AESIN Conference 2019

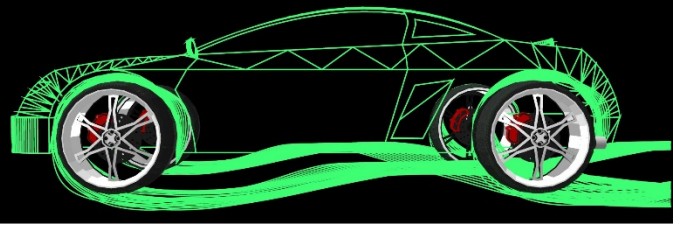
National Motorcycle Museum

01 October 2019

AESIN
AUTOMOTIVE ELECTRONICS INNOVATION

@UKAESIN | #AESIN19

www.aesinconference.com

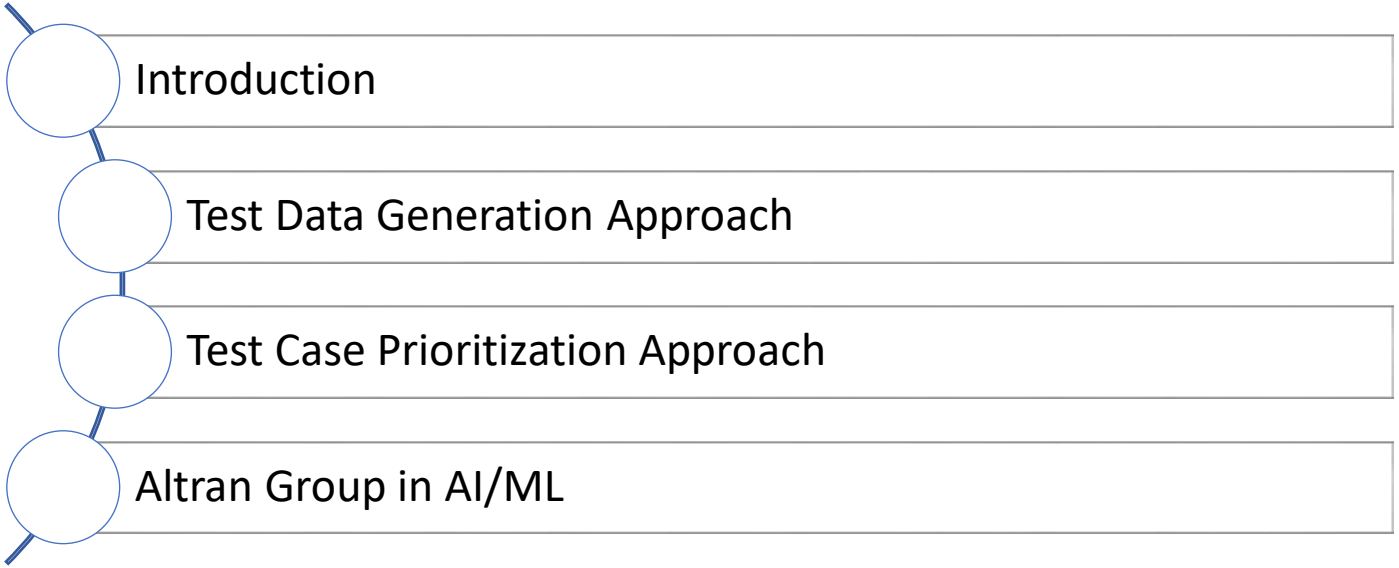


Use of AI to Automate Testing

Thomas Wilson
Gaurav Pahwa

ALTRAN

Agenda



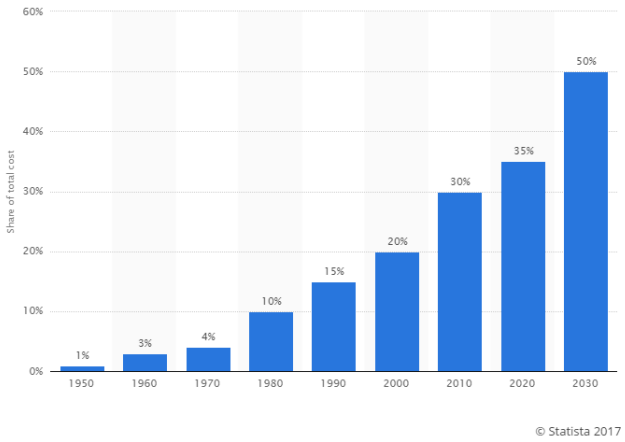
ALTRAN

01.

