

## ИНТЕРВАЛЬНОЕ ВЫЧИСЛЕНИЕ ЭФФЕКТИВНОСТИ ТЕСТИРОВАНИЯ КОМПЬЮТЕРНЫХ ПРОГРАММ

В.Ю. Дубницкий<sup>1</sup>, А.М. Кобылин<sup>1</sup>, И.А. Супрун<sup>2</sup>

(<sup>1</sup>Харьковский банковский институт УАБД НБУ,

<sup>2</sup>Харьковский национальный университет радиоэлектроники)

*В работе предложено для оценки надежности и эффективности компьютерных программ использовать интервальные вычисления. Описан специализированный программный калькулятор для определения оценок эффективности программного обеспечения, использующий модели Мусы, гиперэкспоненциальную, Лапри. Приведены примеры решения задач.*

**интервальное вычисление, эффективность тестирования, компьютерные программы**

**Постановка задачи.** Одной из существенных задач в разработке программного обеспечения является его тестирование на наличие ошибок (так называемых багов). Это обусловлено тем, что на качество программного продукта может влиять достаточно много факторов. Они варьируют в зависимости от назначения программного продукта, сложности поставленной задачи, объемности и других причин. Очень часто их характеристики имеют высокий уровень нестохастической неопределенности. Например: неточная (не совсем точная) работа программы из-за неверного понимания технического задания, или сбои при непредусмотренных исключительных ситуациях, которые существенно влияют на работу всей программы. Их трудно прогнозировать в количественной форме. Адекватным математическим аппаратом для количественного анализа тестирования компьютерных программ в таких условиях служит аппарат интервальных вычислений.

**Анализ литературы.** В предыдущей работе авторов [1] были изложены сведения об использовании этого аппарата для оценки эффективности конверсионных банковских операций и операций с ценными бумагами. С целью повышения скорости принятия решений авторами разработан специализированный программный калькулятор, который реализует основные аксиомы интервальной арифметики, приведенные в работе [2]. В данном сообщении рассмотрено обобщение медов интервальной арифметики для интервального вычисления эффективности тестирования компьютерных программ.

**Цель исследования.** Разработка математического и программного обеспечения для определения в интервальной форме эффективности тестирования программ. Теоретические основы численного анализа этих операций и обоснование способов вычисления их эффективности приведены в работе [3].

**Интервальное оценивание эффективности модели Мусы.** Допущение модели:

- 1) модель учитывает два времени функционирования;
- 2) суммарное время функционирования  $\tau$ , которое отсчитывается в ходе разработки программы вплоть до оценки надежности;
- 3) интенсивность отказов пропорциональна числу неустранимых отказов  $N$ ;
- 4) скорость изменения числа устранимых дефектов, измеряемая относительно времени функционирования, пропорциональна интенсивности отказов;
- 5) в процессе устранения обнаруженных дефектов могут быть внесены новые дефекты;
- 6) в процессе устранения обнаруженных дефектов могут быть выявлены и устранены еще не проявившиеся дефекты;
- 7) не все дефекты могут быть устранены.

Интенсивность отказов определяется по формуле

$$z(t) = \frac{1}{T_0} \exp^{(-Ct(B_0 T_0))}, \quad (1)$$

где  $B_0$  – ожидаемое число отказов в программном обеспечении (ПО);  $T_0$  – средняя наработка на отказ в начале испытаний;  $C$  – коэффициент сжатия тестов (в простейшем случае  $C = 1$ ), учитывающий, что при тестировании дефекты в ПО проявляются чаще, чем при эксплуатации;  $t$  – время эксплуатации.

Формула будет рассчитана следующим образом, в интервальном виде расписываются все величины ( $B_0$ ,  $T_0$ ,  $C$ ,  $t$ ):

$$A_{10} := [B_{0(h)}; B_{0(b)}] * [T_{0(h)}; T_{0(b)}]; \quad A_{11} := [-1; -1] * [C_{(h)}; C_{(b)}] * [t_{(h)}; t_{(b)}];$$

$$A_{12} := A_{11} / A_{10}; \quad A_{13} := \exp^{(A_{12})}; \quad A_{14} := [1; 1] / [T_{0(h)}; T_{0(b)}]; \quad A_{15} := A_{14} * A_{13}.$$

Пример расчета приведен на рис. 1.

