

ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ АВТОМАТИЗАЦИИ ПОИСКА ВЫЧИСЛИТЕЛЬНЫХ ОШИБОК ПРОГРАММНЫХ МОДУЛЕЙ

А.Н. Коварцев, Д.С. Оплачко

Самарский государственный аэрокосмический университет им. С.П. Королева

Проблема обнаружения вычислительных ошибок

В работе рассматривается метод поиска ошибок вычислительного характера для программных продуктов, построенных на основе численного анализа и вычислительной математики, т.е. программы, используемые в различных областях науки и техники, автоматизированные системы научных исследований и т.д.

При построении таких программ в качестве «неделимых» программных единиц обычно выступают простые модули (ПМ), реализующие некие функциональные преобразования из n -мерного пространства действительных или комплексных чисел на m -мерное подпространство действительных (комплексных) чисел. В данном случае важным является то, что наблюдаемые переменные функционально зависят от «объясняющих» переменных. Размерность или масштаб компонент имеет принципиальное значение для разработки методов тестирования. Невысокая размерность ПМ обеспечивает его обозримость и возможность детального исследования.

Формально вычислительный модуль можно интерпретировать как векторную функцию $Y = F(X) = (f_1(X), f_2(X), \dots, f_m(X))^T$, действующую из множества входных параметров $X = (x_1, x_2, \dots, x_n)^T$ модуля на множество вычисляемых данных $Y = (y_1, y_2, \dots, y_m)^T$. Область определения модуля опишем как декартово произведение доменов каждого из параметров

$$\Omega_X = \Omega_{x_1} \times \Omega_{x_2} \times \dots \times \Omega_{x_n}, \quad (1)$$

где Ω_{x_k} – область изменения k -го параметра модуля.

В сложившихся условиях можно сформулировать математическое определение программной ошибки. Для каждого из выходных значений модуля введем множества допустимых значений

$$y_i \in \Omega_{y_i}, \quad (2)$$

которые в совокупности образуют множество значений ПМ

$$\Omega_F = \Omega_{y_1} \times \Omega_{y_2} \times \dots \times \Omega_{y_m}.$$

Ошибочную ситуацию определим как событие C_E

$$C_E : X \in \Omega_E, \quad \text{где } \Omega_E = \{X \mid F(X) \notin \Omega_F\}. \quad (3)$$

Ошибки условно можно разделить на «грубые», «неточности» и «редкие» в зависимости от относительного размера области ошибочных ситуаций (см. рис. 1). Чем больше соотношение объемов $\frac{V(\Omega_E)}{V(\Omega_X)}$,

тем грубее ошибка, и тем легче ее обнаружить.

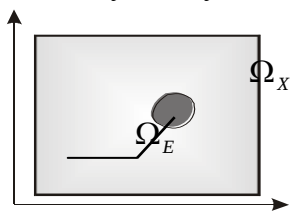


Рис. 1.

Модифицированный метод независимых испытаний

О применении метода статистических испытаний для целей тестирования программных модулей уже известно давно [2]. Сущность метода сводится к генерации случайных векторов исходных данных $\xi = (x_1, x_2, \dots, x_n)$ модуля $F(X)$, распределенных равномерно в области определения $\xi \in \Omega_X$. При условии $\Omega_E \neq \emptyset$, разыгрывая достаточно большое количество независимых векторов ξ , мы, с определенной вероятностью, попадем в область Ω_E и, тем самым, обнаружим ошибку в программе.

В каждой точке ξ вычисляются выходные параметры модуля, которые подвергаются семантическому анализу на принадлежность собственным областям значений. Для ликвидации последствий фатальных ошибок, которые возникают вследствие некорректных арифметических операций над целыми и действительными числами, используется механизм перехвата системных сообщений об ошибках, позволяющий разрешать конфликтные ситуации без аварийного завершения работы программы. Перед началом процесса тестирования программного модуля происходит переустановка стандартных действий операционной системы на случай возникновения математических ошибок. При этом в случае возникновения прерывания вычислительный процесс не прекращается, а управление передается к подсистеме ликвидации последствий математических ошибок.

