UDC 681.3+004.73.052

V.O. BUTENKO[1], O.N. ODARUSHCHENKO[2], V.S. KHARCHENKO[1]

[1] ***National aerospace university "KhAI", Kharkov, Ukraine***
[2] ***RPC "Radiy", Kirovograd, Ukraine***

METRIC-BASED ANALYSIS OF MARKOV MODELS FOR COMPUTER SYSTEM AVAILABILITY ASSESSMENT

Markov and semi-Markov models are widely used to analyse the reliability of complex computer-based systems. Dealing with the different model features is a serious problem, which leads to computational difficulties and may affect the accuracy of the reliability analysis. We discuss the classification attributes (stiffness, largeness, sparsity and fragmentedness) that are used for computer systems reliability analysis. The provided system analysis based on this classification attributes can determine the complexities of computational problem and form recommendations for the most effective methods choose.

Keywords: *classification feature, stiffness, largeness, sparsity, fragmentedness.*

Introduction

High availability is being demanded for safety-critical and life-critical systems, national and international telecommunication systems, commercial applications such as e-commerce systems, financial systems and stock-trading systems. In general those systems can be classified as high availability systems (HAS).

The quantitative assurance of such systems availability is provided by means of stochastic availability models constructed based on the structure of the system hardware and software. As the practical example of such models use the availability analysis of IBM BladeCenter system can be considered [1]. The reliability block diagrams or faults trees cannot easily incorporate the realistic system behavior, such as multiple failure modes, failure/repair dependencies, shared repair facilities of hot swap [1 – 4]. In contrast, those features and dependencies can be captured by flexible state-space models such as Markov chains (MC), semi-Markov processes (SMP) [2, 3]. However, process of construction, storage and solution of such models can be difficult. Here we provide the brief description of main stages in research process using Markov research apparatus.

i) The general analysis of research system architecture, taking into account different defect types (software and hardware defects) and recovery procedures.

ii) Definition the system states space based on the combination of working, no-working, recovery and service states etc.

iii) Definition of initial parameters values, base on known and developed methodologies and provided assumptions.

iv) Development, research and solution of Kolmogorov's differential equations (DE) system. Achieving the *transient measures* as probabilities of each system state on the research time interval [2, 3].

On the last stage of research process (iv), the features of initial real system and research apparatus can cause difficulties with use of numerical methods.

In this paper we propose four classification attributes: stiffness [5], largeness [8], sparsity [21] and fragmentedness [14, 22]. Those attributes can be used to provide the description of research HAS properties, that are presented using Markov modeling and cause difficulties in numerical methods and algorithms use.

Analysis of HAS based on this classification attributes can help to determine the risks of solution the DE that are derived from Markov model (MM), and also provide the detailed system "personality" .

Paper structure. In section 1 we provide the formally definition of the classification attribute *stiffness,* give the example of research system properties that can cause *stiffness* in MM and basic approaches of how to deal with this feature. In sections 2, 3, 4 the formal definitions of classification attributes *largeness*, *sparsity* and *fragmentedness* are presented. In section 5 we describe the impact of each attribute on the process of solution the developed MM and present their combinations. In section 6 we provide two examples of systems analyzed using defined classification attributes. At last we present the conclusions and the problems left for future research.

1. Stiffness

There is no common definition of "stiffness" because of it complexity. In publications [5, 9] authors introduce the "practical" definitions of stiff problems, based on the interpretation of physical processes in research systems. Here we provide the example of "practical" formation of stiffness problem.

In general there are two ways to improve the availability: increase time-to-failure or reduce time-to-recovery. The system failures can be caused by various types of defects (bohrbugs, heisenbugs, aging-related bugs [10]) the rate of which may vary in orders (more then 10^2). The difference in orders of software – hardware system failure and recovery rates values [1] can be shown as an example that system has a feature of *stiffness* [6]. The given difference appears in matrix of coefficients of Kolmogorov DE and lead to inefficiency of explicit numerical methods use [5].