**Интервальное оценивание эффективности модели основной гиперэкспоненциальной.** Допущения модели:

- 1) ПО тестируется в условиях, близким к реальным условиям эксплуатации;
- 2) число дефектов, выявленных на временном интервале, пропорционально текущему числу невыявленных дефектов;

- 3) новые дефекты в процессе корректировки не вносятся; степень обнаружения дефектов равна константе на интервале между проявлениями дефектов;
- 4) степень обнаружения дефектов равна константе на интервале между проявлениями дефектов;
- 5) проявления всех дефектов происходит независимо друг от друга;
- 6) суммарное (кумулятивное) число проявлений дефектов на временном интервале подчиняется пуассоновскому процессу;
- 7) дефекты ПО делятся на K классов, для каждого из которых допущения верхних уровней применяются в отдельности.

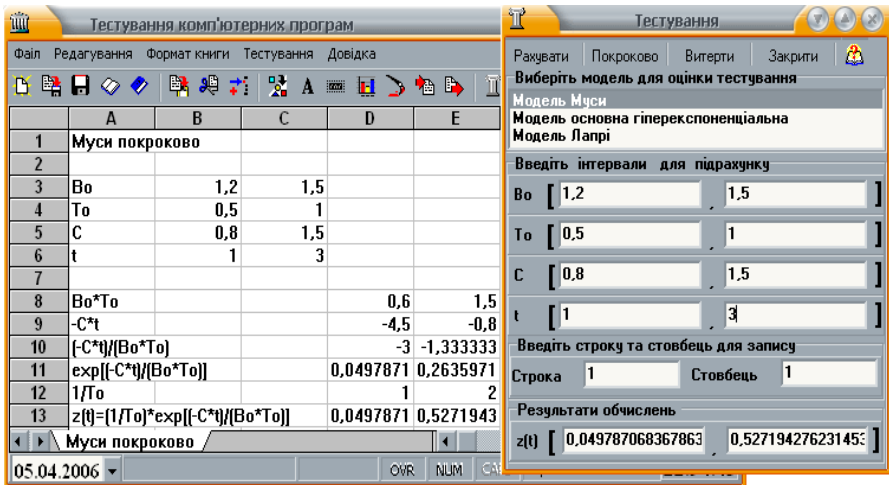


Рис. 1. Пример расчета модели Мусы

Интенсивность отказов определяется по формуле

$$\lambda(t) = a \sum_{i=1}^K p_i b_i \exp(-b_i t), \quad (2)$$

где  $p_i$  – весовой коэффициент, характеризующий интенсивность обнаружения дефектов  $i$ -го класса,  $\sum p_i = 1$ ;  $b_i$  – интенсивность удаления дефектов  $i$ -го класса;  $a$  – число дефектов, присутствующих в ПО в начале фазы тестирования;  $t$  – время эксплуатации.

Формула будет рассчитана следующим образом, в интервальном виде расписываются величины:  $p_i$ ,  $b_i$ ,  $a$ ,  $t$ ;  $k$  не расписываем, потому что это индекс:

$$A_{(i)_{16}} := [-1; -1] * [b_{i_{(h)}}; b_{i_{(b)}}] * [t_{(h)}; t_{(b)}]; \quad A_{(i)_{17}} := \exp^{(A_{(i)_{16}})};$$

$$A_{(i)18} := [p_{i(n)}; p_{i(b)}] * [b_{i(n)}; b_{i(b)}]; A_{(i)19} := A_{(i)18} * A_{(i)17};$$

$$A_{20} := \sum_{i=1}^K A_{(i)19}; A_{21} := [a_{(n)}; a_{(b)}] * A_{20};$$

Пример расчета приведен на рис. 2.

