

Сергій Петрович Колачов
Юрій Петрович Недайбіда
Олексій Вікторович Драглюк

СТОХАСТИЧНО-ДИСКРЕТНИЙ МЕТОД ОЦІНКИ БЕЗВІДМОВНОСТІ СКЛАДНИХ ПРОГРАМНИХ КОМПЛЕКСІВ

Постановка проблеми. Аналіз останніх досліджень і публікацій

Досвід створення та застосування складних інформаційних систем виявив безліч ситуацій, при яких виникають збої та відмови їх функціонування. Це обумовлено дефектами комплексів програм, що призводить до істотних економічних втрат при їх практичному використанні (падіння марсоходу NASA, збій системи багажу аеропорту Денвера, знищення військовою ракетою цивільного літака Airbus A320 і т.д.). Застосування таких систем при низьких показниках їх надійності стає не тільки даремним, але і небезпечним, оскільки може призвести до відмови в складних адміністративних, банківських і технологічних інформаційних системах. А якщо взяти до уваги системи спеціального призначення, то тут відмова може коштувати багатьох життів.

Важливість надійності, як одного з основних показників якості програмного забезпечення масового застосування, ще більш зростає в умовах індустріального виробництва програм, коли програмне забезпечення виступає як “програмна продукція” [1].

У вітчизняній і зарубіжній нормативно-технічній документації, в довідковій літературі властивість надійності, як правило, входить в систему показників якості програмного забезпечення, що підкреслює важливість цієї властивості програмного забезпечення на сучасному етапі інформатизації суспільства. Але відсутність досконалих методів і засобів оцінки надійності програмного забезпечення стримує розвиток засобів програмного забезпечення.

Для кількісної оцінки показників надійності програмного забезпечення використовують математичні моделі, створені для оцінки залежності надійності від заздалегідь відомих або визначених в ході виконання завдання параметрів. Дані параметри, зазвичай, визначаються статистично та не враховують всієї множини факторів, що впливають на функціонування програмних комплексів.

Комбінаторний характер вхідних даних та множина умовних переходів між програмними модулями створюють величезну кількість різних

маршрутів виконання складних програмних комплексів. Така кількість маршрутів, відповідно, не може бути перевірена повністю через обмежений час налагодження та тестування програмного комплексу. Переважна більшість цих маршрутів характеризується досить малою ймовірністю виконання через низьку ймовірність поєднання вхідних даних, що надходять в програмний комплекс. Кількість маршрутів, що дійсно реалізуються, аналітично оцінити досить складно, а досвід налагодження та експлуатації складних програмних комплексів переконливо доводить неможливість перевірки усіх маршрутів при їх тестуванні [2]. Це і підкреслює актуальність обраної теми роботи.

Під надійністю розуміється кількісна властивість, що включає безвідмовність, довговічність, ремонтпридатність та збереженість системи (об’єкта) [3]. Дане визначення відноситься до технічних понять, але в дещо зміненій формі воно може бути застосоване й до програмного забезпечення.

Визначимо термін “надійність програмного забезпечення”. Прийнято вважати, що надійність – це ймовірність того, що при функціонуванні системи протягом деякого періоду часу не буде виявлено жодної помилки [4]. По своїх наслідках ці помилки далеко не однакові, тому надійність має бути визначена як функція не тільки частоти помилок, але і їх значимості, тобто надійність програмного забезпечення є функцією дії помилок на користувача системи. Найбільш коротке тлумачення терміну надійності програмного забезпечення відповідно до останнього підходу наступне. Надійність програмного забезпечення – це ймовірність безвідмовного виконання програм [5]. В даній статті під надійністю програмного забезпечення будемо розуміти його безвідмовність.

У більшості моделей оцінки надійності програмного забезпечення основна увага приділяється етапам життєвого циклу, пов’язаним з фазою розробки. Це пояснюється тим, що необхідно досягти заданого рівня надійності якомога раніше, щоб на подальших фазах життєвого циклу нести менші затрати, пов’язані з

необхідністю лише підтримувати вже досягнутий рівень надійності.