Introduction

ALTRAN

Market Dynamics in Automotive



By 2030 electronics components are predicted to form 50% of total costs of cars.

Testing of these components would consume more than 50% of the development costs.

A. Morozov, K. Ding, T. Chen, and K. Janschek, “Test suite prioritization for efficient regression testing of model-based automotive software,” in 2017 International Conference on Software Analysis, Testing and Evolution (SATE), Nov 2017, pp. 20–29

ALTRAN

AI-based testing techniques are applicable to all subsystems of vehicle

ADAS

- Adaptive cruise control
- Collision avoidance
- Lane support
- Surround view park assist

Body Electronics

- Body Control modules
- Climate control
- HVAC(Heating, Ventilation and Air Conditioning)
- Reliability Testing of Bus systems (CAN, FlexRay etc.)

In Vehicle Infotainment

- Radio
- HMI Testing
- Navigation support and testing
- Integrated cell phone support
- In-car video

Powertrain

- ECU, Transmission control
- Engine & Powertrain Noise Testing
- Turbocharger rev noise
- Gearbox, axle, gear chattering etc.
- Sound quality of powertrain
- Engine and powertrain mount
- Power train durability testing
- Powertrain drivability testing

Chassis & Safety

- Braking(ABS)
- Electronic Stability Program(ESP)
- Tyre Pressure Monitoring System(TPMS)
- Passive Safety (Air bag control, crash sensors)

