

## Introduction to Pairwise Testing. Definition and Examples

Leobardo Valera\* and Martine Ceberio. The University of Texas at El Paso

Nowadays, society relies heavily on software, which could fail for lack of thorough testing. This deficit of proper testing has resulted in big disasters in the past. For instance, this was the case from 1985 to 1987 where the Therac-25 was responsible for at least 6 accidents, when patients with cancer were given a massive overdose of radiation. Another example that is worth mentioning is that of the European spaceship, Ariane 5, which exploded 40 seconds after its launch. Testing over all parameters' values constitutes a significant cost in time and money, and can therefore at times be deemed unreasonable; sometimes, it is just impractical due to the numbers of parameters and/or the number of their values.

Empirical studies have shown that nearly 100% of the software malfunctions result from the interaction of two to six parameters, and the same studies reveal that 80% of these software malfunctions are due to only two parameters. This shifted the focus of testing from fully combinatorial to a reduced version of this – when the number of test cases would be impractical otherwise. Concretely, test suites are sought, which cover only the interaction of  $t$  parameters (combinations of any  $t$  parameters as opposed to all parameters): this is known as Combinatorial Testing, and more specifically, as the generation of  $t$ -wise covering test suites, where  $t$  is usually  $\ll$  than the number of parameters. Now, an interesting problem arising from this change in testing focus is that of determining minimal  $t$ -wise covering test suites (smallest test suites).

Although there has been work done in the area of 2-wise (known as pairwise) covering test suite generation and algorithms exist to obtain them, generating test suites covering interactions of three to six parameters would lead to a higher chance of error-free software. In this work, we focus on pairwise testing algorithms to generate minimal test suites. We describe an algorithm that allows us to know what is the minimum number of test cases necessary to cover all pairwise interactions when the parameters are binary. We have also consider and test greedy algorithms, which consists of a series of comparisons to reduce the number of test cases significantly. Finally, we draw concrete directions for future work, which will allow us to extend our results to  $t$ -wise covering test suites.

Keywords: pairwise testing, combinatorial, Galois theory, orthogonal array