In research works [11, 12] authors present the definition of stiffness based on the problems of numerical solution: inability or ineffectively use of explicit numerical methods; presence of quick perturbations decay; big Lipschitz constants; big difference of Jacobi matrix eigenvalues etc.

One of the most vide used stiffness definition methods is based on the calculation of stiffness index – s [11, 12].

The Cauchy problem $\frac{du}{dx} = F(x,u)$ is said to be stiff on the interval [x_0, X] if for *x* from this interval the next condition is fulfilled:

$$s(x) = \frac{\max\limits_{i=1,n} |Re(\lambda_i)|}{\min\limits_{i=1,n} |Re(\lambda_i)|} >> 1 , \qquad (1)$$

where s(x) – denotes the index of stiffness; $л_i$ – are the eigenvalues of a Jacobi matrix; ($Re\lambda_i < 0, i = 1,2,...,n$).

In work [5] Ernst Hairer and Gerhard Wanner propose two possible methods of prior detection of stiffness in researches DE system. The implementation of automatically stiffness detection can help to avoid the not accurate, in case of stiffness, numerical methods. The first method is based on the analysis of errors on first steps of system DE solution (not more than 15 steps). The second possibility is based on the estimation directly the dominant eigenvalue of the Jacobian of the problem.

In the last 30 years a lot of approaches have been developed to deal with the problem of stiffness [6, 11, 25]. They can be separated into two groups - *stiffness-tolerance* and *stiffness-avoidance* approaches [25]. The main feature of stiffness-tolerance is to solve the stiff MM using *special numerical methods* that can provide highly accurate results. The limitations of this technique are: i) it cannot deal effectively with large models, and ii) computational efficiency is difficult to achieve when highly accurate solutions are sought. The stiffness-avoidance solution, on the other hand, is based on an *approximation algorithm* for systematically converting a stiff MC into a non-stiff chain first which typically has a significantly smaller state space [25]. An advantage of this approach is that it can deal effectively with large stiff MMs, while achieving high accuracy may be problematic. Detailed analysis of given classification attribute can determine the type of *stiffness* and according to it choose the optimal (as combination of estimation time, resource cost and accuracy) computational method [5, 11, 13].

2. Largeness

As a second classification atribute of researched system model the term of *largeness* can be used.

In the modeling process the real object is presented with some level of specification. Determine the level of specification at different stages of the modeling process is unique for each system. The nowadays HAS are complex hardware-software systems. High requirements to the reliability of such systems operating process force the modeler to decompose the system to the elementary parts to provide the accurate in-depth analysis. The process of including more details in model makes it larger and more complicated so its analysis will be more difficult or even intractable [20].

Methods of large MM solution can be divided into two types: *largeness-tolerant* and *largeness-avoidance*.

i) *Largeness-tolerance* approach is based on the detailed specification of research system and automatically generation of it states space. For this stage the special software packages are used, so called *state-space generators* (SAVE [15]), that convert the high-level specification of a model into its equivalent underlying CTMC. Sparsity of Markov chains is exploited to reduce the space requirements but no model reduction is employed [6]. Appropriate data structures for sparse storage are used.

ii) Two most used methods in the *largeness-avoidance* approach are: state-truncation based on the avoiding generation of low probability states [16] and model-level decomposition [17].

The hierarchical approach [1] also can be used to reduce the system MM states-space. It is based on the combination of state-space models and combinatorial models: high-level fault tree model with a number of lower-level MM.

The analysis of MM using classification attribute *largeness* will reduce the time for system assessment using special algorithms and amount of computing resources.

3. Sparsity

Conceptually, sparsity [18] corresponds to systems which are loosely coupled. In the subfield of numerical analysis, a sparse matrix is a matrix populated primarily with zeros [19].

The analysis of classification attribute *sparsity* is important part for the special class of problems. As an example of such class we can use the solution of Kolmogorov DE, which describes the MM of system under research. As the matrix of DE coefficients is presented in mostly diagonal form so given attribute can be accompanying in case of using the apparatus of Markov modeling. If research MM is large the sparsity can cause additional assessment difficulties.

