УДК 004.052:004.413.5:519.21

Я. Чабанюк, В. Кукурба, Л. Гнатів, І. Будз, Р. Петрович
Національний університет “Львівська політехніка”,
кафедра обчислювальної математики та програмування

ОПТИМІЗАЦІЯ МОДЕЛІ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ПОКАЗНИКОМ ВЕЛИЧИНИ ПРОЕКТУ

Побудовано процедуру стохастичної оптимізації для знаходження граничних значень критерію достатності процесу тестування програмного продукту для моделі надійності програмного забезпечення з динамічним показником величини програмного проекту. Проведено дослідження та аналіз критерію, а також створеної для нього процедури за даними тестування програмного забезпечення.

Ключові слова: оптимізація тестування програмного забезпечення, неоднорідний пуассонівський процес, індекс величини проекту, критерій достатності тестування.

The procedure of stochastic optimization for finding the criterion limits the adequacy of the testing process for software reliability model with software dynamic metric size software project is based. The investigation and analysis of criteria were conducted and of procedures established for it according to software testing.

Keywords: optimization of software testing, heterogeneous Poisson process, the index value of the project, the adequacy criterion test.

1. Вступ

Швидкий розвиток програмного забезпечення (ПЗ), комп’ютерної техніки, створення нових та вдосконалення існуючих технологій побудови програмних продуктів, розширення спектра використання автоматизованих систем у сучасному світі зумовлюють паралельний розвиток усіх складових процесу побудови та впровадження ПЗ. Підвищення складності та багатокомпонентність сучасних програмних проектів вимагають спеціалізованого підходу під час створення та застосування. Сьогодні техніка, автоматизовані системи, ПЗ повинні досягати високого рівня надійності, що дозволяє їм ставати ефективним інструментом у світі нових технологій та цілей. Будь-яка неполадка може мати серйозні наслідки та втрати, що неприпустимо у сучасному світі конкуренції. Саме показник надійності стоїть першим серед не менш важливих показників – якість, живучість, безпека, готовність.

Поняття надійності ПЗ нерідко виділяють окремо [1], оскільки при застосуванні цього поняття до програмних засобів враховують особливості і відмінності цих об’єктів від стандартних(традиційних) технічних систем, для яких насамперед розробляється теорія надійності. Першочергова і фундаментальна відмінність програмних проектів від технічних засобів та систем полягає в тому, що програмний продукт не тільки не зношується з часом, що відбувається з технікою, а ще й те, що в результаті процесу використання виявляються та усуваються помилки, не кажучи про можливість модернізації шляхом розширення програми за рахунок нових модулів.

Також підвищуються вимоги до надійності та витривалості програм, виникають задачі оптимізації процесу тестування, якісного прогнозування надійності програмного продукту.

Для розв’язування подібних задач оцінювання та прогнозування сьогодні використовують моделі надійності ПЗ [1–3]. Розвиток програмних технологій та програмування загалом зумовлює потребу розвитку таких моделей. Постають задачі побудови нових моделей, вдосконалення існуючих шляхом пошуку нових параметрів, що зумовлять підвищення ступеня адекватності реальним програмним об’єктам, а також тестування таких моделей у реальних умовах.

Важливою складовою кожної моделі тестування програмного продукту є критерій достатності процесу тестування, який дає змогу керівникам проектів приймати обґрунтовані рішення про завершення цього етапу розробки.

Сьогодні у переважній більшості ІТ компаній такі показники (критерії) мають скоріше якісний та неформалізований характер, що є невиправданим під час розроблення ПЗ відпоідального призначення.

Отже, пошук та дослідження критерію достатності процесу тестування є актуальною задачею програмної інженерії.

2. Аналіз обраної для дослідження моделі надійності ПЗ з індексом величини програмного проекту

Математична модель надійності програмного забезпечення створюється для оцінювання залежності надійності програмного забезпечення від деяких параметрів, пов'язаних з модулями програми на підмножині наборів вхідних даних, за допомогою яких цей модуль контролюється [3]. До інших таких параметрів належать частота помилок, що дає змогу оцінити якість систем реального часу, що працюють в неперервному режимі, і водночас паралельно отримувати інформацію про надійність програмних продуктів [3].

