An Empirical Study On Using Time to First Fault As a Fitness Function for a Genetic Test Suite Prioritization Algorithm

James Devine

April 2, 2009

Time to First Fault

Identify the time it takes a test suite to detect the first fault.

Why Time to First Fault?

- Test suite reduction and prioritization does not guarantee a low time to first fault.
- An important goal in software testing is to find faults quickly.
- Once a fault is found there is reason to question the results of the preceding tests.

- Genetic algorithms seek to replicate the natural process of evolution.
- Good at providing quick approximate solutions to optimization problems.
- Can be run in parallel and using an island model approach.

Genetic Algorithm Technique

- Genetic algorithms seek to replicate the natural process of evolution.
- Good at providing quick approximate solutions to optimization problems.
- Can be run in parallel and using an island model approach.

Genetic Algorithm Technique

- Create a population of strings representing possible solutions to the problem.
- Mate strings to create child strings.
- Introduce random mutation.
- Use a fitness function to determine the fitness of new strings.

- The fitness function in a GA determines the quality of a string.
- This approach will use TTF in determining the fitness of each string.
- Can be run in parallel and using an island model approach.

- Start with a set of correct case study programs.
- Develop a test suite (automatically and user defined).
- Use muJava and Jumble to create many syntactically correct mutant versions of each program.
- The mutant programs will be used to allow for the calculation of TTF.

Hill Climbing

- Used to find local maximum and minimum.
- Starts with a random solution.
- Makes small changes to the solution.
- Terminates when no more improvements can be made.

Simulated Annealing

Hill Climbing

- Used to find local maximum and minimum.
- Starts with a random solution.
- Makes small changes to the solution.
- Terminates when no more improvements can be made.

Simulated Annealing

- Technique comes from thermal annealing.
- Tries random variations of a solution.
- Can get better solutions by trading off computation time.

- Time and space overhead will be compared for each technique.
- Calculate the effectiveness of each technique using mean TTF value for each case study.
- Compare the techniques by their effectiveness.