An experimental and practical study on the equivalent mutant connection: An evolutionary approach

Pedro Delgado-Pérez¹ y Francisco Chicano²

¹Universidad de Cádiz, ²Universidad de Málaga
pedro.delgado@uca.es, chicano@lcc.uma.es

Palabras Clave: Mutation testing, software testing, evolutionary computation

DOI: https://doi.org/10.1016/j.infsof.2020.106317

Resumen (Abstract).

Context: Mutation testing is considered to be a powerful approach to assess and improve the quality of test suites. However, this technique is expensive mainly because some mutants are semantically equivalent to the original program; in general, equivalent mutants require manual revision to differentiate them from useful ones, which is known as the Equivalent Mutant Problem (EMP).

Objective: In the past, several authors have proposed different techniques to individually identify certain equivalent mutants, with notable advances in the last years. In our work, by contrast, we address the EMP from a global perspective. Namely, we wonder the extent to which equivalent mutants are connected (i.e., whether they share mutation operators and code areas) as well as the extent to which the knowledge of that connection can benefit the mutant selection process. Such a study could allow going beyond the implicit limit in the traditional individual detection of equivalent mutants.

Method: We use an evolutionary algorithm to select the mutants, an approach called Evolutionary Mutation Testing (EMT). We propose a new derived version, Equivalence-Aware EMT (EA-EMT), which penalizes the fitness of known equivalent mutants so that they do not transfer their features to the next generations of mutants.

Results: In our experiments applying EMT to well-known C++ programs, we found that (i) equivalent mutants often originate from other equivalent mutants (over 60% on average); (ii) EA-EMT’s approach of penalizing known equivalent mutants provides better results than the original EMT in most of the cases (notably, the more equivalent mutants are detected, the better); and (iii) we can combine EA-EMT with Trivial Compiler Equivalence as a way to automatically identify equivalent mutants in a real situation, reaching a more stable version of EMT.

Conclusions: This novel approach opens the way for improvement in other related areas that deal with equivalent versions.