

development ASOIC: educational-methodical. Complex. Eastern State Technical University. Vladivostok: Izd FESTU, 212.5. *Muromtsev, L. J., Muromtsev, D. J., Tyurin, I. V.* (2010). Information technology in the design of electronic equipment: studies. allowance for stud. Moscow: Publishing Center "Academy", 384. 6. *Norenkov, I. P.* (2009). Fundamentals aided design: Textbook. for universities. Publishing House of the MSTU. Bauman, 430. 7. *Shekhovtsova, V. I.* (2010). Design thinking as a component of design culture specialist. News of NTU "KPI". Filosofiya, psihologiya, pedagogika, 3, 204-207. 8. *McConnell, C.* (2007). How much is a software project. Russian edition, St. Petersburg.: Peter, 297.

Надійшла (received) 25.05.2014

УДК 004.54

В. С. ЯКОВИНА, канд. фіз.-мат. наук, доц., НУ «Львівська політехніка»;
Д. В. ФЕДАСЮК, д-р техн. наук, проф., НУ «Львівська політехніка»;
О. О. НИТРЕБИЧ, аспірант, НУ «Львівська політехніка»

АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ КРИТЕРІЇВ У МОДЕЛЯХ ОЦІНКИ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Ефективне функціонування сучасних інформаційних технологій та комп'ютерної техніки неможливо уявити без якісної розробки програмного забезпечення. У роботі представлено модель оцінювання надійності програмного забезпечення з використанням Марковських ланцюгів вищого порядку, що дозволяє врахувати взаємозалежність виконання компонент програмного продукту. Для пошуку оптимального порядку Марковського процесу, що найбільш точно відображатиме характер виконання програмного забезпечення, проаналізовані інформаційні критерії.

Ключові слова: надійність програмного забезпечення, Марковські моделі вищого порядку, індуктивне моделювання, інформаційні критерії.

Вступ. Кожна помилка в програмному продукті, що використовується в системах відповідального призначення, таких як системи управління технічними об'єктами, на тепло та атомних станціях, медичному обладнанні і т.д. може призвести до серйозних наслідків і навіть аварійних ситуацій. При цьому стислість термінів розробки, обмеженість в людських і фінансових ресурсах часто не дозволяє достатньо протестувати ПЗ, виявити критичні ділянки його роботи та досягти необхідних показників надійності програмного продукту. Тому питання визначення надійності програмної системи є надзвичайно актуальною задачею.

У загальному розроблені моделі прогнозування та оцінювання надійності програмного продукту поділяються на моделі «чорної» та «білої» скриньки [1]. Для адекватного аналізу надійності програмного забезпечення найчастіше використовуються моделі «білої скриньки», які враховують архітектуру програмного продукту та вплив надійності кожної компоненти (функціонально незалежні модулі, які можна протестувати [2]) на надійність програмної системи в цілому.

На сьогодні велика кількість робіт присвячена дослідженню моделей на

© В. С. ЯКОВИНА, Д. В. ФЕДАСЮК, О. О. НИТРЕБИЧ, 2014

основі компонентного підходу, що використовують граф потоку керування для опису архітектури комп'ютерної програми [1, 3 - 9]. У даному підході архітектуру ПЗ можна змоделювати як ланцюг Маркова з дискретним часом, неперервним часом та напівмарківським процесом [3, 4]. Пізніше кожен з моделей класифікують на поглинаючу (містить поглинаючий стан – стан, з якого система вийти не може) та непоглинаючу (не містить поглинаючих станів). Окрім того даний клас моделей можна ще поділити на композиційні та ієрархічні моделі. До композиційних відносяться моделі, які одночасно комбінують архітектуру програмного продукту та характер його помилок для обчислення надійності ПЗ. В ієрархічних моделях спочатку розв'язується архітектурна модель, а потім поведінка помилок системи додається до існуючого результату для прогнозування надійності ПЗ.

Більшість з таких моделей прогнозування надійності розглядають Марковські ланцюги першого порядку [3–5], які моделюють процес використання програмного продукту, вважаючи, що виконання компонент є незалежним, хоча це не завжди відповідає дійсності. У зв'язку зі складністю архітектури сучасних програмних засобів та множиною сценаріїв їх використання, припущення про незалежність виконання компонент такого ПЗ не завжди справджується, а тому для аналізу надійності ефективніше розглядати Марковські ланцюги вищих порядків, використання яких веде до розв'язання ряду задач, основною з яких є визначення оптимального порядку Марковської моделі.

