Postprint of article in
SCIENCE CHINA
Information Sciences

• LETTER •

Toward a K-means clustering approach to adaptive random testing for object-oriented software

Jinfu CHEN[1,2*], Minmin ZHOU[1,2], T H TSE[3], Tsong Yueh CHEN[4], Yuchi GUO[1], Rubing HUANG[1] & Chengying MAO[5]

[1] *School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang* 212000*, China;*
[2] *Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Jiangsu University, Zhenjiang* 212000*, China;*
[3] *Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong;*
[4] *Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne* 3122*, Australia;*
[5] *School of Software and Communication Engineering, Jiangxi University of Finance and Economics, Nanchang* 330013*, China*

Citation Chen J, Zhou M, Tse T H, et al. Toward a K-means clustering approach to adaptive random testing for object-oriented software. Sci China Inf Sci

Testing and debugging are mainstream methods for software quality assurance. In particular, random testing (RT, also known as fuzz testing) and partition testing (PT) are most widely adopted. Classical studies [1, 2] show that PT is only marginally better than RT in many cases but has considerably more overheads. On the other hand, RT does not consider the fact that failure-causing inputs tend to be conglomerated into regions. To address this issue, Chen et al. [3, 4] proposed adaptive random testing (ART), which targets at spreading the test cases as evenly as possible across the entire input domain. The failure-detection effectiveness is improved.

Object-oriented (OO) software has become the de facto standard in the industry. However, traditional testing methods are not immediately applicable. ART is no exception. Adaptations are needed. Ciupa et al. [5] presented the notion of object distance and proposed ART for object-oriented software (ARTOO) based on the original fixed-sized-candidate-set ART (FSCS-ART) algorithm [3]. Chen et al. [6] further proposed the object and method invocation sequence similarity (OMISS) metric for OO software, covering not only distances between objects but also between method invocation sequences. Their OMISS-ART algorithm is based on FSCS-ART as well as the max-min criterion and the forgetting strategy. In particular, the forgetting strategy [7] simply considers *some* of the already executed test cases when selecting the next test case, thus reducing the computational overhead.

In this letter, we propose an alternative approach to the selection of next test cases for OO software, applying the concept of K-means clustering [8]. Cluster analysis gathers similar data together, and has been widely used in many fields. Thus, we group test cases with similar properties into the same cluster. In other words, objects or method invocation sequences (MIS) within a cluster will be similar to one another, and objects or MIS in different clusters will be dissimilar. We then compare any new candidate with all the already executed test cases *collectively*, thus simplifying the test case selection process.

K-means clustering framework. Our framework consists of three components:

(1) In the test case selection component, we create *candidate* test cases by analyzing class diagrams. These test cases are placed in the candidate test case set until reaching its target size. Based on the results from other components, we then select one test case from the candidate set as the *next* test case for execution. After each selection of the next test case, a new candidate test set will be generated.

* Corresponding author (email: jinfuchen@ujs.edu.cn)

(2) The aim of the clustering component is to group the *executed* test cases into subsets such that a typical element in each subset can represent the whole subset. First, all the test cases are transformed into a sequence of vectors, each representing a method invocation. Euclidean distance is used to compute the dissimilarities between these transformed test cases. Then, the transformed executed test cases will be grouped using the K-means clustering algorithm into *k* clusters. A test case is randomly selected from each cluster to form a set with *k* elements, which represents all the executed test cases.

(3) In the distance measuring component, two alternative processes are proposed to transform a test case into a sequence of method invocation vectors. First, our wavelet transform (WT) is adapted from Haar wavelet transform [9]. It is based on a frequency transform that converts every MIS into a frequency vector representing the number of occurrences of each method. On the other hand, it would be inappropriate to use only WT to test MIS because it only counts the number of occurrences of each method without taking into account the order of invocation. Hence, we propose a second transformation process referred to as the trisection frequency conversion (TFC). By combining these two transformation methods, not only do we make use of the number of execution times of each method in the test case, but will also consider the invocation order. We propose three algorithms under our framework: the enhanced ART algorithm, the clustering algorithm, and the next test case selection algorithm. The first algorithm incorporates clustering analysis into ART. It groups the executed test cases and selects the next test case by calling the other two algorithms. WT and TFC are applied to gauge the dissimilarities among test cases.

Exploratory experimental evaluation. To verify the effectiveness of our approach, we compare it with OMISS-ART with forgetting, ARTOO with forgetting, and RT with MIS. We use seven open-source subject programs in C++ or C#. The effectiveness of our approach is evaluated with two measures: (a) the number of test cases executed before finding the first failure (F-measure) and (b) the time taken to find the first failure (Fm time).

The experimental results show the following: (1) For F-measure, our approach has the best performance, followed by OMISS-ART, RT, and then ARTOO; (2) Our approach uses less Fm time than OMISS-ART and RT; (3) Although our approach requires extra time for clustering, the test case selection time only amounts to a small portion of the whole testing time. Hence, we strike a good balance after considering both the test case selection time and execution time.

Conclusion. We have proposed a K-means clustering ART approach with two distance measures to test OO software. Our framework makes use of two dissimilarity measures based on WT and TFC. Our approach is different from forgetting, which considers only some of the already executed test cases when choosing the next test case. We group *all* the already executed test cases into clusters, and select a representative test case from each cluster. We then compute the distances between every candidate test case and the representative test case of each cluster in order to obtain the next test case farthest away from the executed sets. To verify the test effectiveness, we have compared our approach with three popular peer techniques in an exploratory experiment. The results show that our approach outperforms the others in terms of F-measure for all the subject programs.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U1836116, 61762040, and 61872167).

References

1 Duran J W, Ntafos S C. An evaluation of random testing. IEEE Trans Softw Eng, 1984, 10: 438–444.
2 Hamlet D, Taylor R N. Partition testing does not inspire confidence. IEEE Trans Softw Eng, 1990, 16: 1402–1411.
3 Chen T Y, Leung H, Mak I K. Adaptive random testing. In: Proceedings of Asian Computing Science Conference, Chiang Mai, Thailand, 2005. 320–329.
4 Chen T Y, Kuo F-C, Merkel R G, et al. Adaptive random testing: The ART of test case diversity. J Syst Softw, 2010, 83: 60–66.
5 Ciupa I, Leitner A, Oriol M, et al. ARTOO: Adaptive random testing for object-oriented software. In: Proceedings of International Conference on Software Engineering, Leipzig, Germany, 2008. 71–80.
6 Chen J, Kuo F-C, Chen T Y, et al. A similarity metric for the inputs of OO programs and its application in adaptive random testing. IEEE Trans Rel, 2017, 66: 373–402.
7 Chan K P, Chen T Y, Towey D. Forgetting test cases. In: Proceedings of Annual International Computer Software and Applications Conference, Chicago, IL, 2006. 485–494.
8 Hartigan J A, Wong M A. Algorithm AS 136: A K-means clustering algorithm. J Royal Stat Soc C-Appl, 1979, 28: 100–108
9 Stanković R S, Falkowski B J. The Haar wavelet transform: Its status and achievements. Comput Electr Eng, 2003, 29: 25–44.