Storing and manipulating sparse matrices on a computer is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix. Operations using standard dense-matrix structures and algorithms are relatively slow and consume large amounts of memory when applied to large sparse matrices.

Most of the approaches are developed to reduce the size of the transition matrix representation and form the dense matrix, by using structured analysis[8] or symbolic data structures analysis [20] and solving them using lumping algorithms [8, 20] or iterative techniques [8].

It is also recommended [20] to conduct the computation of *sparsity index*. The authors of [20] examine and compare quantitatively, several commonly-used sparsity measures based on intuitive and desirable. Their finding is that only the Gini index has all these attributes. The Gini index is independent of size and dimension. We will introduce the common statement that is based on the performance and calculation of Gini index on the vector.

Gini index (G): Given a vector

$$f = [f(1), \dots, f(N)],$$

with its elements re-ordered and represented by $f_{[k]}$ for $k = 1,2,\dots,N$, where $|f_{[1]}| \le |f_{[2]}|,\dots,\le |f_{[N]}|$, then

$$G(f) = 1 - 2\sum_{k=1}^{N} \frac{|f_{[k]}|}{\|f\|_1}\left(\frac{N-k+1/2}{N}\right). \qquad (2)$$

The analysis of classification attribute sparsity in the process of system research will reduce the time for system assessment by using specialized techniques, algorithms and data structures that take advantage of the sparse structure of the matrix.

4. Fragmentedness

In the modeler provide the assumption of that system parameters can vary in the process of functioning the last classification attribute can be used – *fragmentedness* [22, 23].

The variation of parameters is a plausible concept. For instance, software may well perform different tasks with different importance, which would justify different degree of testing, hence different rates of failure and repair in the respective partitions. Using the principle of multi-fragmentation [14, 22] the assumption of system parameters change can be presented as MM divided into N fragments that are differ in one or more parameters. Here we present the basic terms and definitions that are used to describe and analyse the MM using classification attribute – *fragmentedness*[22, 23, 24].

Macromodel - the model, basic elements of which are independent models (fragments), that describe the system behavior on the define time interval [14, 22].

Fragment (initial, internal, final) – typical independent part of macromodel.

Zone of fragments – set of fragments, in the bounds of which system parameters can vary based on one rule.

Macrograph – state graph which corresponds to the macromodel and describe the process of transitions between fragments.

As an example, in section 6 the computer system with two hardware channels, each running control software is presented using the assumption of parameters change. The use of classification attribute fragmentedness will increase the clarity of research model and take into account some properties of operating system modes. It is necessary to understand that introduction of parameters change assumption can increase the system size (direct affect on the feature *largeness*) and as a result the increase the sparsity of system transitions matrix (affect on the feature *sparsity*).

5. Characteristics combination

Analysis of HAS based on this classification attributes can help to determine the risks of solution the DE that are derived from Markov model (MM), and also provide the detailed system "personality". The possible benefits of use the defined classification attributes in the process of system analysis are presented on the Fig. 1.

For instance, the detection of stiffness in Markov model can reduce the time and increase the accuracy of stiff DE solution [5]. Analysis of classification attribute *largeness* can help to select the most effective special methods for large MM solution [6].

The sparseness of the transition matrix, that is built on the basis of the developed Markov model, directly depends on the dimension of the problem and the specification level of the system under investigation. Accounting in the analysis process of this classification attribute of MM allows to effectively use both the time of research and the device memory, on which the calculations are making, with special algorithms [8, 20]. The last classification attribute allows representing the process of system parameters change with a high level of detalisation, improving the model clarity.

Fig. 2 shows the 16 combinations of the classification attributes, that can determine the "personality" of the system.