Моделі основані на кількості помилок розглядаються в наступних роботах: Jelinski–Moranda [4], Schick–Wolverton [5], Shooman [6], Musa [7], Goel–Okumoto [8], Schneidewind [9] S-подібна модель зростання надійності [10], узагальнена модель негомогенного пуассонівського процесу [11] тощо.

Розглянута модель належить до класу моделей на основі кількості помилок. Припускається, що кількість виявлення помилок у моделі оцінювання та прогнозування надійності ПЗ розподілена за законом Пуассона. Крім того, вважається, що індекс величини проекту є параметром моделі та визначається на основі експериментальних даних і набуває значення з реального діапазону і завжди більший від нуля.

Функція інтенсивності виявлення несправностей для цієї моделі має вигляд:

$$l(t) = ab^{s+1}t^{s}\exp(-bt), \tag{1}$$

де a – коефіцієнт, що визначає загальну кількість помилок в ПЗ, b – коефіцієнт, що характеризує загальну тривалість процесу виявлення помилок, s – індекс величини проекту.

Для інтенсивності (1) функція кумулятивної кількості несправностей має вигляд:

$$m(t) = \int_0^t l(t)dt = a\left[-b^{s}t^{s}e^{-bt} + s\Gamma_{bt}(s)\right], \tag{2}$$

де $\Gamma_z(p) = \int_0^z t^{p-1}e^{-t}dt$, $\operatorname{Re} p > 0$, – неповна гамма-функція.

Загальна кількість помилок в ПЗ визначається кумулятивною функцією при $t \to \infty$, так

$$m(\infty) = as\Gamma(s), \tag{3}$$

де $\Gamma(s)$ – гамма-функція.

Отже, аналітичний вигляд побудованої моделі дає змогу узагальнити вираз для загальної кількості помилок в системі (3), яка залежить від величини та складності проекту і визначається параметрами моделі.

Рівняння (1) та (2) називають моделлю з індексом величини проекту [12].

Особливістю досліджуваної моделі є третій динамічний параметр, який описує індекс величини програмного проекту, що відсутній у всіх існуючих моделей.

Під величиною проекту розуміють комплексний показник, який корелює з метриками складності коду програмного продукту [13]. Встановлення залежності між індексом величини проекту та метриками складності коду є предметом подальших досліджень.

На основі проведених досліджень та аналізу індексу величини програмного проекту зроблена спроба встановлення інтервалів для характеристики програмного продукту:

при $s \in [0; 0{,}7)$ проект вважається невеликим,

при $s \in [0{,}7; 1{,}5)$ проект вважається середньої величини,

при $s \in [1{,}5; 2{,}2)$ проект вважається великим,

при $s \in [2{,}2; e]$ – дуже великим.

Проте остаточне визначення інтервалів, разом з отриманням аналітичного виразу для індексу величини проекту зі статистичною достовірністю потребує окремих досліджень із використанням множини експериментальних даних для різних типів і класів ПЗ.

Для визначення параметрів a, b, s побудованої моделі застосуємо метод максимальної правдоподібності. Спочатку для розв'язання задачі побудови функції максимальної правдоподібності $L(a,b,s)$ [12], необхідно встановити інтервали $(t_i,t_{i+1}]$, $i=\overline{0,n}$, на яких кількості помилок m_i відповідають пуассонівському розподілу [14]. $L(a,b,s)$ є диференційованою та досягає максимуму в інтервалі можливих значень.

Оцінки a, b, s отримуються шляхом розв'язання системи рівнянь:

$$\begin{cases} \dfrac{\partial L(a,b,s)}{\partial a}=0, \\ \dfrac{\partial L(a,b,s)}{\partial b}=0, \\ \dfrac{\partial L(a,b,s)}{\partial s}=0. \end{cases} \tag{4}$$

Встановлення таких властивостей точкових оцінок $\hat{a}, \hat{b}, \hat{s}$, як правдивість, ефективність, незміщеність потребує подальших досліджень.