Вероятность обнаружения ошибки зависит от соотношений «объемов» области ошибок и области определения модуля $P_E = \frac{V(\Omega_E)}{V(\Omega_X)}$ (см. рис. 1). При тестировании

программного модуля методом статистических испытаний, когда случайный вектор равномерно распределен в области определения исходных данных Ω_X , вероятность обнаружения ошибки будет пропорциональна числу испытаний. Пусть n – число исходных данных модуля $V(\Omega_X) = 1$. Положим число испытаний $N = 1000000$ (период современных датчиков псевдослучайных чисел), тогда минимальный размер области ошибочных ситуаций для гарантированного обнаружения ошибки при регулярном расположении точек не должен превышать $D(\Omega_E) \geq 1/\sqrt[n]{N}$, где $D(\Omega_E)$ – диаметр области ошибочных ситуаций, когда ошибка еще обнаруживается.

При выявлении ошибочных ситуаций требуется слишком большое число испытаний, превышающее период датчика псевдослучайных чисел. Поэтому для увеличения вероятности обнаружения ошибки предлагается искусственно увеличить область ошибочных ситуаций за счет уменьшения разрядной сетки представления действительных чисел. В этом случае диаметр области ошибочных ситуаций увеличивается в несколько раз. Тот же эффект можно получить, уменьшив размеры множества Ω_X .

Декомпозиция области поиска пространства независимых переменных

Пусть

$$x = 0.a_1a_2\dots a_t10^e \quad (4)$$

– нормализованное представление действительных чисел в ЭВМ с t разрядной арифметической сеткой. Здесь a_i – значащие цифры числа, $a_1 > 0$.

В пределах одного порядка на ЭВМ можно представить $\chi = 10^t - 1$ различных чисел. На плоскости в пределах одного порядка существует χ^2 различных чисел. В общем случае, в n -мерном пространстве в тех же условиях число различных чисел равно

χ^n . В этом смысле, например на плоскости в единичном квадрате действительные числа ЭВМ распределены неравномерно (см. рис. 2).

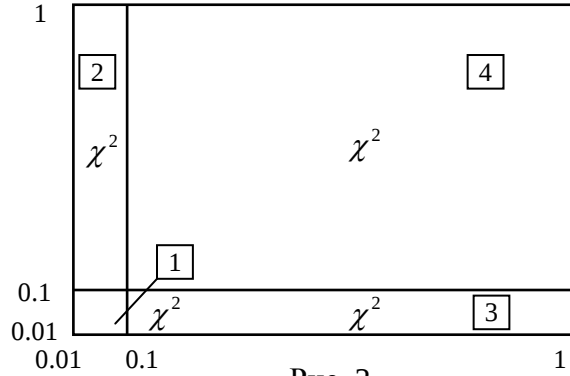


Рис. 2

Например, в зоне 4 плотность размещения точек составляет $\chi^2/0.81$, в зонах 2 и 3 плотность равна $\chi^2/0.081$, а в зоне 1 — $\chi^2/0.0081$. Из чего следует, что, если $S(\Omega_E) = 0.81/\chi^2$ и область ошибочных ситуаций попадет в 4 зону, то ошибка не будет обнаружена даже, если реализовать полный перебор всех точек, поскольку размеры области Ω_E меньше минимально возможного расстояния между соседними точками из-за ограниченности разрядной сетки ЭВМ. Вероятность обнаружения ошибочной ситуации в этом случае может быть подсчитана по формуле

$$p(\Omega_4) = p_4 \cdot p_E^4 = \frac{S(\Omega_4)}{1} \cdot \frac{S(\Omega_E)}{S(\Omega_4)} = \frac{0.81}{\chi^2},$$

где p_4 — вероятность попадания очередного случайного вектора в зону 4, p_E^4 — вероятность обнаружения ошибки в 4-й зоне.

Если область ошибочных ситуаций находится в зоне 2 или 3, то вероятность $p(\Omega_{2,3})$ можно подсчитать по формуле

$$p(\Omega_{2,3}) = p_{2,3} \cdot p_E^{2,3} = \frac{S(\Omega_{2,3})}{1} \cdot \frac{S(\Omega_E)}{S(\Omega_{2,3})} = p(\Omega_4).$$

Аналогичные результаты получаются и для зоны 1, следовательно

$$p(\Omega_1) = p(\Omega_{2,3}) = p(\Omega_4).$$

Последнее означает, что, разыгрывая случайные вектора в единичном квадрате, вероятности обнаружения ошибки в эквивалентных по степени дискретизации зонах квадрата остаются одинаковыми. Однако если в зоне 4 ошибка не может быть обнаружена даже в результате полного перебора всех точек (скрытая ошибка), то в зонах 2, 3 и 4 ошибку можно обнаружить.