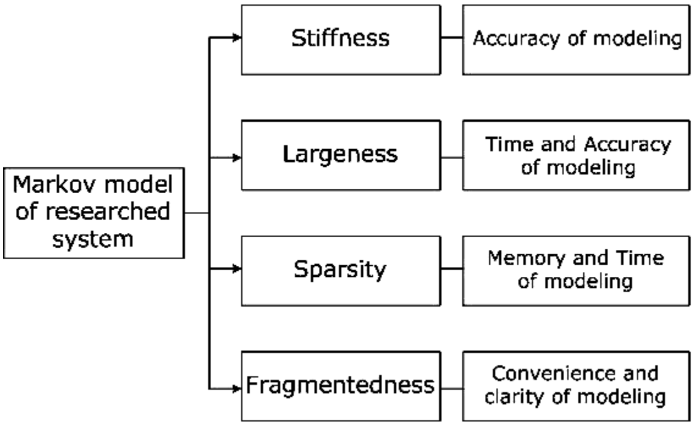

Fig. 1. Possible results of classification attributes use

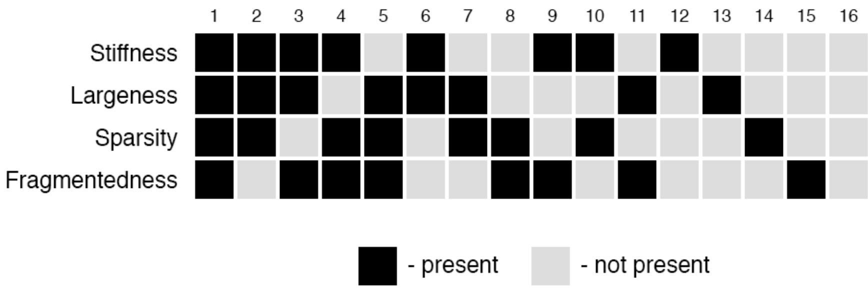

Fig. 2 Classification attributes combination

6. Example

Here we provide two examples to describe the classification attributes use.

a) As the first example we consider the fault-tolerant computer system (FTCS) with two hardware channels each executing software control. The system can be described using continuous-time MC (CTMC). The system graph is presented on the Fig. 3.

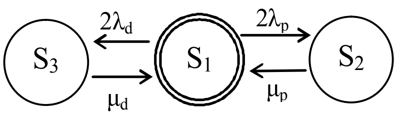

Fig. 3. FTCS one-fragment graph

The system parameters are following:

- λ_d and μ_d – software failure and recovery rates;
- λ_p and μ_p – hardware failure and recovery rates.

Informally, the operation of the system is as follows. Initially the system is working correctly – both hardware and software channels deliver the service as expected. If during the operation one of the hardware channels has failed the system operation will be failed over to the second channel until the first channel is "repaired".

Similarly, a software component may fail, in which case a failover will take place to the other channel, etc.

We provide the assumption that λ_d >> μ_d and their ratio is about 10^3 [24].

Also we suppose that research system parameters are constant. From the model (Fig. 3) we can derive the following system of Kolmogorov equations (3), initial conditions (4) and matrix of its DE coefficients (5).

$$\begin{cases} \dfrac{dP_1(t)}{dt} = -(2\lambda_p + 2\lambda_d)P_1(t) + \mu_p P_2(t) + \mu_d P_3(t); \\ \dfrac{dP_2(t)}{dt} = -\mu_p P_2(t) + 2\lambda_p P_1(t); \\ \dfrac{dP_3(t)}{dt} = -\mu_d P_3(t) + 2\lambda_d P_1(t); \end{cases} \quad (3)$$

$$P_1(0) = 1, P_2(0) = 0, P_3(0) = 0. \quad (4)$$

$$\begin{pmatrix} -(2\lambda_p + 2\lambda_d) & \mu_p & \mu_d \\ 2\lambda_p & -\mu_p & 0 \\ 2\lambda_d & 0 & -\mu_d \end{pmatrix}. \quad (5)$$

Based on the presented system transitions matrix (3) and provided assumptions we can derive that the system is stiff, no-large, no-sparse and no-fragmented. According to the classification attributes combinations (Fig. 2) the system refers to the 12 combination. The main attention, in case of this system, is concentrated on the problem of stiffness that affect choose of effective (as combination of estimation time, resource cost and accuracy) solution methods [5, 11, 13].

b) As the second example we consider the same system but with use of assumption of software parameters change. The system operating graph is presented on Fig. 4.