Але при такому підході не враховуються умови, сфера застосування та експлуатації програмного забезпечення, що обумовлює необхідність визначення надійності програмного забезпечення на всіх стадіях його життєвого циклу.

Формулювання мети статті. Виклад основного матеріалу

Запропонований метод числової оцінки надійності представляє програмний комплекс як стохастичний граф, а надійність модулів даного комплексу визначається дискретними методами [6].

Нехай дано програмний комплекс, що складається з М окремих програмних модулів. За структурою програмного комплексу будується стохастичний граф, що містить вершини відповідних модулів. Вершини 0 та (М+1) – фіктивні. Вершина 0 – витік, а вершина (М+1) – стік графа. Кожен програмний модуль викликається із заданою ймовірністю, виходячи з мети функціонування або значень вхідних даних.

На рисунку 1 наведено приклад стохастичного графа програмного комплексу, де вершина 0 – це витік графа, а вершина 5 – його стік. Метою є знаходження ймовірності безвідмовного вирішення задачі програмним комплексом.

Нехай P_{ij} – ймовірність переходу від і-го програмного модуля до j-го, а $P_i(t_i)$ – ймовірність безвідмовного функціонування i-го програмного модуля протягом часу t_i .

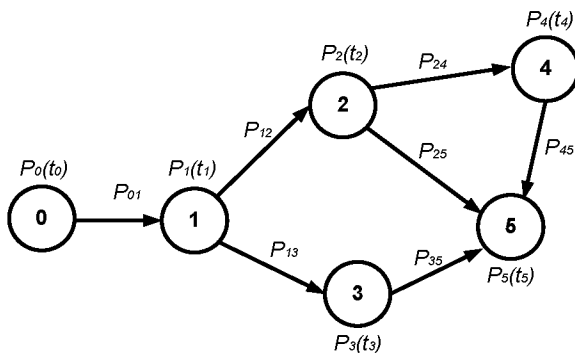


Рис. 1. Приклад стохастичного графу програмного комплексу

Згідно моделі визначення надійності Муса [6]:

$$P_i(t_i) = e^{-\frac{t_i}{\tau_i}}, \tag{1}$$

де t_i – періоду експлуатації програмного забезпечення;

τ_i – середнє напрацювання до відмови після тестування на етапі експлуатації.

$$\tau_i = \tau_0 \exp\left(\frac{CT_i}{n_i \tau_0}\right), \tag{2}$$

де τ_0 – середнє напрацювання до відмови до початку тестування модуля;

C – коефіцієнт, що враховує ущільнення тестового часу в порівнянні з часом реальної експлуатації;

T – час тестування модуля;

n – число відмов, що відбулися за час тестування модуля.

Невідомий показник τ_0 можна оцінити з наступного співвідношення:

$$\tau_0 = \frac{1}{fKN},$$

де N – початкова кількість помилок в програмному забезпеченні (вважається, що на кожні 1000 операторів, в середньому, припадає 10 помилок);

K – коефіцієнт появи помилок, що визначається емпіричним шляхом по однотипних програмах (зазвичай змінюється від $1,5 \times 10^{-7}$ до 4×10^{-7});

f – середня швидкість виконання одного оператора програми, що дорівнює відношенню середньої швидкості виконання програмного забезпечення до числа команд (операторів).

Враховуючи (2), вираз (1) отримує вид:

$$P_i(t_i) = e^{-\frac{t_i}{\tau_0} \exp\left(\frac{CT_i}{n_i \tau_0}\right)}$$

Так як вершини 0 та (М + 1) – фіктивні, то припустимо, що час перебування в них дорівнює нулю, а ймовірності безпомилкової роботи в них – одиниці.

