Using Genetic Algorithms to Improve Test Suite Prioritizations

Alexander Conrad

Alexander Conrad

conrada@allegheny.edu

Department of Computer Science
Allegheny College
http://www.cs.allegheny.edu

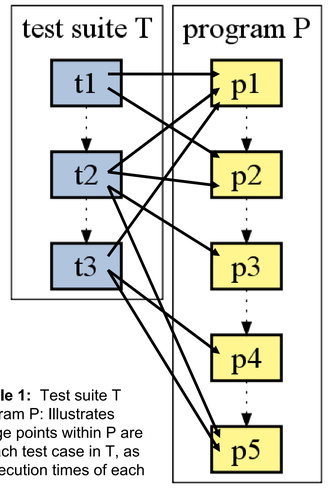
Gregory M. Kapfhammer

gkapfham@allegheny.edu

Sponsor: Harold M. State Research Fellowship

Regression Testing

- **Test Case:** A small piece of code designed to test a part of a larger program.
- **Test Suite:** A collection of test cases.
- **Regression Test Suite:** All of the test cases written for a single program.
- Regression testing is an important technique for **ensuring that modifications do not create new defects** in other parts of the program.



Testcase	Runtime	p1	p2	p3	p4	p5
t1	5	x	x			
t2	10	x	x	x		x
t3	4	x			x	x

Figure 1, Table 1: Test suite T covering program P: Illustrates which coverage points within P are covered by each test case in T, as well as the execution times of each test case.

Test Suite Prioritization

- Regression testing can be **extremely expensive -** commercial test suites are often enormous and can take weeks to run.
- **Test Suite Prioritization:** Reordering the test cases in a test suite so that requirements are covered as quickly as is possible.
- Prioritization does not reduce the amount of time necessary for the test suite to run, but **allows the suite to reveal defects earlier** than an unprioritized suite, enabling engineers to begin addressing them sooner.
- **Coverage Effectiveness (CE):** A metric that reflects how rapidly a test suite covers a set of requirements, such as segments of a program's code.

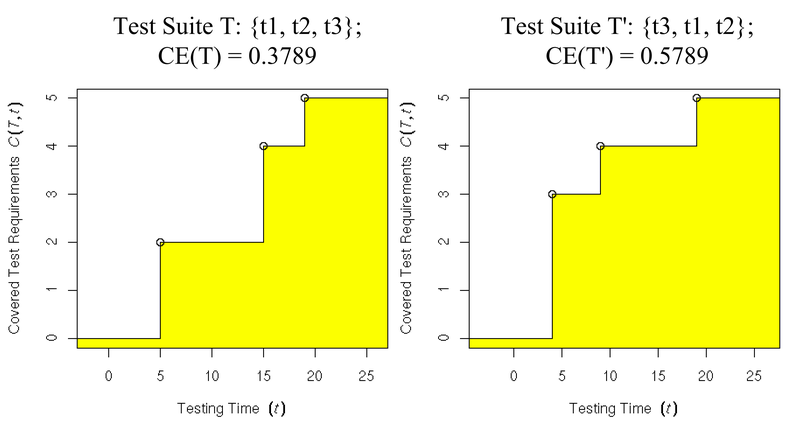


Figure 2: Example of an unprioritized test suite T and the prioritized version T'.

Genetic Algorithms

- A **genetic algorithm** is an **evolutionary** approach to solving optimization problems, imitating the biological processes of mutation and recombination.

Parts of a Genetic Algorithm

- **Chromosome:** Represents the smallest indivisible unit of data.
- **Individual:** A collection of chromosomes representing one potential solution to the problem.
- **Population:** A collection of different individuals.
- **Mutation Operator:** Randomly moves the chromosomes around within a small number of individuals.
- **Crossover Operator:** Creates new individuals by recombining patterns of chromosomes in existing individuals.
- **Fitness Function:** Ranks the quality of each individual in the population; the most "fit" individuals are selected most often for crossover.

