Software testing based on global search of several variables functions discontinuity

Kovartsev A.N., Popova-Kovartseva D.A., Gorshkova E.E.

Samara State Aerospace University

Abstract. Testing of software products (SP) is one of the most important stages of SP lifecycles, when it can be found not only the errors, connected to computational algorithm coding process, but also those ones input at the stages of mathematical model (MM) development. The idea of the proposed software testing algorithm, directed on detection of fatal errors in MM, is based on a global search of several variables functions discontinuity. Moreover, a heuristic characteristic function, that takes into account the peculiarities of the problem, is used, which can significantly increase the efficiency of the error situations search algorithm.

Keywords: Global search algorithm, discontinuity point, mathematical model, software module, optimization function.

Citation: Kovartsev A.N., Popova-Kovartseva D.A., Gorshkova E.E. Software testing based on global search of several variables functions discontinuity. Proceedings of Information Technology and Nanotechnology (ITNT-2015), CEUR Workshop Proceedings, 2015; 1490: 389-396. DOI: 10.18287/1613-0073-2015-1490-389-396

1. Introduction

Modern mathematical models (MM) allow to obtain extensive and highly accurate information about the processes occurring in nature and in technology by calculation. Computational experiments carried out with the mathematical model implemented as a computer program provides research shortening and its cost reduction. MM used in the calculations are finally realized in the form of software products (SP). Errors (including computational by nature errors) committed during the development of a mathematical model are directly inherited into correspondent SP [1, 2]. In this connection, it is appropriate to combine the troubleshooting procedures in mathematical models with the process of testing the respective SP.

If we consider a class of programs that are based on numerical analysis and computational mathematics, i.e., programs that implement some of mathematical models, the most difficult identifiable errors will be fatal computational errors such as "division by zero", resulting in exponent overflow and program execution failure [3]. Testing of these programs is a challenging task since the area of erroneous combinations of input data in this case is small, and the probability of accidental exposure in this area of one of the test data set points, implemented, for example, by means of stochastic methods of testing, is negligible.

At the same time, order loss or overflow errors can be efficiently traced when viewed from the perspective of solving the problem of search of second order discontinuity points and use of algorithms for global optimization (GO) of several variables functions [4].

2. Problem statement and basic definitions

Formally, the computer module can be interpreted as a vector function $Y = F(X) = (f_1(X), f_2(X), \dots, f_m(X))^T$, that acts from set of current module inputs $X = (x_1, x_2, \dots, x_n)$ to set of calculated values $Y = (y_1, y_2, \dots, y_m)^T$.

The module domain will be described as the Cartesian product of the domains of each parameter $\Omega_X = \Omega_{X1} \times \dots \times \Omega_{Xn}$, where Ω_{Xn} – the domain of the n-th module parameter. By analogy, the set of admissible values of the function $F(X)$ is presented by $\Omega_F = \Omega_{Y1} \times \dots \times \Omega_{Ym}$, where Ω_{Ym} - the range of the m-th component of the vector function.

We will define the area of error situations as the set $\Omega_E \subset \Omega_X$, wherein when $X = \Omega_E$ there is a fatal computing error in a software module $F(X)$. In this case, if $f_i(X) = \pm\infty$ then the function has a second-order discontinuity.

The measure of error situations is zero ($\mu(\Omega_E) = 0$) for second-order discontinuity points, which creates serious difficulties in finding errors of this type. These errors are classified as "rare" in [3]. A comprehensive test focused on detecting "rare" errors may require a huge number of function tests, almost exhaustive search of all values of the module source data. This approach could not be implemented within a reasonable time.

The idea of the method of the second order discontinuity points searching [3] is based on the assumption that the second-order discontinuity points for mathematical functions could be found by solving one of optimization problems:

$$\max_X f_k(X) \quad \left(\min_X f_k(X)\right), \qquad k = \overline{1,m} \tag{1}$$

Indeed, the function value at the second-order discontinuity point is designated as $\pm\infty$. It could be associated for the computer with a large maximum positive or negative number, for example, $\pm M_{sup}$, the global extreme of the function in the broadest sense of the word. Applying the methods of global optimization, we are sure to discover function second-order discontinuity points.