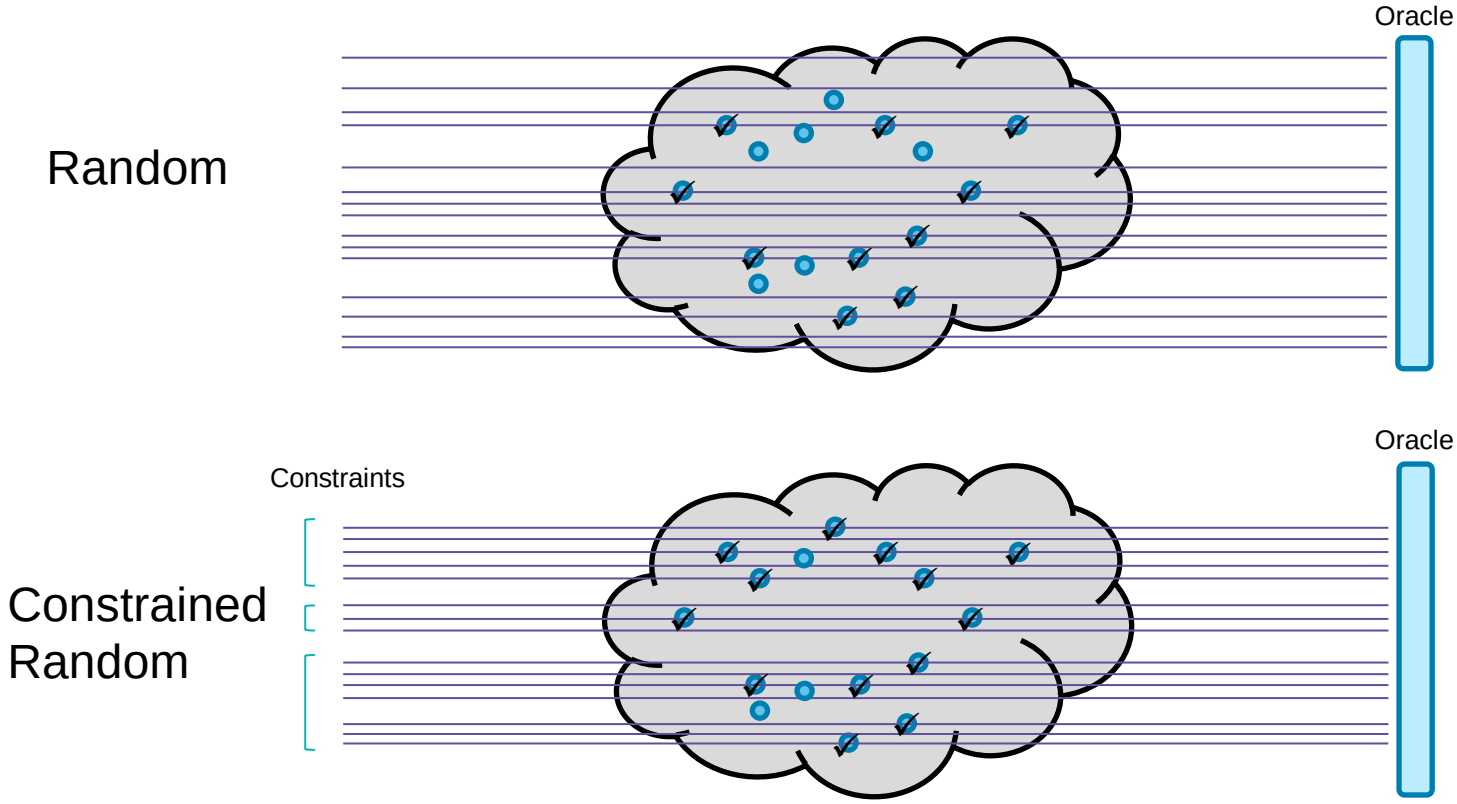
Instrument Cluster

- Vehicle and engine status

02.

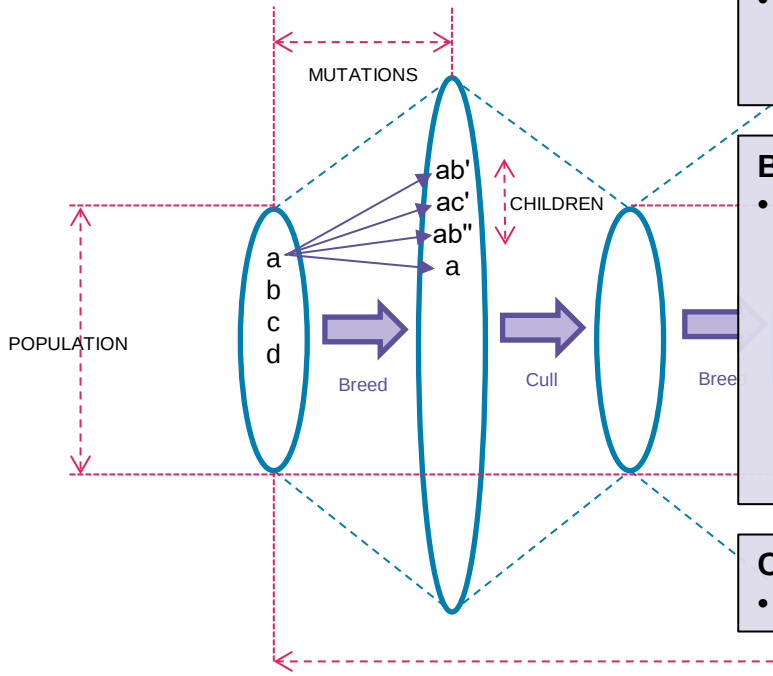
Test Data Generation Approach

ConTestor and Constrained Random Test Input Generation



ALTRAN

Genetic Algorithm for Test Input Generation – Concept



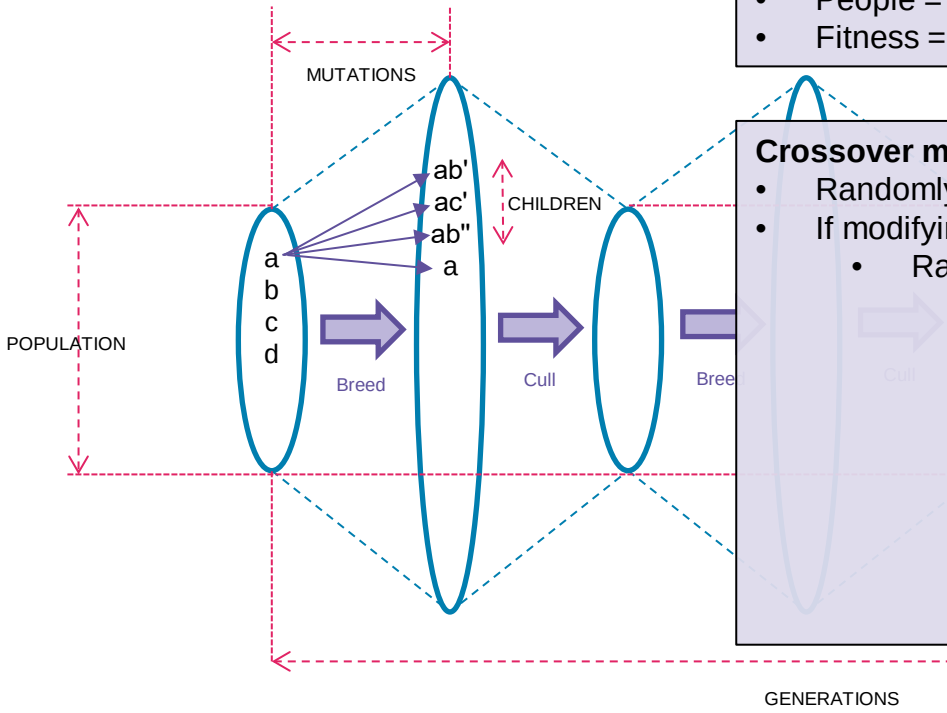
Algorithm:
- For GENERATIONS
  - Breed
  - Cull

