Reducing Interface Mutation Costs with Multiobjective Optimization Algorithms

Tiago Nobre, Departamento de Informática, Universidade Federal do Paraná, Curitiba, PR, Brazil
Silvia Regina Vergilio, Departamento de Informática, Universidade Federal do Paraná, Curitiba, PR, Brazil
Aurora Pozo, Departamento de Informática, Universidade Federal do Paraná, Curitiba, PR, Brazil

ABSTRACT

To reduce mutation test costs, different strategies were proposed to find a set of essential operators that generates a reduced number of mutants without decreasing the mutation score. However, the operator selection is influenced by other factors, such as: number of test data, execution time, number of revealed faults, etc. In fact this is a multiobjective problem. For that, different good solutions exist. To properly deal with this problem, a selection strategy based on multiobjective algorithms was proposed and investigated for unit testing. This work explores the use of such strategy in the integration testing phase. Three multiobjective algorithms are used and evaluated with real programs: one algorithm based on tabu search (MTabu), one based on Genetic Algorithm (NSGA-II) and the third one based on Ant Colony Optimization (PACO). The results are compared with traditional strategies and contrasted with essential operators obtained in the unit testing level.

Keywords: Ant Colony Optimization, Fault-Based Criteria, Integration Testing, Multiobjective Optimization, Evolutionary Algorithms

1. INTRODUCTION

Mutation testing is an important research topic in the software testing area. It is accomplished by generating a set $M$ of mutants for a program $p$ under test. To generate $M$, a set of mutation operators is applied. They are rules that produce single modifications in $p$, simulating errors that generally are introduced during the software development. The goal is to find a test set $T$ that is capable of distinguishing the behaviour of $p$, given by the produced output, from the mutants’ behaviour. If the output of $p$ differs from the output of $m$, $m$ is called dead. If the output of $p$ is correct, $p$ does not contain the fault described by $m$. At the end, a quality measure for $T$, called mutation score, is given by the number of dead mutants divided by the total number of generated mutants.

DOI: 10.4018/jncr.2012070102
The mutation testing presents high efficacy in terms of revealed faults, however with high computational costs, since it is necessary to execute a great number of mutants. To reduce such costs diverse strategies exist (Barbosa, Maldonado, & Vincenzi, 2001; Namin, Andrews, & Murdoch, 2008; Offutt, Lee, Rothermel, Untch, & Zapf, 1996; Vincenzi, Maldonado, Barbosa, & Delamaro, 1999). They consider that many mutants are unnecessary and redundant, mainly with respect to the mutation score, that is, many of them are easily killed. Many times they are killed by test data generated to kill other mutants and they could be discarded. Then, the idea of these strategies is to generate a lower number of mutants, without reducing the mutation score. The problem addressed is the determination of an essential set of operators, known as sufficient set problem (Offutt et al., 1996).

The sufficient set problem is related to the determination of a subset $O'$ for a set of mutant operators $O$. $O'$ generates a reduced number of mutants that are killed by a test set $T$, and $T$ is also capable to kill all the mutants generated by the complete set $O$. In this way, a reduced set of mutants (generated by the essential set of operators $O'$) is used, but the global score, with respect to $O$, is maintained.

The existing strategies are based on empirical procedures to determine a set of essential operators that maximizes the score with a minimum number of mutants. However, they do not include in their formulation other objectives to be satisfied by the test set $T$, such as: to minimize the number of test data, to maximize the number of revealed faults, and so on. They do not offer a multiobjective treatment to the problem. A more adequate approach that considers the sufficient set problem as multiobjective was introduced by Banzi, Nobre, Pinheiro, Árias, Pozo, and Vergilio (2012). The approach uses Pareto dominance concepts and obtains solutions that represent a trade-off among the objectives. Differently from the traditional strategies, it offers a set of good solutions, since many times it is not possible to determine an optimal solution considering all the factors that impact the problem. However, the approach was only used in the unit testing level and was not evaluated in the interface mutation context.

In the unit test, each module is tested individually and the mutation operators used in this level describe faults related to the implementation and logic of the programs. The interface mutation testing (Delamaro, Maldonado, & Mathur, 2001) uses operators adequate to the integration test level, which describe faults related to the communication and interaction among modules. Some traditional strategies were extended to the integration level (Vincenzi et al., 1999) and obtained good results. This fact and the results obtained by the multiobjective approach in the unit level are motivations to the work herein described that applies the multiobjective approach in the integration testing. The goal is to solve the sufficient set problem and to reduce costs of the interface mutation testing. This paper is an extension of a previous work (Nobre, Vergilio, & Pozo, 2012). Now the results are presented in a more complete and detailed way and compared with different traditional strategies. In addition to this, the set of essential interface mutation operators is constrained with the essential operator set obtained in the unit test.

The experimental evaluation was conducted with the testing tool Proteum/IM (Delamaro et al., 2001) with the same programs used in the unit level. Three algorithms, based on different optimization strategies, are evaluated to explore the search space: NSGA-II is one of the most known and used MOA based on Genetic algorithm (Ishibuchi et al., 2009); PACO is based on Ant Colony System algorithm (Dorigo & Socha, 2006), an algorithm that has shown good results for combinatorial problems like the treated here. And finally, MTabu is not evolutionary but based on a Tabu search algorithm, one of the most applied algorithms from the operations research field (Gendreau & Potvin,, 2010). The problem is the search for sets of mutation operators that maximize the mutation score and minimize the number of mutants.
Social Anti-Percolation and Negative Word of Mouth
www.igi-global.com/chapter/social-anti-percolation-negative-word/21168?camid=4v1a

Adaptive Simulated Annealing Algorithm to Solve Bio-Molecular Optimization
www.igi-global.com/chapter/adaptive-simulated-annealing-algorithm-to-solve-bio-molecular-optimization/153825?camid=4v1a