3. An optimization approach for solving the problem of finding the second-order discontinuity points

The most effective method among the well-known one-parameter methods of multiextremal optimization is information-statistical method of R.G. Strongin [5]. This method is based on the use of the approximate posterior probability distribution of the global extremum location, which is formed during a function testing that implements a more balanced strategy for function global minimum search. This strategy is so effective that it is often transferred from one-dimensional case to the case of several variables functions optimization.

To simplify the situation, we will consider the scalar function $f(X) = \max |f_i(X)|$.

For optimizable function we assume it to be a Lipschitz function with constant K, that means the condition $| f(x') - f(x'') | \le K | x' - x'' |$ is fulfilled.

In [5] it is shown that the function extremum search is realized by maximizing of a fairly simple characteristic function:

$$R(i) = m(x_i - x_{i-1}) + \frac{(y_i - y_{i-1})^2}{m(x_i - x_{i-1})} - 2(y_i + y_{i-1}), \qquad (2)$$

where m - the Lipschitz constant estimate, which is calculated during the function extremum search:

$$m = \begin{cases} 1 & M = 0, \\ rM & M > 0, \end{cases} \qquad M = \max_i \frac{| y_i - y_{i-1} |}{(x_i - x_{i-1})},$$

where r – parameter.

To solve the problem of finding of several variables functions discontinuity irreparable points, we will choose the characteristic function similar to (2).

From the point of solving the problem of finding of second-order discontinuity points the Strongin method didn’t manage to be effective because this method successful for Lipschitz functions, that are continuous by definition, loses its properties being applied to a class of discontinuous functions.

Let us analyze the characteristic function (2), which was deduced during information-statistical approach.

For simplicity, we assume $r = 1$. Formula (2) is presented in the following form:

$$R(i) = m\Delta x_i + \frac{\Delta y_i^{\,2}}{m\Delta x_i} - 2(y_i + y_{i-1}), \;\; \Delta x_i = (x_i - x_{i-1}), \;\; \Delta y_i = (y_i - y_{i-1}).$$

Providing that

$$m = \max_i \frac{| \Delta y_i |}{\Delta x_i} \approx \max_i | f'(x_i) | = \max_i | f_i' |,$$

we have

$$R(i) = \max| f_i' | \Delta x_i + \frac{f_i'^{\,2}}{\max| f_i' |} \Delta x_i - 2(y_i + y_{i-1}).$$

Using the first two terms of the Taylor expansion of optimizable function at the point x_{i-1} ($y_i \approx y_{i-1} + f_i' \Delta x_i$), we obtain

$$R(i) = \frac{(\max| f_i' | - f_i')^2}{\max| f_i' |} \Delta x_i - 4y_{i-1}.$$

Hense,

$$R(i) \approx \frac{(\max\left|f_i'\right| - f_i')^2}{\max\left|f_i'\right|}\Delta x_i - 4y_{i-1}. \tag{3}$$

Analysis of the formula (3) shows that the characteristic function "inclined" to post test points in the vicinity of local minima of a function to be optimized, while tending towards the global minimum, or in uncertainty intervals, the dimensions of which are large compared to other intervals of the function.

We modify the characteristic function (3), based on the following assumptions.

Firstly, the algorithm of second-order discontinuity points search, in principle, should not rely on local extrema of the function, it should react to a rapid increase or decrease in the function growth rate.

Secondly, the function itself does not contain any information about its future behavior. The first-order derivative shows the function growth rate, while the second-order derivative - acceleration of this growth. It is obvious that the second-order derivative is more informative in this sense, therefore, we will choose $(f_i'')^2$ as the first component of the characteristic function. Squaring the second-order derivative is due to the fact that we don't need information about the sign of the function curvature at the point of discontinuity.