Breed:
- For each person (Primary parent) in population:
  - For CHILDREN
    - Randomly pick a mate (Other parent)
    - Child := Primary parent
    - For MUTATIONS
      - Child := Crossover mutate(Child, Other parent)
    - Add Child to New population
  - Add Primary parent to New population

Cull:
- Reduce population to POPULATION fittest people

ALTRAN

Genetic Algorithm for Test Input Generation – Specifics



Instantiation:

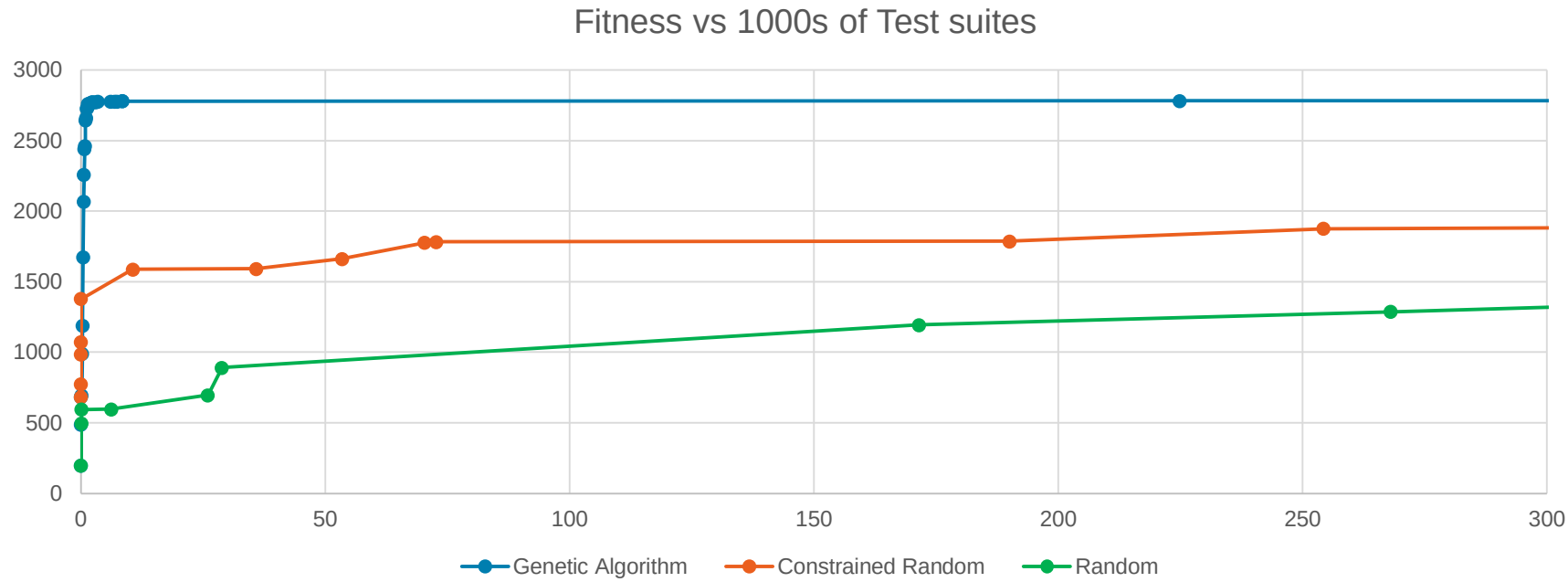
- People = Test suites of test cases of test steps
- Fitness = Most VCs covered in least test steps

Crossover mutation:

- Randomly either delete, duplicate or modify a test case
- If modifying a test case
  - Randomly either
    - Delete (delete a random test step),
    - Duplicate (duplicate a random test step, placing it at a random position in the test case),
    - Modify (replace a random field with the value that field has in another random test step of the same type),
    - Randomise (replace a random field with a random value), or
    - Add (add a new test step with random values for all fields).