An important feature of this MM (Fig. 4) is that as a result of software repair (e.g. restart of the failed channel) we assume that the rate of software failure of both channels will deteriorate by a small constant $\Delta\lambda_d$. Also we assume that the rate of software repair of both channels will decrease on the small $Дм_d$ [24]. In [14, 22, 23, 24] the detailed description of this system operating process, justification of assumptions about system parameters change and process of multi-fragmental MM construction are presented.

Also we provide the assumption, similar to the previous example, that $л_d$ >> $м_d$ and their ratio is about 10^3 [24].

From the MM (Fig. 4) we derive the matrix of system transition rates (6), where i=(1,..,n) – number of system fragments:

$$\begin{pmatrix}
-(2\lambda_p+\lambda_d) & \mu_p & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
2\lambda_p & -\mu_p & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_d & 0 & -\mu_d & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_d & -\left(\Lambda_d-i\Delta\lambda_d+2\lambda_p\right) & \mu_p & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\lambda_p & -\mu_p & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Lambda_d-\Delta\lambda_d & 0 & -(\mu_d-i\Delta\mu_d) & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & -(\mu_d-n\Delta\mu_d) & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \mu_d-n\Delta\mu_d & -\left(\Lambda_d-n\Delta\lambda_d+2\lambda_p\right) & \mu_p & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2\lambda_p & -\mu_p & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \Lambda_d-n\Delta\lambda_d & 0 & -(\mu_d-n\Delta\mu_d) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \mu_d-n\Delta\mu_d & -2\lambda_p & \mu_p \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 2\lambda_p & -\mu_p
\end{pmatrix} \quad (6)$$

Based on the presented system transitions matrix (6) and provided assumptions we can derive that the system is stiff, no-large, sparse and fragmented which refers to the 4th combination (Fig. 2). Constructing MM (Fig. 4) we suppose that system consist of N=5 fragments, with states-space amount S_i = 1,…20.

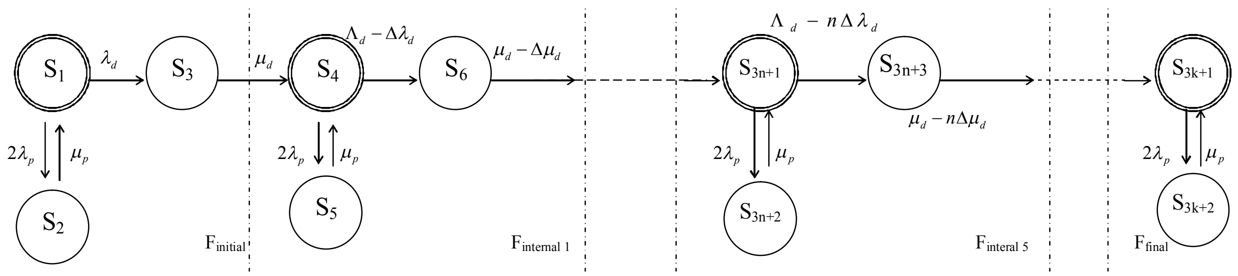

Fig. 4. FTCS multi-fragment graph

Conclusion

An important aspect in the process of HAS modeling using MM approach is consideration of all system and research methodology features.

In this paper we propose four classification attributes: stiffness [5], largeness [8], sparsity [21] and fragmentedness [14, 22].

Those attributes can be used to provide the description of research HAS properties, that are presented using Markov modeling and cause difficulties in numerical methods and algorithms use.

Analysis of HAS based on this classification attributes can help to determine the risks of solution the DE that are derived from Markov model (MM).

Analysis of proposed classification attributes, their combinations and two examples of those attributes use are presented in this paper.

In our future work we intend to extend the analysis based on classification attributes presented in the paper. As a result we are hoping to define the best solution method that can easily deal with the complex problem. Under the *complex problem* we mean that in research system is stiffness, largeness, sparse and fragmented.

References

1. Availability analysis of multi–component blade server systems [Text] / W. Earl Smith, K.S. Trivedi, L. Tomek, J. Ackeret // Allen Press, Inc. IBM Systems Journal. – 2008. – Vol. 47, № 4. – P. 1 – 20.