На сьогодні питання використання Марковських ланцюгів вищих порядків для оцінювання надійності ПЗ залишається недослідженим, при цьому досвід їх використання в задачах прогнозування прогнозу погоди, вибору адекватної екологічної моделі дає підстави припустити, що вони дозволяють більш точно визначити надійність програмного продукту. Однак використання Марковських ланцюгів вищих порядків у задачах оцінювання та прогнозування надійності програмного продукту [10] вимагає розв'язку наступних основних трьох задач:

- визначення оптимального порядку Марковського ланцюга;
- визначення ймовірностей переходів між компонентами;
- обчислення надійності програмного продукту.

У даній роботі на основі модифікованої моделі Гокаля вищого порядку для оцінки надійності ПЗ, що враховує залежність виконання компонент програмної системи, а також проаналізовані критерії вибору оптимальної довжини ланцюга Марковського процесу вищого порядку у відповідності до характеристик програмного продукту, що є одним з основних завдань під час використання Марковських ланцюгів вищих порядків.

Визначення надійності на основі Марковського ланцюга вищого порядку. Застосування Марковського процесу вищого порядку дозволить більш точно оцінити надійність програмного продукту [6, 7].

Взявши за основу модель оцінки надійності програмного забезпечення Гокаля [2], надійність цілої системи обчислюється за формулою:

$$R = \prod_{i=1}^n R_i. \quad (1)$$

У свою чергу надійність кожної компоненти з використанням Марковських процесів вищого порядку обчислюється за формулою:

$$R_i = e^{-\int_0^{V_{ij...kl}t_{ij...kl}} \lambda_l(t) dt}, \quad (2)$$

Для знаходження $V_{ij...kl}$ – очікуваної кількості відвідувань компоненти i в залежності від виконання попередніх N компонент, потрібно розв'язати систему лінійних рівнянь:

$$V_{ij...kl} = q_{ij...kl} + \sum_{i=1}^{n-1} V_{ij...kl} p_{ij...kl}. \quad (3)$$

Окрім того з протоколу виконання програмного забезпечення потрібно обчислити такі числові вхідні параметри:

- $p_{ij...kl}$ – ймовірність переходу в компоненту l в залежності від виконання попередніх N компонент;
- $q_{ij...kl}$ – початковий ймовірнісний вектор;
- $t_{ij...kl}$ – час виконання компоненти l у залежності від виконання попередніх N компонент.

Для знаходження інтенсивності відмов кожної компоненти можна скористатись моделями «чорної скриньки», побудованих на основі результатів модульного тестування: модель Гоеля-Окумото, Муси, S-подібну або модель з динамічним показником складності проекту [11].

Отримавши числові значення усіх параметрів моделі, можна обчислити надійність кожної компоненти за формулою (2) та значення надійності комп'ютерної програми у цілому (1).

Критерії визначення оптимального порядку моделі. Практичне використання запропонованої моделі передбачає визначення оптимального порядку Марковського процесу, який залежатиме від програмної системи, що досліджується, тобто від об'єкту моделювання. Це є задача структурної ідентифікації, або, іншими словами, задача побудови моделей за експериментальними даними, що зводиться до формування за даними вибірки деякої множини F моделей різної структури та вибору серед них найбільш адекватної.

У загальному критерії, що використовуються при виборі оптимальної моделі, можна розділити на три основні групи [12]:

- статистичні критерії, що базуються на обчисленні величини помилки моделі на одній і тій же частині вибірки, де й оцінюються параметри, та застосуванні апарату перевірки статистичних гіпотез;
- зовнішні критерії (застосовуються в МГУА – метод групового урахування аргументів) ґрунтуються на розбитті вибірки на дві й більше частин. На одній частині – «внутрішній» – здійснюється оцінювання параметрів, на іншій – «зовнішній» – визначається прогнозна здатність моделей. Застосовуються чотири групи таких критеріїв: точності, узгодженості, балансу і динамічні;
- критерії з явним штрафом за складність (зокрема, критерії AIC, BIC та Малоуза), до помилки моделі «додають» адитивні або мультиплікативні штрафні

члени, які залежать від кількості точок у вибірці і від складності (числа параметрів) моделі.

