

# Paper Review:

## Empirically Studying the Role of Selection Operators During Search-Based Test Suite Prioritization

Braden D. Licastro  
Department of Computer Science  
Allegheny College  
licastb@allegheny.edu  
<http://www.fullforceapps.com/>

January 21, 2013

### 1 Summary

This paper describes a genetic algorithm that uses coverage information to prioritize (i.e., reorder) a regression test suite [1]. It is proposed that using genetic algorithms to reorder a test suite can, on average, expose faults quicker and more reliably than using other methods. Prioritizing test suites is not a new idea, in fact it is widely used and there are many methods. This research delves further into the method of prioritization and presents a set of algorithms that increases the effectiveness of the optimized test suite. In the research article it is argued that many prioritization algorithms, though more effective than a non prioritized suite, can be improved upon. These claims were substantiated by taking said suites, running the prioritization algorithm and analysing the effectiveness of the results. It was concluded that the coverage effectiveness is greater when using the proposed genetic algorithms than it was when using the various mutations, crossover methods, or selection operators.

### 2 Critique

Throughout the research article, the test cases that were presented and tested left several questions open for discussion. Would the genetic algorithms still perform if used with a very large test suite. These suites may perform on smaller collections but when presented a large collection will the performance remain the same, degrade, or improve? Do the test cases in the research paper scale, and if not how would it be determined whether the CE rating averages are still valid in terms of performance versus existing methods.

In addition to test suite size another question arises regarding comparison of CE scores before running a test suite. The fact that the algorithms are based on a performance average (CE score), the prioritization may not always be the most effective using the method proposed. On average it will be more efficient, but in the cases where it is not it could be argued that it would be beneficial to use another method of prioritization. Would it be too inefficient to use the method proposed and running a parallel prioritization using the method with the next highest performance average? In the event of the genetic algorithm not being the most effective prioritization method, another can be used and the one that presents the highest CE score will be used for the suite to be run.

### 3 Synthesis

After reading the research paper I concluded that additional research could be performed to substantiate the claims outlined in the paper and possibly improve the effectiveness. As a future area

of research it would be beneficial to move away from the base set of programs that were generated to validate the claims outlined. I believe that these programs, although a reliable source of data, can not reliably show the whole picture. By taking a program and test suite developed by an outside developer or group of developers and running the prioritization algorithms, I believe the results could be skewed in a positive or negative outcome due to variables that were overlooked. It would also be beneficial to study the data from a program with a very large test suite and verify that the claims are still true.

Lastly an area for future research would be using several algorithms in parallel and using the prioritization with the highest CE value to run the prioritization. There may be cases where a genetic algorithm may be outperformed by an existing technology. By supplementing the proposed algorithm with another high efficiency method could increase the effectiveness of the prioritization in certain cases.

## References

- [1] Alexander P. Conrad, Robert S. Roos, and Gregory M. Kapfhammer. Empirically studying the role of selection operators during search-based test suite prioritization. In *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference*, Portland, Oregon, July 2010.