Обрана модель має переваги над переліченими моделями цього класу. Щодо моделей надійності [7–9], то їхня функція інтенсивності строго спадає при $t>0$, що не відповідає результатам тестувань програмного забезпечення, які показують, що на початках процесу можливе різке зростання інтенсивності, а потім строге спадання. В S-подібній моделі зростання надійності [10] з'являється параметр t перед експонентою, що дає змогу змінити форму кривої, проте ця зміна не є довготривалою через сильний вплив експоненти. Що ж до узагальненої моделі негомогенного пуассонівського процесу, то в ній з'являється параметр n, що визначає складність (величину) проекту. Індекс n набуває дискретних значень $\{0,1,2,3\}$, котрі із зростанням описують збільшення складності проекту. Як зазначають автори моделі [11] її недоліками є відсутність формалізації параметра величини проекту n; його апріорне встановлення, незалежно від реальних експериментальних даних тестування ПЗ; значення параметру n з діапазону цілих чисел, що не дасть змоги описувати поведінку реального ПЗ, яке за своєю величиною і складністю має скоріше неперервний, ніж дискретний характер.

3. Опис критерію достатності процесу тестування ПЗ

Важливим прикладним аспектом моделі надійності ПЗ є встановлення кількісного критерію достатності процесу тестування програмного продукту, який дає змогу керівникам програмних проектів більш обгрунтовано приймати рішення про виділення ресурсів на тестування та про завершення цього етапу розроблення ПЗ. Так, сьогодні, у переважній більшості ІТ компаній такі показники мають скоріше якісний та неформалізований характер на зразок "задоволення замовника", які жодним чином не можна використовувати, наприклад, під час розроблення ПЗ відповідального призначення.

На противагу до параметрів a та b, залежність індексу величини проекту s (який відсутній в усіх інших моделях на основі розподілу Пуассона) виявляє чітку особливість, яку можна покласти в основу критерію достатності процесу тестування. Ця особливість полягає в тому, що при переході до пуассонового розподілу кількості виявлення помилок залежність $s(t)$ стає гладкою, а значення s наближається до постійної величини.

Таку поведінку залежності $s(t)$ можна зрозуміти, врахувавши, що індекс величини проекту (параметр s моделі) є основним параметром, що визначає форму і функцію розподілу, а відповідно і щільність ймовірності випадкової величини, яка у нашому випадку є кількістю виявлення помилки. Отже, на пізніх етапах тестування ПЗ, коли корельовані помилки виявлені та усунені, а час виявлення тих помилок, що залишились відповідає пуассоновому розподілу, якісна характеристика розподілу (параметр s) вже практично не змінюється, а змінюються переважно кількісні характеристики (параметри a та b), що дає змогу формалізувати критерій достатності процесу тестування ПЗ так:

$$\frac{ds(t)}{dt} \to 0 \tag{5}$$

Залежність функції (5) показує різку зміну форми кривої при переході експериментальних даних до пуассонового розподілу. Тобто за критерієм достатності процесу тестування можна визначити загальну кількість помилок у програмному продукті за допомогою рівняння (3) і, порівнявши її з кількістю вже виявлених та виправлених помилок, прийняти обґрунтоване рішення про розподіл ресурсів проекту зі створення програмного продукту.

4. Постановка задачі

Основним завданням даної роботи є побудова процедури стохастичної оптимізації критерію достатності процесу тестування моделі надійності програмного забезпечення з індексом величини програмного продукту, що дасть змогу визначити граничні значення цього параметра, що, своєю чергою дасть можливість отримати точніші результати застосування формули (3), визначити загальну кількість помилок; дослідити цю модель за експериментальними даними тестувань програмних продуктів, із застосуванням побудованої процедури.

5. Побудова процедури стохастичної оптимізації для критерію достатності процесу тестування

Розглянемо наступну задачу про знаходження максимуму функції інтенсивності

$$l(s,t) = \hat{a}\hat{b}^{s+1}t^s \exp\left(-\hat{b}t\right) \tag{6}$$

де $\hat{a}$ – значення параметра a, що визначає загальну кількість помилок у ПЗ після останньої тестової ітерації, $\hat{b}$ – значення параметра b після останньої тестової ітерації [12].