Для визначення оптимального порядку Марковського процесу в задачах оцінки надійності програмного продукту пропонується використовувати інформаційні критерії, тобто критерії зі штрафом за складність, адже вони не є тестами перевірки гіпотези та не використовують рівень значущості [13]. У загальному інформаційні критерії базуються на методі максимальної правдоподібності та відрізняються різною функцією штрафів за складність моделі (кількість її параметрів). Якщо можна оцінити дисперсію помилок моделі, то використовується критерій Малоуза [14], в інакшому випадку критерії сімейства AIC та BIC. Ці критерії вибору моделі – інструменти порівняння відповідності реальному процесу декількох моделей на основі однакових спостережуваних даних та вибору із даної множини моделей найбільш адекватної. Ці критерії широко використовувались в ряді досліджень, присвячених задачам прогнозування погоди, вибору адекватної екологічної моделі. Слід зауважити, що якщо модель, що найбільше відповідає реальному процесу, існує, але не належить множині, з якої здійснюється вибір, дані критерії не в змозі її визначити. Разом з тим, застосування цих критеріїв в задачах моделювання надійності програмного продукту залишається недослідженим. Таким чином метою цієї роботи є дослідження використання інформаційних критеріїв для визначення оптимального порядку Марковського ланцюга, яким моделюють процес використання програмної системи в задачах аналізу її надійності.

Варто зауважити, що у теорії інформаційних критеріїв розглядувані моделі, за визначенням, є лише апроксимацією реального процесу, саме тому Джордж Бокс зробив знамениту заяву: «Всі моделі неправильні, але деякі з них корисні». Окрім того, «краща модель» для аналізу даних залежить від розміру вибірки, адже деякі властивості моделі часто можуть бути виявлені тільки під час збільшення розміру спостережуваної послідовності.

У роботі проаналізовано два класи інформаційних критеріїв – ті, що базуються на мінімізації відстані Кульбака-Лейблера (міра відхилення моделі, що розглядається, від реального об'єкту) та на факті Байеса та показано їхнє застосування для випадку моделювання використання програмного забезпечення Марковським ланцюгом вищого порядку.

Критерії, побудовані на основі мінімізації відстані Кульбака-Лейблера. Інформаційний Критерій Акаїке (AIC) використовується для вибору найбільш адекватної моделі з набору моделей та базується на розширеному методі максимальної правдоподібності, був вперше опублікований Акаїке у 1974 [15]. Модель обирається так, щоб вона мінімізувала відстань Кульбака-Лейблера, яка є несиметричною мірою віддаленості двох ймовірнісних розподілів, один з яких зазвичай є "істинним", а інший – "тестовим".

В загальному випадку такий критерій обчислюється наступним чином:

$$AIC = 2k - 2\ln(L), \quad (4)$$

де k – число параметрів в статистичній моделі, L – значення функції максимальної правдоподібності моделі.

Розглянемо програмний продукт, що складається з S компонент (функціональні одиниці, які можна тестувати незалежно одна від одної). Вважаємо, що процес використання ПЗ моделюється Марковським процесом, який містить S станів.

Через $n_{ij..kl}$ позначимо кількість переходів з компоненти i в j у залежності від перебування в попередніх компонентах ($i \rightarrow j \rightarrow \dots \rightarrow k \rightarrow l$) у спостережуваних послідовності, а $p_{ij..kl}$ – ймовірність переходів між компонентами у даній послідовності. Варто зазначити, що значення $p_{ij..kl}$ обчислюються за формулою:

$$p_{ij..kl} = n_{ij..kl} / n_{ij..k}, \quad (5)$$

де $n_{ij..k} = \sum_{l=1}^S n_{ij..kl}$.

Тоді функція правдоподібності в такому випадку для критерію AIC дорівнює:

$$L = \prod_{i,j,\dots,k,l} p_{ij..kl}^{n_{ij..kl}}. \quad (6)$$

де $i, j, \dots, k, l = \overline{1, S}$.