Организуем автономное тестирование с 1 по 3 зоны единичного квадрата. В этом случае вероятности можно подсчитать по формулам

$$p(\Omega_{2,3}) = p_E^{2,3} = \frac{S(\Omega_E)}{S(\Omega_{2,3})} = \frac{0.81}{0.081 \cdot \chi^2} = 100 p(\Omega_4), \quad p(\Omega_1) = \frac{0.81}{0.0081 \cdot \chi^2} = 1000 p(\Omega_4).$$

Естественно, что в области с размерами в 100 и 1000 раз большие, чем $S(\Omega_E) = 0.81/\chi^2$ могут попасть вектора независимых статистических испытаний.

Полученные результаты подсказывают способ модификации метода статистических испытаний, основанный на разбиении области поиска на классы эквивалентности

$$\Omega_X = \bigcup_{\alpha} \Omega_{\alpha} \quad (\Omega_{\alpha} \cap \Omega_{\beta} = \emptyset). \quad (5)$$

В качестве отношения эквивалентности разбиения (5) выбирается условие

$$\exp(x) = \exp(x'), \quad x, x' \in \Omega_\alpha, \quad (6)$$

где $\exp(x)$ – операция взятия порядка от числа, представленного в ЭВМ в нормальной форме (4).

Как было показано выше мощности множеств классов эквивалентности в n мерном пространстве $|\Omega_{\alpha 1}| = |\Omega_{\alpha 2}| = \dots = |\Omega_{\alpha m}| = \chi^n$.

Автономное тестирование каждого из множеств эквивалентности обеспечивает равновероятное обнаружение редких ошибок в областях, пропорциональных размерам множеств Ω_α , что увеличивает общую вероятность обнаружения ошибок по сравнению с тестированием во множестве Ω_X .

Организация параллельных вычислений

Применение может быть ограничена только трудоемкостью метода решения задачи, поскольку сам метод чрезвычайно прост как в реализационном плане, так и с точки зрения требований, предъявляемым к свойствам тестируемой функции.

Существует множество способов формального определения сложности численного метода. По отношению к классу задач численной оптимизации рассмотрим наиболее универсальный подход, при котором сложность класса задач $N(\nu)$ определяется как функция параметра ν , равная минимально возможной трудоемкости метода, решающего всякую задачу с погрешностью, не превышающей ν . Трудоемкость метода определяется как число тактов работы алгоритма численного метода. В нашем случае можно определить как число обращений к исследуемой функции.

Для оценки сложности можно рассмотреть следующую модель.

Будем считать, что пространство поиска Ω_X исследовано с погрешностью ν , если в ν -окрестность ($\Omega_\nu(X)$) произвольной точки X , $X \in \Omega_X$ попала хотя бы одна точка, из числа N независимых испытаний. Если при этом $\Omega_\nu(X) \subset \Omega_E$, то будет обнаружена ошибочная ситуация.

Пусть Ω_X – единичный квадрат. Для простоты $\Omega_\nu(X)$ определим как

$$\Omega_\nu(X) = [x_1 - \nu/2, x_1 + \nu/2] \times [x_2 - \nu/2, x_2 + \nu/2] \times \dots \times [x_n - \nu/2, x_n + \nu/2].$$

Вероятность события $A: \xi \in \Omega_\nu(X)$ обозначим p_ν . При этом для единичного квадрата $p_\nu = \frac{V(\Omega_\nu(X))}{V(\Omega_X)}$, $V(\Omega_X) = 1$, $p_\nu = \nu^n$. (7)

Поставим задачу определения такого количества испытаний N , чтобы в множество $\Omega_\nu(X)$ с доверительной вероятностью δ попала хотя бы одна точка.