Побудуємо матрицю $G = G(t)$, $t = t_0, t_1, \dots, t_{M+1}$, елементами якої є добутки $P_{ij} P_i(t_i)$, в яких $i, j = 0, \dots, M + 1$:

$$G = \begin{pmatrix} 0 & P_{01} & P_{02} & \dots & P_{0M} & P_{0M+1} \\ 0 & 0 & P_{12}P_1(t_1) & \dots & P_{1M}P_1(t_1) & P_{1M+1}P_1(t_1) \\ 0 & P_{21}P_2(t_2) & 0 & \dots & P_{2M}P_2(t_2) & P_{2M+1}P_2(t_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & P_{M1}P_M(t_M) & P_{M2}P_M(t_M) & \dots & 0 & P_{M+1}P_M(t_M) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Введемо поняття кроку, під яким будемо розуміти одиничний перехід від одного програмного модуля до іншого.

Нехай n – максимальне число кроків на шляху від вершини 0 до вершини (М + 1). Щоб знайти ймовірність безвідмовної роботи за два кроки, необхідно просумувати з відповідними ймовірностями добутки ймовірностей по всіх шляхах, що містить дві вершини (одна з них нульова). Це досягається зведенням матриці G в квадрат. Для отримання ймовірності безвідмовного функціонування за три кроки G зводимо в куб і так далі в залежності від n .

Побудуємо матрицю виду:

$$T = I + G(t) + G^2(t) + \dots + G^n(t),$$

де I – одинична матриця.

Якщо в графі є цикли, то n буде дорівнювати нескінченності, а матриця T буде мати вигляд:

$$T = I + G(t) + G^2(t) + \dots + G^n(t) = I(I - G(t))^{-1}.$$

Елемент матриці T з номером (0, М + 1) являє собою вираз для визначення ймовірності безвідмовної роботи всього програмного комплексу з урахуванням всіх можливих

послідовностей викликів окремих програмних модулів.

Якщо в графі є цикли, то, щоб знайти значення елемента з номером $(0, M+1)$ відповідно до правил обчислення значень елементів оберненої матриці, вираз для ймовірності безвідмовної роботи програмного комплексу можна представити у вигляді:

$$Y(t) = Q(t) / R(t),$$

де $Q(t)$ – алгебраїчне доповнення елемента з номером $(M+1, 0)$ матриці;

$R(t)$ – головний визначник матриці $(I - G(t))$.

Виконавши зазначені перетворення, одержимо шукані вирази для ймовірності безвідмовного функціонування програмного комплексу з

урахуванням усіх можливих маршрутів обчислень при наявності у графі циклів.

Висновки

Запропонований метод дозволяє робити загальну оцінку надійності складних програмних комплексів при невідомих параметрах складових програмних модулів та їх імовірнісної залежності в стохастичному графі.

Недоліками методу є те, що при його застосуванні для складних програмних комплексів зростає масштаб обчислень та не завжди можливе досить точне визначення значень необхідних параметрів для проведення розрахунків.

Література

1. Карповский Е.Я. Надежность программной продукции / Е.Я. Карповский, С.А. Чижов. – К.: Техника, 1990. – 160 с. 2. Липаев В.В. Проектирование математического обеспечения АСУ / В.В. Липаев. – М.: Советское радио, 1977. – 397 с. 3. Шохін Б.П. Проектування автоматизованих систем управління. Навчальний посібник / Б.П. Шохін. – К.: ВПІ НТУУ

“КІП”, 2003. – 160с. 4. Майерс Г. Надежность программного обеспечения / Г. Майерс. – М.: Мир, 1980. – 356 с. 5. Липаев В.В. Надежность программных средств / В.В. Липаев. – М.: СИНТЕГ, 1998. – 232 с. 6. Смагин В.А. Метод оценивания и обеспечения надежности сложных программных комплексов. – Режим доступа: http://sir35.ru/Cmagin/Method_15032.htm.

В статье предложен метод оценки безотказности сложных программных комплексов, определяющий программный комплекс как стохастический граф с дискретным определением безотказности составных программных модулей.

Ключевые слова: информационная система, программный комплекс, программный модуль, надежность программного обеспечения, показатели надежности программного обеспечения.

The method of reliability assessment of complex software systems, which determines the software system as a stochastic graph with discrete components of the definition of reliability software modules.

Key words: information system software package, software module, software reliability, software reliability indicators.