Тестування комп'ютерних програм

Файл Редагування Формат книги Тестування Довідка

	A	B	C	D	E	F	G
1	Основна гіперекспоненціальна модель покровоко						
2							
3	a	10	15	t	2	3	
4	p1	0,1	0,1	b1	0,5	1	
5	p2	0,4	0,4	b2	1	1,5	
6	p3	0,3	0,1	b3	1	1,2	
7	p4	0,1	0,3	b4	0,9	1	
8	p5	0,1	0,1	b5	1	2	
9							
10	-b1*t			-3	-1		
11	-b2*t			-4,5	-2		
12	-b3*t			-3,6	-2		
13	-b4*t			-3	-1,8		
14	-b5*t			-6	-2		
15	exp(-b1*t)			0,0497871	0,3678794		
16	exp(-b2*t)			0,011109	0,1353353		
17	exp(-b3*t)			0,0273237	0,1353353		
18	exp(-b4*t)			0,0497871	0,1652989		
19	exp(-b5*t)			0,0024788	0,1353353		
20	p1*b1			0,05	0,1		
21	p2*b2			0,4	0,6		
22	p3*b3			0,1	0,36		
23	p4*b4			0,09	0,3		
24	p5*b5			0,1	0,2		
25	p1*b1*exp(-b1*t)			0,0024894	0,0367879		
26	p2*b2*exp(-b2*t)			0,0044436	0,0812012		
27	p3*b3*exp(-b3*t)			0,0027324	0,0487207		
28	p4*b4*exp(-b4*t)			0,0044808	0,0495897		
29	p5*b5*exp(-b5*t)			0,0002479	0,0270671		
30	сума(p1*b1*exp(-b1*t))			0,014394	0,2433665		
31	a*(сума(p1*b1*exp(-b1*t)))			0,1439404	3,6504981		

Вибірть модель для оцінки тестування

Модель Муси

Модель основна гіперекспоненціальна

Модель Лапрі

Введіть інтервали для підрахунку

a [ 10 15 ]

t [ 2 3 ]

p1 [ 0,1 0,1 ]

p2 [ 0,4 0,4 ]

p3 [ 0,3 0,1 ]

p4 [ 0,1 0,3 ]

p5 [ 0,1 0,1 ]

b1 [ 0,5 1 ]

b2 [ 1 1,5 ]

b3 [ 1 1,2 ]

b4 [ 0,9 1 ]

b5 [ 1 2 ]

Введіть строку та стовбець для запису

Строка 1 Стовбець 1

Результати обчислень

v(t) [ 0,143940356491938 3,65049808707136 ]

05.04.2006

Рис. 2. Пример расчета эффективности для основной гиперэкспоненциальной модели

**Интервальное оценивание эффективности модели Лапри.** Допущения модели:

- 1) ПО тестируется в условиях, близких к реальным условиям эксплуатации;
- 2) число дефектов, выявленных на временном интервале, пропорционально текущему числу не выявленных дефектов;
- 3) степень обнаружения дефектов происходит независимо друг от друга;
- 4) новые дефекты в процессе корректировки не вносятся;
- 5) проявление всех дефектов происходит независимо друг от друга;
- 6) суммарное (кумулятивное) число проявлений дефектов на временном интервале подчиняется пуассоновскому процессу;

7) дефекты ПО делятся на два класса (серьезные и тривиальные).

Интенсивность отказов определяется по формуле

$$\lambda(t) = \frac{p_1 \xi_1 \exp(-\xi_1 t) + p_2 \xi_2 \exp(-\xi_2 t)}{p_1 \exp(-\xi_1 t) + p_2 \exp(-\xi_2 t)}, \quad (3)$$

где  $p_1, p_2, \xi_1, \xi_2$  – параметры определения надежности программы;  $t$  – время эксплуатации.

Формула будет рассчитана следующим образом, в интервальном виде расписываются все величины ( $p_1, p_2, \xi_2, t$ ):

$$A_{22} := [-1; -1] * [\xi_{1(n)}; \xi_{1(b)}] * [t_{(n)}; t_{(b)}]; A_{23} = \exp(A_{22});$$

$$A_{24} := p_1 * A_{23}; A_{25} := \xi_1 * A_{24}; A_{26} := [-1; -1] * [\xi_{2(n)}; \xi_{2(b)}] * [t_{(n)}; t_{(b)}];$$

$$A_{27} := \exp(A_{26}); A_{28} := p_2 * A_{27}; A_{29} := \xi_2 * A_{28};$$

$$A_{30} := A_{25} + A_{29}; A_{31} := A_{24} + A_{28}; A_{32} := A_{30} / A_{31};$$

Пример расчета приведен на рис. 3.