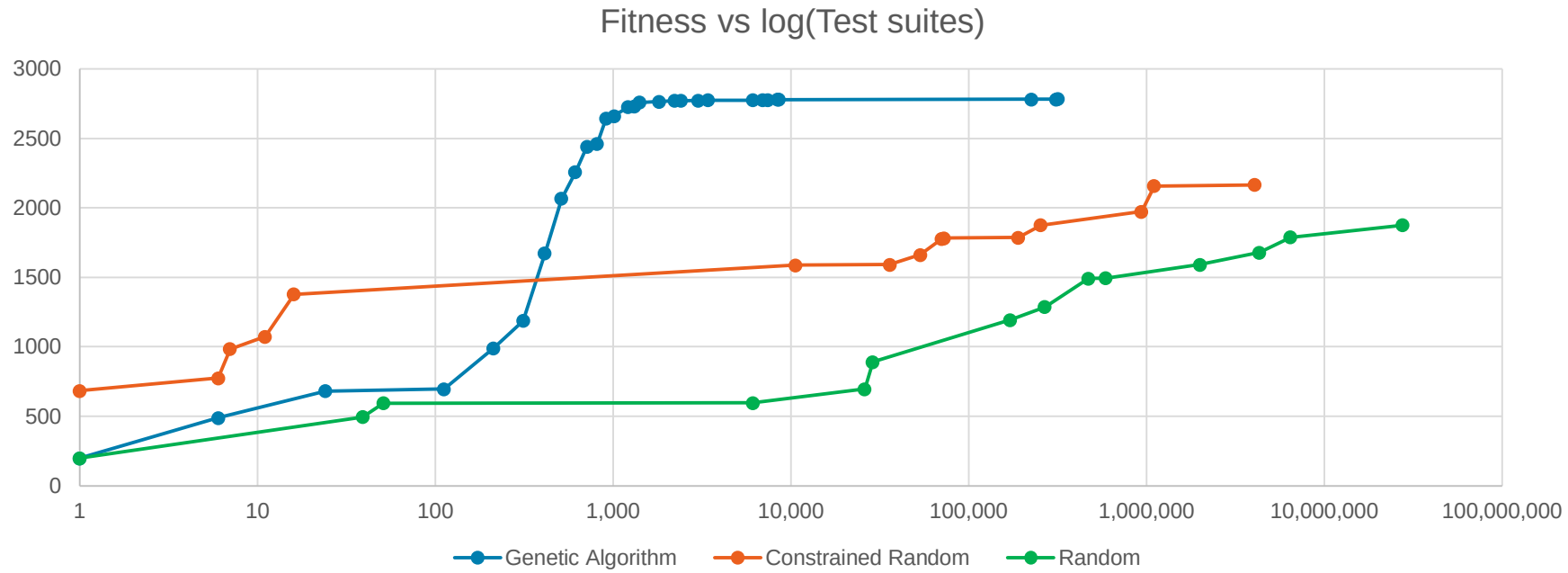
ALTRAN

Genetic Algorithm vs Constrained Random vs Random Test Inputs



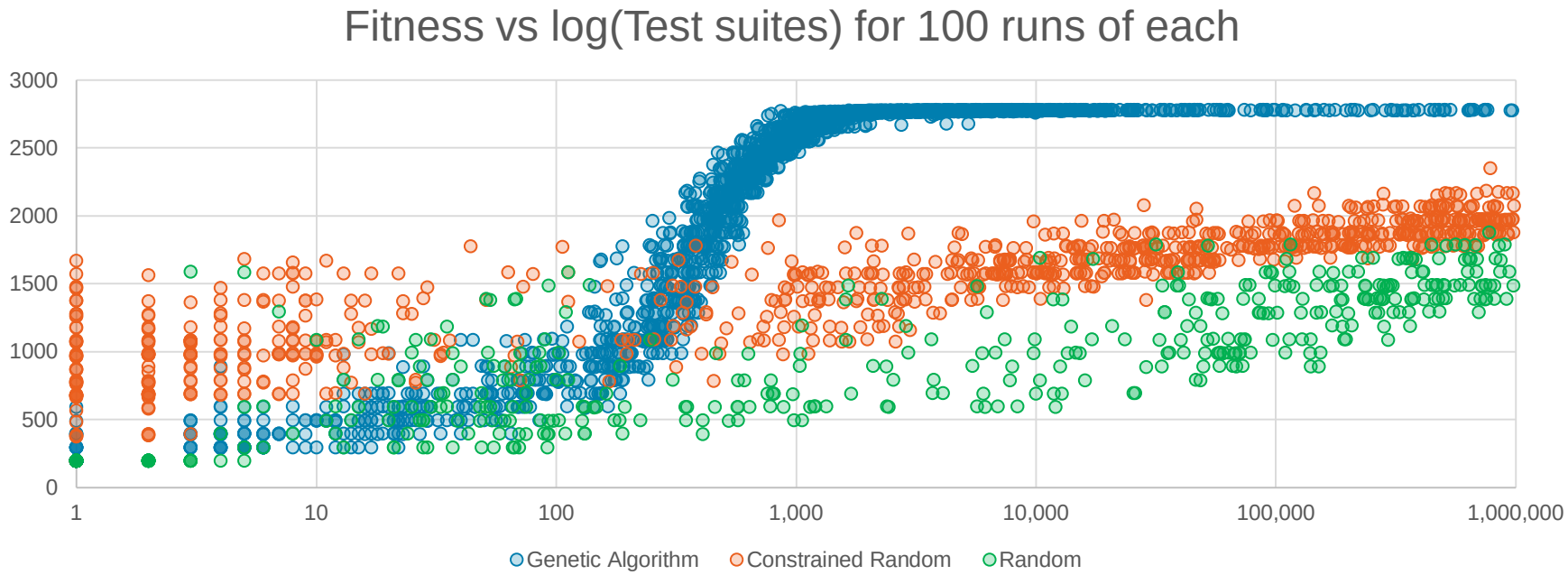
ALTRAN

Genetic Algorithm vs Constrained Random vs Random Test Inputs



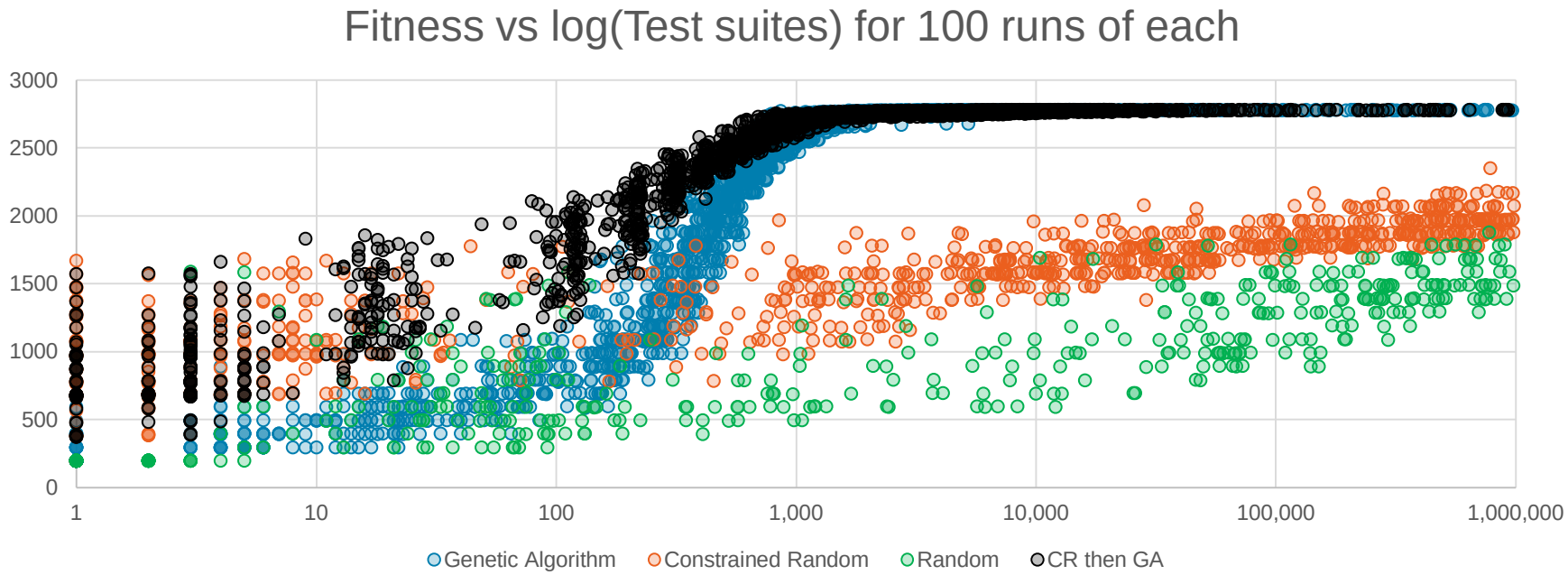
ALTRAN

Genetic Algorithm vs Constrained Random vs Random Test Inputs



ALTRAN

Genetic Algorithm vs Constrained Random vs Random Test Inputs



Acknowledgement: This work was supported by the HICLASS project, funded by the Aerospace Technology Institute and Innovate UK, as project number 113213.

Test data generation on FourSight

What is FourSight?