As a measure of uncertainty of partial interval, by analogy with the Strongin method, we will choose the length of the interval, improved by certain adjustment parameter *r*. On this basis we can choose the function $R(i) = (f''(x_i))^2(\Delta x_i)^r$ as the characteristic function of the irreparable discontinuity points search method. In a multidimensional case, for example for a function of 2 variables, the second-order differential could be used instead of the second-order derivative, and a characteristic function could be represented in the form:

$$R_{(i)} = (d^2 f(X_i))^2 (\Delta_{x1}\Delta_{x2})^r. \tag{4}$$

4. Second-order discontinuity points search algorithm

Let us choose a unit square $\Pi = [0,\ 1] \times [0,\ 1]$, which is proportionally divided into four smaller squares during the fission process, as the errors searching range in computing module.

To calculate the second differential of the function by its values, calculated at the nodal points of the grid, it is easy to build a complete second-order polynomial $P_{2,2}(X)$, then $d^2 f(X) \approx d^2 P_{2,2}(X)$.

Summarizing the results, it is possible to construct an algorithm of finite-differences method (FDM) of second-order discontinuity points search:

Step 1. Regular full interpolation grid $\{X_{ij}\}\ i, j = 0,1,2$ is formed for the current square Q_k. Values of the function at the vertices of the square and in the center were known while building on the previous step of the algorithm. We should only calculate the function values at the midpoints of the edges of a square. As a result, we have $Z^{(k)} = \left\| z_{i,j}^{(k)} \right\|, \quad i, j = 0,1,2$. A source square Q_k, bounded by regular interpolation grid nodes $\{X_{ij}\}$ $i, j = 1,2,3,$ is divided into four new squares $Q_k = Q_{k+1} \cup Q_{k+2} \cup Q_{k+3} \cup Q_{k+4}$.

Step 2. Using the matrix $z^{(k)}$ an interpolation polynomial $P_{2,2}^{(k)}(X)$ is constructed for Q_k. In the centers X_{k+l}^{C} of the squares $Q_{k+l}, l = 1,2,3,4,$ the characteristic function is calculated $R(k+l) = (d^2 P_{2,2}^{(k)}(X_{k+l}^{c}))^2 (h_k)^{2r}$. Here h_k - the length of the squares side Q_{k+l} .

Step 3. Calculated values of the characteristic function are entered in the list R in descending order of the function values, i.e. ($\forall i \ R(i) \ge R(i+1)$.

Step 4. As a "decisive" item from the list R we choose one with a maximum characteristic, i.e., R (1). Thus, it is deleted from the list R.

Step 5. If algorithm stop condition is not fulfilled ($(\max_i |z_i| > M_{\sup}) \vee (\min_i h_i < \varepsilon)$), then go to step 1.

Stop condition of the algorithm described in step 5 contributes a conclusion of its work, if the test function is outside the domain of the function, which is tantamount to the appearance of the error.

5. Computational experiments

Test functions shown in Table 1 were considered to evaluate the effectiveness of the FDM algorithm. The first two test functions were constructed of known functions by their modifications. For example, the known modification of Rastrigin test function [6], which has 625 local minimum and one global in the unit square, has been converted into a function having a large number of local extrema and one "slit-shaped" second-order discontinuity.

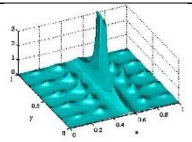
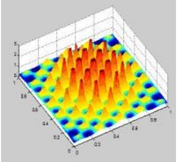
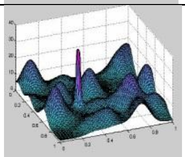
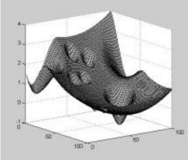
The second test function was constructed using division of one by Griewank function, which gave rise to a large number of second-order discontinuity points. The third Kovartsev test function was formed by the addition of a single second-order discontinuity point to a linear combination of error functions generating smooth function with more than 20 local extrema.

