous Markov chains. Operations Research Letters 12(5) (Nov 1992), 283-291.
19. Van Moorsel A.P., Wolter K.: Numerical solution of non-homogeneous Markov processes through uniformization. In: ESM. (1998), 710-717.
20. Arns M., Buchholz P., Panchenko A.: On the numerical analysis of inhomogeneous continuous-time Markov chains. INFORMS Journal on Computing 22(3) (Aug 2010), 416-432.
21. Andreychenko A., Crouzen P., Mikeev L., Wolf V.: On-the-fly uniformization of time-inhomogeneous infinite Markov population models. arXiv preprint:1006.4425 (2010).
22. Grassmann W.: Transient solutions in markovian queueing systems. Computers & Operations Research 5(2) (Jan 1978), 161.
23. Reibman A., Trivedi K.: Numerical transient analysis of Markov models. Computers & Operations Research 15(1) (Jan 1988), 19-36.
24. Muppala J. K., Trivedi K. S.: Numerical transient solution of finite markovian queueing systems. Oxford Statistical Science Series (1992), 262-262.

25. Malhotra M., Muppala J. K., Trivedi K. S.: Stiffness-tolerant methods for transient analysis of stiff Markov chains. Microelectronics Reliability 34(11) (Nov 1994), 1825-1841.
26. Stewart W.J.: Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling. Princeton University Press (2009).
27. Ò Leary D. P., Stewart G. W., Vandergraft J. S.: Estimating the largest eigenvalue of a positive definite matrix. Mathematics of Computation 33(148) (Oct 1979), 1289.
28. Van Moorsel A., Sanders W.: Transient solution of Markov models by combining adaptive and standard uniformization. IEEE Transactions on Reliability 46(3) (1997), 430-440.

Omówienie

Artykuł opisuje zastosowanie łańcuchów Markowa z czasem ciągłym do modelowania systemów telefonicznej obsługi klienta (Call Center). Wykorzystany model uwzględnia zjawisko niecierpliwości klientów, tzn. klient może się rozłączyć albo natychmiast, gdy znajdzie się w kolejce (ang. *balking*), albo gdy czas oczekiwania w kolejce przekroczy jego cierpliwość (ang. *abandonment*). W celu uwzględnienia zmian matrycy intensywności przejść w czasie, model jest rozwiązywany w interwałach odpowiadających skokowym zmianom matrycy intensywności przejść (np. zmiana obsady, przerwy, awarie) oraz zmianom natężenia ruchu wynikającym z prognozy ruchu. Modelowanie poszczególnych (jednorodnych w czasie) interwałów, tzn. wyliczanie odpowiednich (przejściowych) rozkładów prawdopodobieństwa stanów, odbywa się za pomocą zmodyfikowanego algorytmu uniformizacji, wykrywającego osiągnięcie, z predefiniowaną dokładnością, stanu stacjonarnego, w celu zmniejszenia nakładu obliczeniowego. Metoda umożliwia bardziej dokładne modelowanie systemów zmiennych w czasie niż popularne metody bazujące na przybliżeniach stacjonarnych, z wydajnością umożliwiającą ich bezpośrednie zastąpienie w zastosowaniach praktycznych, takich jak np. planowanie obsady.

Address

Maciej Rafał BURAK: Katedra Zastosowań Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie, ul Sikorskiego 37, 70-313 Szczecin, maciej.burak@zut.edu.pl