Функція (6) – неперервно диференційована функція, яка досягає єдиного максимуму в точці $t = t_0$.

Оскільки значення функції $l(s,t)$ змінюється з похибкою, що зумовлена значеннями $\hat{a}$, $\hat{b}$ [12], то для знаходження максимуму $l(s,t)$ не можна використати градієнтний метод. Для розв'язування такої задачі використаємо метод Кіфера – Вольфовиця [15].

Відповідно до методу Кіфера і Вольфовиця [15], потрібно виміряти значення $l(s,t)$ у двох точках $s+b(t), s-b(t)$ і визначити вираз $[l(s+b(t)) - l(s-b(t))]/(2b(t))$. Згідно з [16] отримуємо співвідношення для встановлення асимптотичних значень параметра s. Отже, маємо оптимізаційну процедуру:

$$\frac{ds(t)}{dt} = \frac{a(t)}{2b(t)} \times \left[\hat{a}\hat{b}^{\,s(t)+b(t)+1} t^{s(t)+b(t)} e^{-\hat{b}t} - \hat{a}\hat{b}^{\,s(t)-b(t)+1} t^{s(t)-b(t)} e^{-\hat{b}t} \right], \tag{7}$$

де неперервну функцію $a(t)$ необхідно бути вибрати так, щоби по-перше, послідовність $s(t)$ не зупинилась надто швидко, а по-друге, так, щоб погасити вплив випадкових похибок:

$$\int_0^\infty a(t)dt = \infty .$$

Для виконання другої умови достатньо виконання:

$$\int_0^\infty \left(\frac{a(t)}{b(t)}\right)^2 dt < \infty .$$

З врахуванням цих умов вибрано наступні функції:

$$a(t) = \frac{a}{t}, b(t) = \frac{b}{t^{\frac{1}{4}}},$$

де a, b – деякі додатні константи, наприклад $a = 1, b = 1$.

З врахуванням початкової умови значення $s(t)$ останньої тестової ітерації отримуємо задачу Коші. Розв'язком цієї задачі буде граничне значення індексу величини програмного продукту, використання цього критерію у формулі (3) дає змогу отримати точніше значення загальної кількості помилок у тестовому продукті.

6. Дослідження моделі та побудованої процедури за результатами експериментальних даних тестування ПЗ

Для дослідження було взято емпіричні дані першого та другого експериментів роботи [14].

Дослідження проводили так, щоб імітувати використання процедури стохастичної оптимізації для критерію достатності процесу тестування на етапі тестування проекту з розроблення програмного забезпечення.

Під час дослідження етап тестування з 1200 ітерацій розбивався на інтервали по 50 ітерацій. Далі після кожного завершення кожного інтервалу тестування методом Ньютона з використанням пакета Mathcad 14.0 розв'язували систему рівнянь (4) на інтервалі $(0, t_i]$ і отримували точкові оцінки параметрів a, b та s. Отримані наближені значення підставляли у вираз (2) і будували залежність $m(t)$ на інтервалі $(t_{i-1}, t_i]$.

Таблиця 1

Значення параметрів тестів [14]

t_i	1200	1200
	1-ий експеримент	2-ий експеримент
a	38,879	35,542
b	0,005217	0,005198
s	0,265	0,264
l	3,472	3,174
Виявлено помилок	35	32
$s(\infty)$	0,272	0,271
$\mu(\infty)$	35,075	32,07

Отримавши значення параметрів $\hat{a}$, $\hat{b}$ та використовуючи оптимізаційну процедуру (7), визначаються граничні значення $s(t)$, які, своєю чергою, використовувались для знаходження загальної кількості помилок за допомогою формули (3).