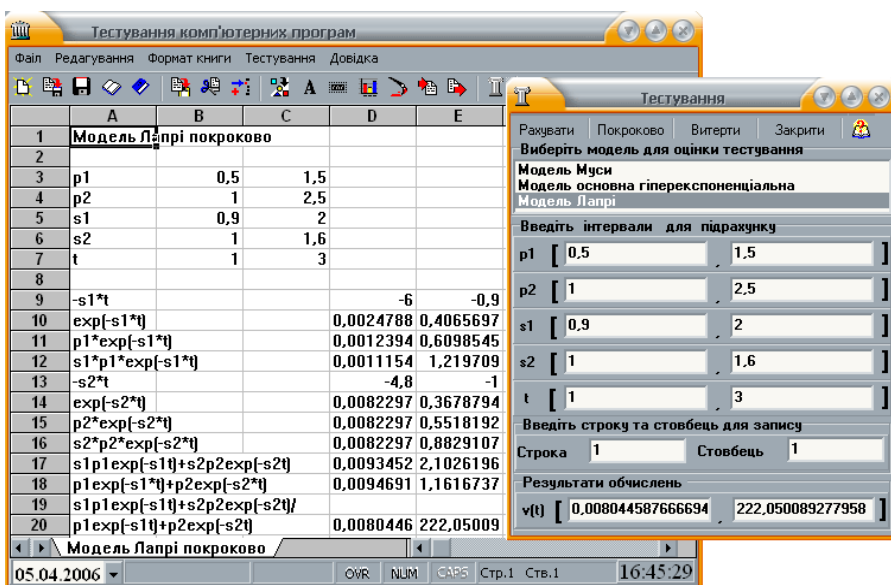


Рис. 3. Пример расчета эффективности для модели Лапри

**Реализация на ЭВМ.** Для реализации алгоритмов определения эффективности тестирования программ разработана программа в среде программирования Delphi7 с использованием элементов управления ActiveX: F1Book, ChartFx и Microsoft Visual C++ 6.0 для написания небольших COM.

В программе предусмотрена выдача сообщений на украинском языке.

Для удобства проведения подсчетов внутри программы создана отдельная форма под названием **ТЕСТИРОВАНИЕ (ТЕСТУВАННЯ)**, она служит для выбора нужного метода, ввода значений, выбора вида отчета (**ОБЫЧНЫЙ (РАХУВАТИ)**), или **ПОШАГОВЫЙ (ПОКРОКОВО)**), для определения места вывода отчета в электронной таблице (с помощью ввода строки и столбца, которые определяют верхнюю левую ячейку начала отчета), а также с помощью нею можно получить информацию о выбранном методе.

На листе электронной таблицы F1Book записываются результаты выполнения программы с маленькими комментариями. Предусмотрено отображения названия модели и вводимых значений, а при пошаговом – вывод расчетных формул.

Сначала вызывается форма для тестирования. На первом шаге осуществляется выбор нужного метода; на втором – вводятся все необходимые значения; на третьем – выбор вида отчета и его вывод в электронной таблице.

В программе предусмотрено графическое представление результатов и некоторые другие редактирующие возможности для более удобного использования.

**Выводы.** 1. Показана эффективность применения интервальных вычислений для решения задач связанных с тестирование программ в условиях, в которых применение традиционных методов прогнозирования невозможно или усложнено отсутствием сведений о статистических свойствах переменных.

2. Приведены сведения о специализированном программном калькуляторе, который реализует правила интервальной арифметики для оценивания эффективности тестирования компьютерных программ.

## ЛИТЕРАТУРА

1. Дубницкий В.Ю., Кобылин А.М., Супрун И.А. Интервальное вычисление эффективности конверсионных банковских операций и операций с ценными бумагами // БизнесИнформ. – 2005. – № 9-10. – С. 71-76.
2. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. – М.: Мир, 1987. – 259 с.
3. Методы моделирования и оценки качества и надежности программного обеспечения. / В.С. Харченко, В.В. Скляр, О.М. Тарасюк. – Учеб. пособие. – Х.: Национальный аэрокосмический университет “ХАИ”, 2004. – 159 с.

Поступила 19.05.2006

**Рецензент:** доктор технических наук, профессор Е.П. Путятин,  
Харьковский национальный университет радиоэлектроники.