In literature the efficiency of search algorithms is typically evaluated using the operating characteristics machine [7]. Operating characteristic is a dependence of an error $P_{\alpha lg}$ detection probability on the number of iterations of the algorithm. In our case it depends on the amount of calls to the tested function $N_{f.}$

Since the second-order discontinuity points can be found in any of the global optimization algorithms the efficiency of the proposed algorithm FDM was compared with the efficiency of direct method GO, for example, the modified bisection method (BM) [8].

Operating characteristics of FDM and BM methods built for test functions 1-3 (table 1) are presented in figure 1. Operating characteristics of the FDM algorithm are denoted by solid line in the figure 1, operating characteristics of the BM algorithm - by dashed line.

Table 1. A set of test functions

Function	General view
Modified Rastrigin function: $f(x_1,x_2)=1/(10\sin^2(\pi z_1)$ $+(z_1-1)^2(1+10\sin^2(\pi z_2))$ $+(z_2-1)^2),$ $z_1=1-(11-20(x_1-a+0.55))/4;$ $z_2=1-(11-20(x_2-b+0.55))/4;$ $x_1,x_2\in[0;1].$ A set of local extrema. "Slit-shaped" second-order discontinuity.	
Modified Griewank function: $f(x_1,x_2)=1/((x_1^2+x_2^2)/400$ $-\cos(x_1)\cos(x_2/\sqrt{2})+1)$ $x_1,x_2\in[0;1].$ A set of second-order discontinuity points.	
Test Kovartsev function: $f(x_1,x_2)=\sum_{i=0}^{19}(i+1)e^{-\frac{(x_1-a_{1i})^2+(x_2-a_{2i})^2}{0.01}}$ $x_1,x_2\in[0;1].$ $+1/(1-e^{-\frac{(x_1-b_1)^2+(x_2-b_2)^2}{0.01}})$ 20 local extrema. One second-order discontinuity point.	
GKLS test functions. Continuous twice differentiable function. 10 local extrema. One global extremum. No points of discontinuity is observed.	

As we can see from the graphs, the efficiency of the proposed FDM algorithm is much greater than the efficiency of the bisection method for functions #1 and #3. It happens as bisection method is focused on the optimization of continuous functions conceptually, which forces it to make more detailed "view" of the function areas with Lipschitz constant evaluation increase. This situation arises whenever the function is calculated at points of its discontinuity. FDM method conversely is looking for areas of rapid growth of the test function.

Function 2 (Griewank) has so many points of irreparable discontinuity that their detection is easily implemented with the same efficiency using FDM and BM methods. According to Fig. 1, the BM method has a slight advantage over the FDM method. The significant advantage of FDM over BM was recorded for all other functions.

Figure 2 shows the operating characteristics of these methods, built for continuous GKLS test function.