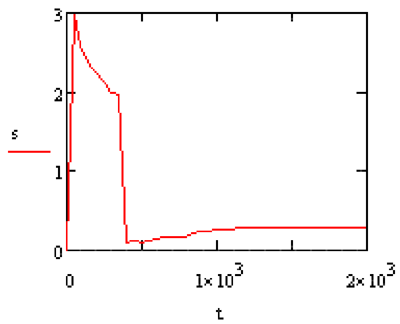

Рис. 1. s(t)

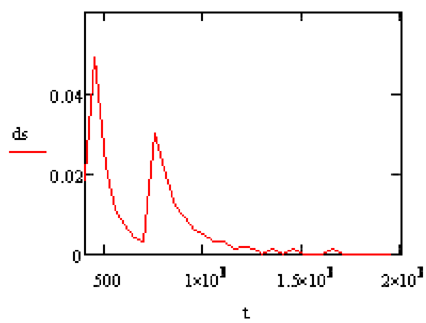

Рис. 2. ds/dt

Причому значення параметра $s(t)$ стабілізується ще до $t = 2000$ і далі не змінюється.

Також було досліджено дані, отримані з реальних тестувань програмного забезпечення. Дані тестування були отримані у вигляді виявлених помилок 5-ти типів. Розглядали окремо кожен тип помилок і отримали такі результати:

Таблиця 2

Значення параметрів реальних тестувань

Тип помилок	$s(1300)$	$s(\infty)$, за (7)	Виявлено помилок	$\mu(\infty)$, за (3)
Bloker	0,153	0,387	122	138,966
Trivial	0,154	0,252	56	64,904
Minor	0,144	0,201	479	573,506
Major	0,122	0,296	1204	1516
Critical	0,139	0,222	666	801,526

З результатів дослідження видно, що процес тестування не вийшов на фінальну стадію і його необхідно продовжити, – це випливає з відмінності між $s(1300)$ та $s(\infty)$, а також з різниці між виявленою кількістю помилок та очікуваною (3).

Варто відзначити доцільність окремого дослідження кожного типу помилок, оскільки програмний проект може спеціалізуватися на різнопланових завданнях, що можуть викликати різну складність проекту відносно типів помилок.

Важливою умовою коректності досліджуваних даних є однакова інтенсивність тестувань на однакових проміжках часу в сенсі застосування однакових програмних та людських ресурсів.

Висновок

Побудовано процедуру стохастичної оптимізації для критерію достатності процесу тестування математичної моделі надійності програмного забезпечення з динамічним показником складності програмного проекту. Створена процедура дає змогу оцінити граничне значення критерію достатності процесу тестування. В результаті можна з більшою точністю оцінити процес тестування. Окремими питаннями процедури оптимізації є встановлення умов стійкості та збіжності процедури [16, 17].

1. Половко А.М., Гуров С.В. Основы теории надежности. – СПб.: БХВ-Петербург, 2006. – 704 с. 2. Липаев В.В. Надежность программных средств. – М.: СИНТЕГ, 1998. – 232 с. 3. Тейер Т., Липов М., Нельсон Э. Надежность программного обеспечения. Пер. с англ. – М.: Мир, 1981. 4. Z. Jelinski and P. Moranda. Software reliability research // in Statistical Computer Performance Evaluation. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 465–484. 5. G.J. Schick and R.W. Wolverton Assessment of software reliability // Proc. Oper. Res. – Physica-Verlag. – Wirzberg-Wien. – 1973. – P. 395–422. 6. M.L. Shooman Probabilistic models for software reliability prediction // in Statistical Computer Performance Evaluation. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 485–502. 7. J.D. Musa A theory of software reliability and its application // IEEE Transactions on Software Engineering. – SE-1(3). – 1975. – P. 312–327. 8. A.L. Goel, K. Okumoto Time-Dependent Error-Detection Rate Model for Software and other Performance Measures // IEEE Transactions on Reliability. – Vol. R-28. – No. 3. – 1979. – P. 206–211. 9. N.F. Schneidewind Analysis of Error Process in Computer Software // Sigplan Note. – Vol. 10. – No.6. – 1975. – P.337–346. 10. S. Yamada, M. Ohba, S. Osaki S-shaped reliability growth modeling for software error detection // IEEE Transactions on Reliability. – Vol. R-32. – No.5. – 1983. – P. 475–478. 11. Тимошенко Ю.О., Дідковська М.В. Узагальнена модель негомогенного пуассонівського процесу для оцінювання надійності програного забезпечення // Проблеми програмування. – 2004. – № 2–3. – С.480–489. 12. Чабанюк Я. М., Яковина В.С., Федасюк Д.В., Сенів М.М., Хімка У.Т. Критерій достатності процесу тестування програмного продукту. // Вісник Нац.