Пусть $v_{p_\nu}(N)$ – число возникновения события A , реализующегося в N независимых испытаниях с вероятностью p_ν , тогда условие попадание хотя бы одной точки в область $\Omega_\nu(X)$

$$P\{v_{p_\nu}(N) \geq 1\} = \delta, \text{ или}$$

$$P\{v_{p_\nu}(N) \geq 1\} = \sum_{k=1}^N C_N^k p_\nu^k (1-p_\nu)^{N-k} = 1 - P\{v_{p_\nu}(N) = 0\} = 1 - p_\nu (1-p_\nu)^N = \delta. \quad (8)$$

Из (8) с учетом (5) получим следующую оценку сложности тестирования вычислительных модулей:

$$N = \frac{\ln(1-\delta)}{\ln(1-\nu^n)}. \quad (10)$$

Если организовать двоичное деление множества исходных данных на классы эквивалентности, используя отношение (9), то формула (4) примет вид

$$N = \frac{\ln(1-\delta)}{2^n \ln(1-\nu^n)}. \quad (11)$$

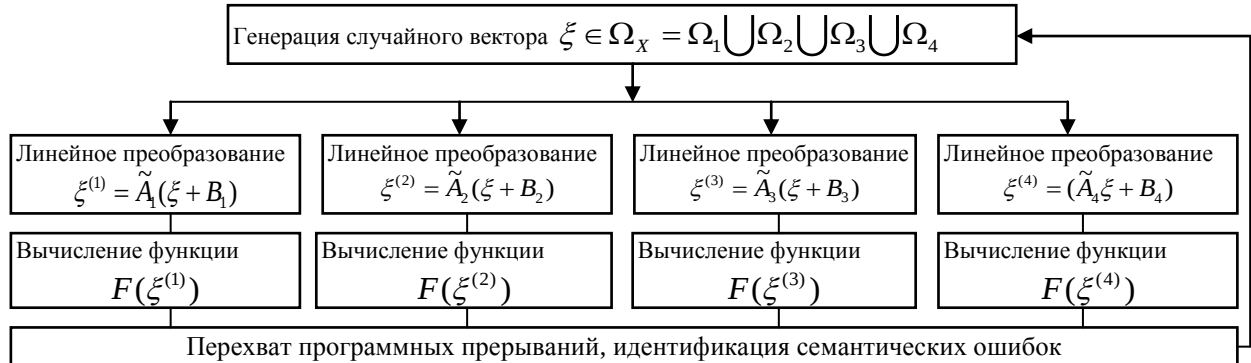


Рис. 3. Схема алгоритма параллельного тестирования вычислительных модулей

Для повышения эффективности всего алгоритма тестирования в данном случае целесообразно организовать параллельные вычисления в каждом из подмножеств области независимых переменных вычислительного модуля. На рис. 3 для двумерной области исходных данных представлена схема алгоритма параллельного тестирования.

Первоначально генерируется случайный вектор, равномерно распределенный в единичном гиперкубе (в данном случае на единичном квадрате). Данное ограничение не существенно, поскольку любой прямоугольник можно с помощью линейного преобразования превратить в единичный квадрат.

В каждом из эквивалентных множеств на параллельных процессорах, независимо друг от друга, вычисляется тестируемая функция $F(\xi^{(i)})$. Результаты вычислений анализируются в блоке перехвата программных прерываний и анализа семантических ошибок. Данный алгоритм работает циклически до тех пор, пока не будет произведено необходимое количество независимых испытаний.

Обсудим еще один важный вопрос, связанный с выбором датчика случайных чисел для алгоритма тестирования модулей. Формально для этих целей пригоден любой датчик псевдослучайных чисел, имеющий достаточно большой период. Однако для целей тестирования модулей наиболее важной характеристикой является обеспечение одинаковой плотности заполнения точками любого участка области тестирования. Таким требованиям удовлетворяют ЛП_τ последовательности, поскольку метод генерации равномерно распределенных векторов, предложенный в работе [3], гарантирует попадание в любой двоичный участок куба не менее чем $2^{\tau+1}$ точек.

Литература

1. Ермаков С. М., Михайлов Г. А. Статистическое моделирование. – 2-е изд., дополн. – М.: Наука. Главная редакция физико-математической литературы, 1982. – 296с.
2. Коварцев А. Н. Современные технологии разработки тестирования программных средств. – Самара: СГАУ, 1999. – 150с.
3. Соболев И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями. М.: Наука, 1981.