2. Volkov, L. Managing the operation of the aircraft systems: Tutorial [Text] / L. Volkov. – M.: Vyshaya Shkola, 1981. – 368 p.

3. Ventsel', E. Probability theory and its applications in engineering [Text] / E. Ventsel', L. Ovcharov. – M.: Nauka, 2000. – 480 p.

4. Archana, S. Availability Models in Practice [Text] / S. Archana, R. Srinivasan, K. S. Trivedi. // Proc. of Int. workshop on fault–tolerant control and computing (FTCC–1), 22–23 May, Seoul, Korea. – 2000.

5. Hairer, E. Solving ordinary differential equations II: Stiff and differential–algebraic problems [Text] / E. Hairer, G. Wanner. – Berlin: Springer, 1993. – 640 p.

6. Malhotra, M. Stiffness–tolerant methods for transient analysis of stiff Markov chains [Text] / M. Malhotra, J.K. Muppala, K.S. Trivedi // Microelectronic Reliability. – 1994. – Vol. 34, № 11. – P. 1825–1841.

7. Reibman, A. Numerical Transient Analysis of Markov models [Text]/ A. Reibman, K.S. Trivedi // Computer Opns. Res. – 1988 – Vol.15, №1. – P. 19 – 36.

8. Buchholz, P. Structured analysis approach for Large Markov Chains. A Tutorial. [Text] / P. Buchholz // Proc. of Performane'96, Lausanne, Switzerland.

9. Kakhaner, D. Numerical methods and software [Text]: trans. from eng. / D. Kakhaner, K. Mouler, S. Nesh. – M.: Mir, 1998. – 320 p.

10. Achieving and assuring high availability [Text] / K. Trivedy, G. Ciardo, B. Dasarathy, M. Grottke, A. Rindos, B. Vashaw // Distributed and Network–Centric Systems: proc. 13th IEEE Workshop on Dependable parallel, 22nd IEEE International Parallel & Distributed Processing Sypmosium. – 2008.

11. Arushanyan, O. Numerical solution of ordinary differential equations using FORTRAN [Text] / O. Arushanyan, S. Zaletkin. – M.: Moscow State University, 1990. – 336 p.

12. Rakitsky, U. Numerical methods for stiff differential equation systems solution [Text] / U. Rakitsky, S. Ustinov, I. Chernorudskiy. – Ì.: Nauka, 1979. – 208 p.

13. Queuing Networks and Markov Chains [Text]/ G. Bolch, S. Greiner, H. de Meer, K.S. Trivedy. – A Wiley-Interscience Publication: John Wiley & Sons, Inc., 1998. – 747 p.

14. Odarushchenko, O. Computer systems reliability assessment taking into account software failure and recovery rates change / O. Odarushchenko, E. Odarushchenko, V. Yakovlev // Theory and technique of information send, receive and processe: Math. 8th International conference. – Kharkov: KhNURe, 2002. – P. 269 – 271.

15. The system availability estimator [Text] / A. Goyal, W.C. Carter, E. de Souza e Silva, S.S. Lavenberg, K. S. Trivedy // IEEE Computer Society Press. – 1986. – P. 84 – 89.

16. An approach to solving large reliability models [Text] / M. Boyd, M. Veeraraghvan, J.B. Dugan, K.S. Trivedy // Proc. Of IEEE/AIAA DASC Symposium, San Diego. – 1988.

17. Tomek, L.A. Fixed pont iteration in availability modeling/ L.A. Tomek and K.S. Trivedy// Springer: ProceedingsoOf the 5th International GI/ITG/GMA Conference on Fault–Tolerant Computing Systems. – 1991. – P. 229–240.

18. Gene, H. Golub. Matrix computations [Text]/ Golub H. Gene, Charles F. Van Loan . – JHU Press. – 1996. –694 p.

19. Stoer, J. Introduction to Numerical Analysis [Text]/ J. Stoer, R. Bulirsch. Berlin, New York: Springer-Verlag. – 2002. –732 p.