Якщо підставити значення функції правдоподібності нашої моделі ПЗ (6) у критерій AIC (4) та замість параметру k підставити кількість параметрів в моделі p -порядку, що містить S компонент ($k = S^p(S-1)$), то отримаємо

$$AIC(p) = 2S^p(S-1) - 2 \ln \left(\prod_{i,j,\dots,k,l} p_{ij..kl}^{n_{ij..kl}} \right). \quad (7)$$

Застосувавши відповідні перетворення з логарифмом, одержимо таку формулу

$$AIC(p) = 2S^p(S-1) - 2 \sum_{i,j,\dots,k,l} n_{ij..kl} \ln(p_{ij..kl}). \quad (8)$$

Для $p=2$ формула матиме вигляд:

$$AIC(p) = 2S^2(S-1) - 2 \sum_{i=1}^S \sum_{j=1}^S n_{ij} \ln(p_{ij}),$$

для $p=3$ формула матиме вигляд:

$$AIC(p) = 2S^3(S-1) - 2 \sum_{i=1}^S \sum_{j=1}^S \sum_{k=1}^S n_{ijk} \ln(p_{ijk})$$

і т.д.

Значення порядку p , що мінімізує критерій AIC (8) відповідає оптимальному порядку Марковського ланцюга вищого порядку для даної моделі ПЗ.

Як бачимо у формулі (8) даний критерій не залежить від величини вибірки. Тому критерій AIC застосовується у випадку великого розміру вибірки (спостережуваної послідовності), тобто коли програмне забезпечення тестується багато раз і відповідно записується послідовність виконання компонент програмного продукту.

Для того, щоб можна було застосувати даний критерій у випадку невеликої кількості тестувань ПЗ та відповідно невеликої вибірки даних, варто використовувати критерій AIC другого порядку AICc [16], що в загальному випадку має формулу:

$$AICc = -2\ln(L) + \frac{2kn}{n-k-1} = AIC + \frac{2k(k+1)}{n-k-1}, \quad (9)$$

де n – розмір спостережуваної послідовності. Очевидно, що зі збільшенням розміру вибірки AICc збігається до AIC (адже оскільки $n \gg k$, то $n-k-1 \rightarrow n$).

Для випадку пошуку оптимального порядку p Марковського ланцюга моделі використання програмного продукту, що містить S станів, за допомогою критерію AIC другого порядку, відповідно маємо формулу:

$$AICc(p) = AIC(p) + \frac{2S^p(S-1)(S^p(S-1)+1)}{n-k-1}. \quad (10)$$

Варто зазначити, що для того щоб обчислити відстань Кульбака-Лейблера для p -тої Марковської моделі, можна скористатись формулою [12]:

$$\Delta_p = AIC(p) - AIC_{\min}. \quad (11)$$

Також існують інші критерії, що базуються на мінімізації відстані Кульбака-Лейблера, але всі вони є розширенням критерію AIC (QAIC, TIC та ін.).

Байєсівські критерії. Альтернативою критерію AIC, яку запропонував Schwarz (1978) є критерій BIC [17], що включає розмір спостережуваної послідовності:

$$BIC(p) = -2\ln(L) + \ln(n)k. \quad (12)$$

BIC має інтерпретацію Баєйса, тобто цей критерій може бути розглянутий як апроксимація відношення апостеріорної ймовірності до апіорної. Не важко помітити, що цей критерій є аналогом AIC з більш строгою функцією штрафів (вже починаючи з розміру вибірки $n \geq 8$, $\ln(n)k$ перевищує $2k$).

Особливістю даного критерію є його застосування у випадку невеликого розміру спостережуваної вибірки.

Тобто у випадку використання критерію BIC для знаходження оптимального порядку Марковського процесу, що моделює процес використання програмного продукту, отримаємо формулу за аналогією із AIC:

$$BIC(p) = 2S^p(S-1)\ln(n) - 2 \sum_{i,j,\dots,k,l} n_{ij\dots kl} \ln p_{ij\dots kl}. \quad (13)$$

Одним із критеріїв, що належать до даної групи і використовується для вибірок $n \leq 20$ є HQIC [18], який ґрунтується на припущенні, що найбільш адекватною моделлю, яка відповідає реальному процесу є та, яка містить мінімальне число вільних параметрів статистичної моделі.

$$HQIC(p) = -2\ln(L) + \ln(\ln(n))k. \quad (14)$$

Однак такий критерій не знайшов широкого застосування на практиці.

