Regression Testing - Test case prioritization

Filip Lejhanec (50%) & Niklavs Meiers (50%)

We have implemented the Test Case Prioritization problem solver using 4 different approaches: Genetic Algorithm; Hill Climber; Random generation; NSGA-II using MOEA Framework. We have evaluated all algorithms on both the small and big datasets by running each algorithm 10 times in both cases and presenting the average result obtained including an example output. The fitness score is equal to the APFD value of the tests. The number of tests has been set as 5 for the small dataset (as per instructions) and 10 for the big dataset.

Genetic algorithm (Max Generation: 500; Population Size: 500)

Small Data Set - Average fitness score for the best candidate after 10 runs: 0.77

```
Generation: 2 New best: 0.7666666666666667[t107, t147, t166, t150, t212]
1 1 0 1 0 1 1 1 0
0 0 0 0 1 0 1 0 0
0 0 1 0 0 0 1 0 0
0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 1 0 0
```

The Genetic algorithm finds the best solution very fast (usually within 5 generations).

Big Data Set - Average fitness score for the best candidate after 10 runs: 0.61

```
Generation: 39 New best: 0.6105263157894738[t1014, t10057, t1083, t10122, t1080, t10494, t10363, t10897, t10850, t10532]
0 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0
0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 1 1 1 0 0 0 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0
```

It can be observed that the algorithm finds a good solution very fast (around 50 generations). Moreover, if the algorithm is let to run for longer it will fine-tune and produce even a little bit better solution.

Hill Climber (Candidates Generated: 10; Iterations: 25)

The algorithm considers swapping any test for any other test not currently in the set or swapping two neighbouring tests in the set as the neighbourhood (around 10000 neighbours for the big data set)

Small Data Set - Average fitness score for the best candidate after 10 runs: 0.77

```
Candidate: 0 Iteration: 3 New best: 0.7666666666666667[t125, t152, t147, t150, t101]
1 1 0 1 0 1 1 1 0
0 0 1 0 0 0 1 0 0
0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 1 0 1
1 1 0 1 0 1 1 1 0
```

The Hill Climber algorithm finds the best solution very fast (usually within 2-3 iterations).

Big Data Set - Average fitness score for the best candidate after 10 runs: 0.61

```
Candidate: 8 Iteration: 6 New best: 0.6131578947368421[t1083, t1014, t10057, t10124, t1080, t10547, t10874, t10097, t10228, t10503]
0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 1
0 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0
```

It can be observed that the algorithm finds a good solution very fast (around 5-10 iteration). The solution does not change any more after finding the local maximum. The algorithm is about 10x faster at finding a good solution than GA.

Random (Candidates Generated: 250000)

250000 candidates were generated - the equivalent of 500 population size and 500 generations for a GA algorithm.

Small Data Set - Average fitness score for the best candidate after 10 runs: 0.74

```
Best found 0.7666666666666667 After tying 3262 candidates.
1 1 0 1 0 1 1 1 0
0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 1 0 1
0 0 1 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0
```

In most iterations, the random algorithm found the optimal solution. Therefore, the difference between Random and GA algorithms for the small data set is not substantial. This is because the search space (215 * 214 * 213 * 212 * 211) is still manageable by an average computer and there are multiple best solutions.

Big Data Set - Average fitness score for the best candidate after 10 runs: 0.48

```
Best found 0.4894736842105263 4 After tying 93497 candidates.[t1042, t10371, t1035, t10595, t10370, t10543, t10871, t10327, t10035, t10777]
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 1 1 0 1
0 1 0 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0
0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0
```

The random algorithm starts to struggle when the big data set is used. This is due to the significantly bigger search space. GA performs much better than the random algorithm (APFD is 0.61 - 0.48 = 0.12 higher).

MOEA Framework NSGA-II (Max Generation: 100000; Population Size: 100)

Small Data Set - Average fitness score for the best candidate after 10 runs: 0.76

```
Fitness: -0.766667
Variable 0: 112
Variable 1: 147
Variable 2: 207
Variable 3: 150
Variable 4: 195
```

Finds the best solution for the small data set relatively easily.

Big Data Set - Average fitness score for the best candidate after 10 iterations: 0.61

```
Fitness: -0.607895
Variable 0: 147
Variable 1: 416
Variable 2: 336
Variable 3: 925
Variable 4: 67
Variable 5: 429
Variable 6: 682
Variable 7: 627
Variable 8: 945
Variable 9: 791
```

Finds a decent solution, about the same performance as our GA with regards to speed and quality of the result.

Conclusion

Overall the Hill Climber performed the best since it always found the best solution the fastest. The Random algorithm was the worst, which was expected. The GA that we implemented and the NSGA-II were very similar in all aspects.