ун-ту «Львівська політехніка» Комп'ютерні науки та інформаційні технології. №672. – 2010. – С. 346–358. 13. McCabe T.J. A complexity measure // IEEE Transactions on Software Engineering. – Vol. SE-2. – No. 4. – 1976. – P. 308–320. 14. K.-Y. Cai, D.-B. Hu, Ch.-G. Bai, H. Hu, T. Jing Does software reliability growth behavior follow a non-homogeneous Poisson process // Information and Software Technology. – Vol. 50. – 2008. – P. 1232–1247. 15. Невельсон М.Б., Хасьминский Р.З. Стохастическая аппроксимация и рекуррентное оценивание. М.:Наука, 1972. – 304 с. 16. Чабанюк Я.М., Кукурба В.Р., Чернівчан В.Я. Стохастична оптимізація в напівмарковському середовищі // Тези міжнародної наукової конференції // Проблеми стійкості та оптимізації динамічних систем детермінованої та стохастичної структури. Чернівці,2010. – С. 165. 17. Горун П.П., Кійковська О. І., Кукурба В.Р. Хімка У.Т. Стійкість стохастичної оптимізації з марковськими переключеннями // Abstracts XV International Conference Problems of decision making under uncertainties (PDMU-2010). – Lviv. May, 17–21. Ukraine.– С. 66.

УДК 519.711.2+616.222

Н. Джичка
Тернопільський національний технічний університет імені Івана Пулюя

ОБҐРУНТУВАННЯ МАТЕМАТИЧНОЇ МОДЕЛІ КОЛИВАНЬ ГОЛОСОВИХ ЗВ'ЯЗОК ЛЮДИНИ У ВИГЛЯДІ ПЕРІОДИЧНО КОРЕЛЬОВАНОГО ВИПАДКОВОГО ПРОЦЕСУ ДЛЯ ЗАДАЧІ ДІАГНОСТУВАННЯ МІКРОПОШКОДЖЕНЬ ГОЛОСОВИХ ЗВ'ЯЗОК

Розглянуто питання вибору та обґрунтування адекватної математичної моделі коливань голосових зв'язок людини для задачі діагностування мікропошкоджень голосових зв'язок. Коливання голосових зв'язок людини є випадковим процесом, в якому присутні повторюваність та випадковість, тому для їхнього опису вибрана математична модель у вигляді періодично корельованого випадкового процесу. Наведено результати перевірки коливань голосових зв'язок на стаціонарність та повторюваність.

Ключові слова: коливання голосових зв'язок, патологія голосових зв'язок, діагностування мікропошкодження голосових зв'язок, математична модель, випадковий процес, періодично корельований випадковий процес.

This article is devoted to the grounding of suitable mathematical model of human's vocal cords oscillations for the purpose of diagnostic micro damage of vocal cords. Periodically correlated stochastic process is chosen to describe the oscillations of vocal cords, because of it's repeatedness and stochastic. The article presents the results of vocal cords vibration test for stationarity and repeatability.

Keywords: vocal cord's oscillations, vocal cord's pathology, diagnostic micro damage of vocal cords, mathematical model, stochastic process, periodically correlated stochastic process.

Постановка задачі

Проблема вчасного діагностування раптового пошкодження голосу особливо важлива для людей голосо-мовних професій: лекторів, дикторів радіо та телебачення, співаків, вчителів, телефоністів, диспетчерів, акторів. Постійні ненормовані навантаження на голос викликають насамперед пошкодження голосових зв'язок, що згодом може призвести до ускладнення змикання голосової щілини, дисфонії, афонії і втрати працездатності [1].

Нормальний голос приємний на слух, має відповідний баланс ротового і носового резонансу, достатньо гучний, відповідає віку та статі, має відповідну модуляцію, включаючи висоту тону та гучність.