Аналогічно, використовуючи цей критерій для знаходження оптимального порядку p Марковської моделі використання програмного продукту, отримаємо:

$$HQIC(p) = 2S^p(S-1)\ln(\ln(n)) - 2 \sum_{i,j,\dots,k,l} n_{ij\dots kl} \ln(P_{ij\dots kl}). \quad (15)$$

На відміну від критеріїв, що базуються на мінімізації Кульбака-Лейблера, та обирають з поміж заданої множини моделей найбільш адекватну (вважаючи, що «точної» моделі може і не бути в розглядуваній множині моделей), критерії Байєса вибирають з певною ймовірністю «правдиву» модель, і ймовірність вибору такої «точної» моделі прямує до 1, при збільшенні розміру вибірки (тобто дані критерії передбачають наявність «точної» моделі серед множини моделей, що розглядається).

Висновки. У даній роботі представлено модель оцінювання надійності програмного забезпечення із використанням Марковських ланцюгів вищого порядку. Для пошуку оптимального порядку такої моделі проведений аналіз використання інформаційних критеріїв для визначення оптимального порядку Марковського процесу та показано випадки застосування даних критеріїв залежно від кількості спостережуваних даних вибірки.

Так, при малому обсязі вибірки емпіричних даних стосовно поведінки надійності програмного продукту доцільно використовувати критерій BIC, який накладає сильніші штрафи вже при розмірі вибірки $n > 8 (\ln(n)k > 2k)$, що повинно дозволити уникнути необґрунтованого збільшення оптимального порядку моделі через недостатню кількість емпіричних даних. і збільшує імовірність вибору "точної" моделі (за умови існування останньої). За наявності великої вибірки емпіричних даних питання вибору критерію залишається відкритим і потребує експериментальних досліджень, оскільки AIC з одного боку не передбачає наявності "точної" моделі, а оцінює ступінь близькості моделей тестової множини до "істинної" моделі, з іншого боку критерій BIC при збільшенні розміру вибірки збільшує імовірність вибору "точної" моделі (за умови існування останньої).

Таким чином, зроблені висновки потребують експериментального підтвердження, тому подальші дослідження будуть присвячені практичній перевірці отриманих результатів.

Список літератури: 1. Goseva-Popstojanova, K. Comparison of architecture-based software reliability models [Text] : 12th inter. symp. / K. Goseva-Popstojanova, A. P. Mathur, K. S. Trivedi // Software Reliability Engineering. – 2001. – P. 22–31. 2. Gokhale, S. An analytical approach to architecture-based software performance reliability prediction [Text] / S. S. Gokhale, W. E. Wong, J. R. Horgan, S. Kishor // Performance Evaluation. – 2004. – Vol. 58 (4). – P. 391–412. 3. Pham, H. System Software Reliability [Text] / H. Pham // Springer series in reliability engineering, Springer-Verlag London Limited. – 2006. – 441 p. 4. Goševa-Popstojanova, K. Architecture-based approach to reliability assessment of software systems [Text] / K. Goševa-Popstojanova, S. Trivedi // Performance Evaluation 4. – 2001. – P. 179–204. 5. Krishnamurthy, S. On the estimation of reliability of a software system using reliabilities of its components [Text] : proc. of the eighth inter. symp. / S. Krishnamurthy, A. Mathur // Software Reliability Engineering. – 1997. – P. 146–155. 6. Burkhart, W. Testing Software and Systems [Text] : 23rd Ifip Wg 6.1. / W. Burkhart, Z. Fatiha // International Conference. – 2011. – 236 p. 7. Takagi, T. Accurate Usage Model Construction Using High-Order Markov Chains [Text] : proc. of 17th. inter. symp / T. Takagi, Z. Furukawa, T. Yamasaki // Software Reliability Engineering. – 2006. – P. 1-2. 8. Palviainen, M. The reliability estimation, prediction and measuring of component-based software. [Text] / M. Palviainen, A. Evesti, E. Ovaska // Journal of Systems and Software. – 2011. – Vol. 8 (6). – P.105-1070. 9. Ying Zhang. Software Reliability Test Methods Based on Component Composition and Markov Process. [Text] / Ying Zhang, Ping He //