20. Derisavi, S. Optimal state–space lumping in Markov chains [Text]/ S. Derisavi, H. Holger, W.H. Sanders // Inf. Process. Lett. – 2003. – Vol. 87, 6. – P. 309 – 315.

21. Hurley, N. Comparing measures of sparsity [Text] / N. Hurley, S. Rickard // IEEE Trans. on Information Theory. – 2009. – Vol. 55, № 10. – P. 47.23 – 47.41.

22. Odarushchenko, O. Multi–fragmental Marko models use in case of assessment the reliability of hardware–software complex systems [Text] / O. Odarushchenko // Coll. of scientific publ. NAS of Ukraine. PANU. – 1997. – Vol.1, №5. – C. 102 – 105.

23. Kharchenko, V. Fundamentals of design and constructions the automated control systems for aircraft technical state control Study guide [Text] / V. Kharchenko, G. Timonkin, V. Sychev // KVKIU, Kharkov. – 1992. – P. 276.

24. Empirical evaluation of accuracy of mathematical software used for availability assessment of fault–tolerant computer systems [Text] / V. Kharchenko, P. Popov, O. Odarushchenko, V. Zhadan // RT&A . – 2012. – Vol. 03, №26. – P. 85 – 97.

25. Bobbio, A. A aggregation technique for transient analysis of stiff Markov chains [Text] / A. Bobbio, K.S. Trivedy // In: Computers, IEEE Transactions. – 1986. – Vol. C-35. – P. 803 – 814.

Поступила в редакцию 15.02.2013, рассмотрена на редколлегии 20.03.2013

Рецензент: д-р техн. наук, проф., проф. каф. инженерии программного обеспечения Б.М. Конорев, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

МЕТРИЧНИЙ АНАЛІЗ МАРКІВСЬКИХ МОДЕЛЕЙ ДЛЯ ОЦІНКИ ГОТОВНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ

В.О. Бутенко, О.М. Одарущенко, В.С. Харченко

Марківські та напівмарківські моделі широко використовуються для аналізу надійності складних комп’ютерних системи. Властивості даних моделей формують різноманітні труднощі в процесі їх обчислення та можуть вплинути на точність аналізу надійності досліджуваної системи. В роботі розглянуто класифікаційні ознаки (жорсткість, розмірність, розрідженість, фрагментність), що використовуються для аналізу надійності комп’ютеризованих системи. Аналіз надійності систем за даними класифікаційними ознаками дозволяє визначити складність обчислювальної задачі та сформулювати рекомендації щодо вибору найбільш ефективного методу розв’язку.

Ключові слова: класифікаційна ознака, жорсткість, розмірність, розрідженість, фрагментність.

МЕТРИЧЕСКИЙ АНАЛИЗ МАРКОВСКИХ МОДЕЛЕЙ ДЛЯ ОЦЕНКИ ГОТОВНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

В.О. Бутенко, О.М. Одарущенко, В.С. Харченко

Марковские и полумарковские модели широко используются для анализа надежности сложных компьютерных систем. Особенности данных моделей могут сформировать множество трудностей в процессе их исследования и повлиять на точность анализа надежности исследуемой системы. В работе рассмотрены классификационные признаки (жесткость, размерность, разреженность, фрагментность), используемые для анализа надежности компьютеризированных систем. Анализ надежности систем по данным классификационным признакам позволяет определить сложность решаемой вычислительной задачи и сформулировать рекомендации по выбору наиболее эффективного метода решения.

Ключевые слова: классификационный признак, жесткость, размерность, разреженность, фрагментность.

Бутенко Валентина Олеговна – аспирантка, Национальный аэрокосмический университет имени М.Е. Жуковского «ХАИ» , e-mail: v.odarushchenko@csn.khai.edu.

Одарущенко Олег Николаевич – канд. техн. наук, ведущий научный сотрудник НПП «Радий», e-mail: skifs2005@mail.ru.

Харченко Вячеслав Сергеевич – д-р техн. наук, профессор, зав. каф. Национального аэрокосмического университета имени М.Е. Жуковского «ХАИ», e-mail: v.kharchenko@khai.edu.