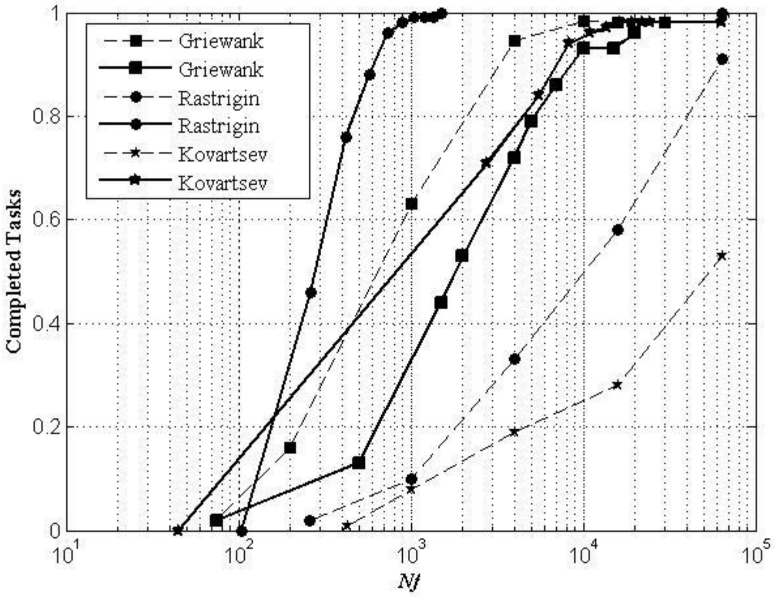


Fig. 1. – Operating characteristics of FDM and BM algorithm

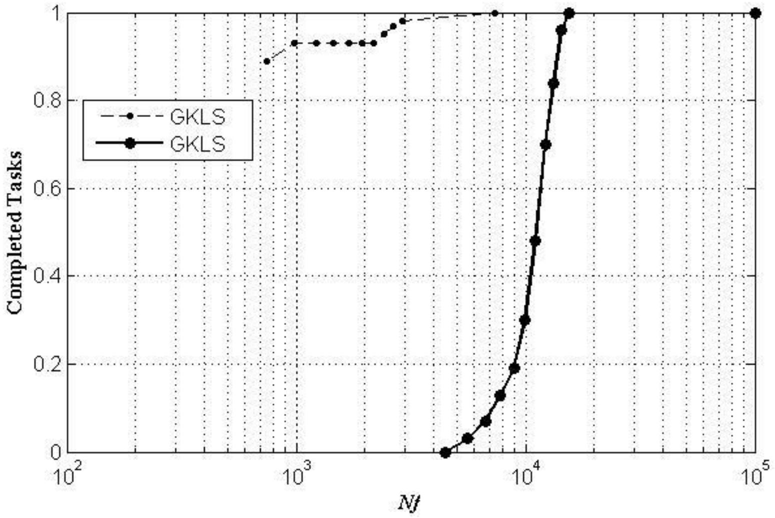


Fig. 2. – Operating characteristics of algorithms for continuous functions

It is clear from the figure, the efficiency of FDM algorithm for continuous functions is much lower than efficiency of BM algorithm. The finite difference method is forced to examine more carefully the space of the optimized variables in case of continuous functions when there are no second-order discontinuity points (test software module has no errors). In this connection, there is an idea to use the FDM and BM algorithms for computational modules testing in a combined form and stop the calculation when either the FDM finds an error or BM stops.

Conclusion

The algorithm of global search of second-order discontinuity points, designed to detect fatal errors in mathematical models of computational algorithms, is considered in this paper. The basic idea of the algorithm is the addition of a heuristic original characteristic function to a classical algorithm of global optimization, where the first is based on the analysis of the characteristic Strongin function and considers the peculiarities of the problem. It has significantly increased the efficiency of the error search algorithm. This result allows us to hope on the organization of the error search for the functions of large dimensions. In the future, we plan to develop FDM algorithm using modern supercomputer technologys.

Acknowledgements

This work was supported by the state Ministry of Education and Science of the Russian Federation as part of the activities of the Program of competitiveness improving of SSAU among the world's leading research and education centers for 2013-2020.

References

1. **Lipaev VV.** Program testing. M.: Radio and connection, 1986; 296 p. [in Russian]
2. **Kotliarov VP, Kolikova TV.** Basis of software testing. M.: BINOM, 2006; 285 p. [in Russian]
3. **Kovartsev AN.** Automation of software development and testing. Samara State Aerospace University, 1999; 150 p. [in Russian]
4. **Kovartsev AN, Popova-Kovartseva DA.** On efficiency of parallel algorithms for global optimization of functions of several variables. Computer Optics, 2011; 35(2): 256-261. [in Russian]
5. **Strongin RG.** Global optimum search. M.: Znanie, 1990. [in Russian]
6. **Abakarov ASh, Sushkov YuA.** Adaptation of random search using autocatalytic curve. SPb.: SPbGU, 2005; 67-75. [in Russian]
7. **Gergel VP, Strongin RG.** Absolute. Software system for global optimization method studying. Textbook. Nizhegorodsky University Press, 1998; 141 p. [in Russian]
8. **Kovartsev AN, Popova-Kovartseva DA, Abolmasov PV.** Efficiency study of global parallel optimization for multivariable function. Vestnik NNGU, 2013; 3(1): 252-261. [in Russian]