Advanced Materials Research. – 2013. – Vol. 660. – P.169-173. **10.** Яковина., В. Використання інформаційного критерію Акаїке в задачах моделювання надійності програмного забезпечення [Текст] / В. Яковина, О. Нитребич, Д. Федасюк // Вісник Національного університету "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2012.– № 732 – С. 190–192. **11.** Чабанюк, Я. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту [Текст] / Я. М. Чабанюк, В. С. Яковина, Д. В. Федасюк, М. М. Сенів, У. Т. Хімка // Інженерія програмного забезпечення. – 2010. – №1. – С. 24–29. **12.** Степашко, В. Методы и критерии решения задач структурной идентификации [Текст] / В. С. Степашко, Ю. Л. Кочерга // Автоматика. – 1985. – №5. – С. 29-37. **13.** Burnham, P. Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach [Text] / P. Burnham, D. Anderson // Springer. – 2002. – 488 p. **14.** Mallows, C. Some comments on Cp [Text] / C.L. Mallows // Technometrics. – 1973. – V.15. – P. 661-667. **15.** Akaike, H. A new look at the statistical model identification [Text] / H. Akaike // IEEE Trans. Auto. Control. – 1974. – P. 716-723. **16.** Liew, K. The Performance of AICC as an Order Selection Criterion in ARMA Time Series Models [Text] / K. Liew, S. Mahendran S. // Pertanika J. Sci. & Technol. – 2002. – P. 25-33. **17.** Schwarz, G. Estimating the Dimension of a Model [Text] / G. Schwarz // Annals of Statistics. – 1978. – P. 461–64. **18.** Hannan, E. The determination of the order of an autoregression [Text] / E. J. Hannan, B. G. Quinn // Journal of Royal Statistical Society. – 1978. – Vol. 41. – P. 190-195.

Bibliography (transliterated): **1.** Goseva-Popstojanova, K. (2001). Comparison of architecture-based software reliability models. 12th International Symposium on Software Reliability Engineering, 22-31. **2.** Gokhale, S. S., Wong, W. E., Horgan, J. R., Kishor, S. (2004). An analytical approach to architecture-based software performance reliability prediction. Performance Evaluation, 58 (4), 391–412. **3.** Pham, H. (2006). System Software Reliability. Springer series in reliability engineering, Springer-Verlag London Limited, 441. **4.** Goševa-Popstojanova, K., Trivedi, S. (2001). Architecture-based approach to reliability assessment of software systems. Performance Evaluation 4, 179–204. **5.** Krishnamurthy, S., Mathur, A. (1997). On the estimation of reliability of a software system using reliabilities of its components. Proceedings of the Eighth International Symposium on Software Reliability Engineering, 146–155. **6.** Burkhart, W., Fatiha, Z. (2011). Testing Software and Systems. 23rd Ifip Wg 6.1 International Conference, 236. **7.** Takagi, T., Furukawa, Z., Yamasaki. T. (2006). Accurate Usage Model Construction Using High-Order Markov Chains. Supplementary Proceedings of 17th International Symposium on Software Reliability Engineering, 1–2. **8.** Palviainen, M., Evesti, A., Ovaska, E. (2011). The reliability estimation, prediction and measuring of component-based software. Journal of Systems and Software, 8 (6), 105-1070. **9.** Ying Zhang, Ping He. (2013) Software Reliability Test Methods Based on Component Composition and Markov Process. Advanced Materials Research, 660, 169-173. **10.** Yakovyna, V., Nytrebych, O., Fedasyuk, D. (2012). Usage of the information Akaike criterion in reliability modeling software problems // Journal of National University "Lviv Polytechnic". Computer Science and Information Technology, 732, 190–192. **11.** Chabanyuk, Y. M., Yakovyna, V. S., Fedasyuk, D. V., Seniv, M. M., Himka, U. T. (2010). Construction and studying software reliability models with index of the project value. Software Engineering, 1, 24–29. **12.** Stepashko, V., Kocherha, Y. (1985). Methods and criteria for solving structural identification. Automatics, 5, 29–37. **13.** Burnham, P., Anderson, D. (2002). Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer, 488. **14.** Mallows, C. (1973). Some comments on Cp. Technometrics, 15, 661–667. **15.** Akaike, H. (1974). A new look at the statistical model identification. IEEE Trans. Auto. Control., 716–723. **16.** Liew, K., Mahendran, S. (2002). The Performance of AICC as an Order Selection Criterion in ARMA Time Series Models. Pertanika J. Sci. & Technol., 25–33. **17.** Schwarz, G. (1978). Estimating the Dimension of a Model. Annals of Statistics., 461–64. **18.** Hannan, E., Quinn, B. (1978). The determination of the order of an autoregression. Journal of Royal Statistical Society, 41, 190–195.

Надійшла (received) 25.05.2014