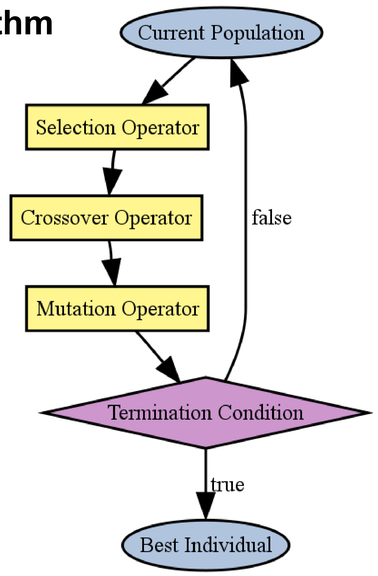


Figure 3: Simple diagram of a genetic algorithm.

Experiment Design

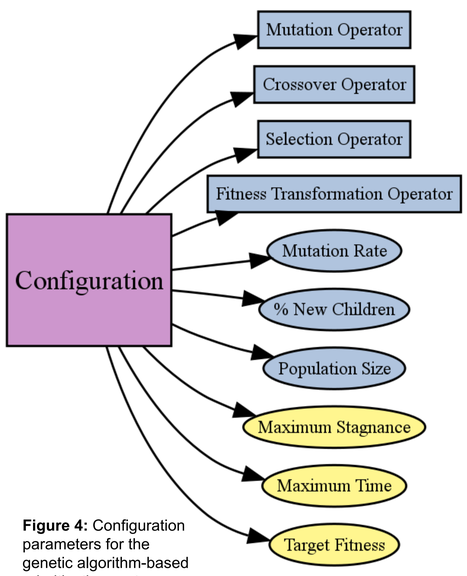


Figure 4: Configuration parameters for the genetic algorithm-based prioritization system.

- **A genetic algorithm-based prioritization system** was implemented in the **Java** programming language.
- **Coverage data** for test suites was collected from eight real-world case study applications.
- An **empirical study** was performed in which each of the case study application's test suites were prioritized.
- This experiment was repeated for **multiple configurations** (30,618 total) of the prioritizer.

Experimental Results

- The genetic algorithm produces prioritizations of **significantly higher coverage effectiveness** than the control prioritization method (a random search through the domain of potential prioritizations).
- The genetic prioritizer's **execution time is slightly greater** than that of the control method.
- Some crossover operators, particularly the POS (position based) and OX2 (order based) operators, showed promise because they rapidly produced prioritizations of noteworthy CE values.
- This research demonstrates the **viability of a genetic algorithm as a method for test suite prioritization.**

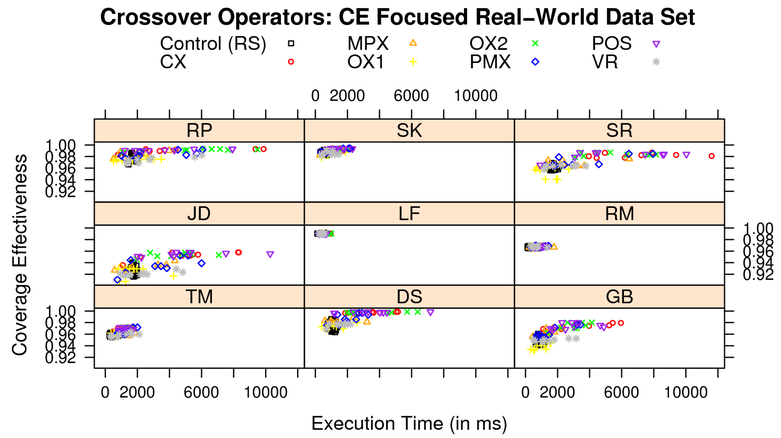


Figure 5: Quality of test suite prioritizations with respect to execution time of the prioritizer, broken down by case study application and the chosen crossover operator.

Contributions and Future Work

- This research is one of the **first** that has been done on applying genetic algorithms to the problem of regression test suite prioritization.
- The **source code** for the genetic algorithm-based test suite prioritization system will be released in the near future.
- **Senior Thesis Research:**
  - A **detailed empirical study** comparing genetic algorithm-based prioritization and greedy-based prioritization (the current industry standard).
  - **Continue development** of the genetic algorithm-based prioritizers, producing a releasable research tool which can be used by others.
  - Create a **comprehensive suite** of visualization and statistical analysis functions for the analysis of regression test suite prioritization techniques.