FourSight is the successor to NATS' hugely successful iFACTS system, both developed by Altran. Introduced in 2011, iFACTS predicts an aircraft's location up to 18 minutes into the future. Potential conflicts can then be easily identified and action taken early to avoid them. As well as the obvious safety benefits, the system has also helped cut emissions and resulted in a 40% capacity increase for some airspace sectors.

How is test data generated on FourSight?

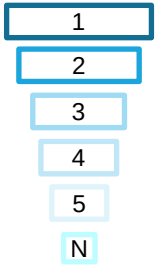
FourSight testing uses a combination of constrained random data generation and a 'Mutation Engine' to use techniques from Genetic Algorithms to give additional coverage. The 'Mutation Engine' typically gives an additional 10% test coverage beyond constrained random.

03.

Test Case Prioritization Approach

Test Case Prioritization Objective

- Test case prioritization seeks to find the ideal ordering of test cases
- Test cases are prioritized in order of its effectiveness to uncover faults.

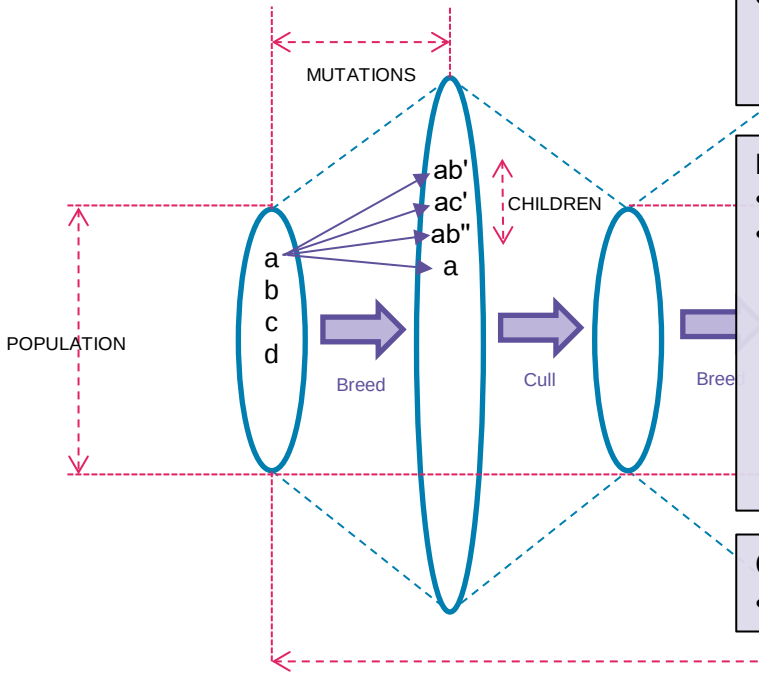


Effectiveness of prioritization is measured using Average Percentage of Fault Detected (APFD) Metric.

Benefits

- Leads to early detection of defects giving more time to fix and stabilize the system.
- Better quality software releases within limited time duration leading to higher customer satisfaction.
- Saves regression cost if lower ranked test cases are deprioritized to save precious time.

Genetic Algorithm for Test Case Prioritization – Concept



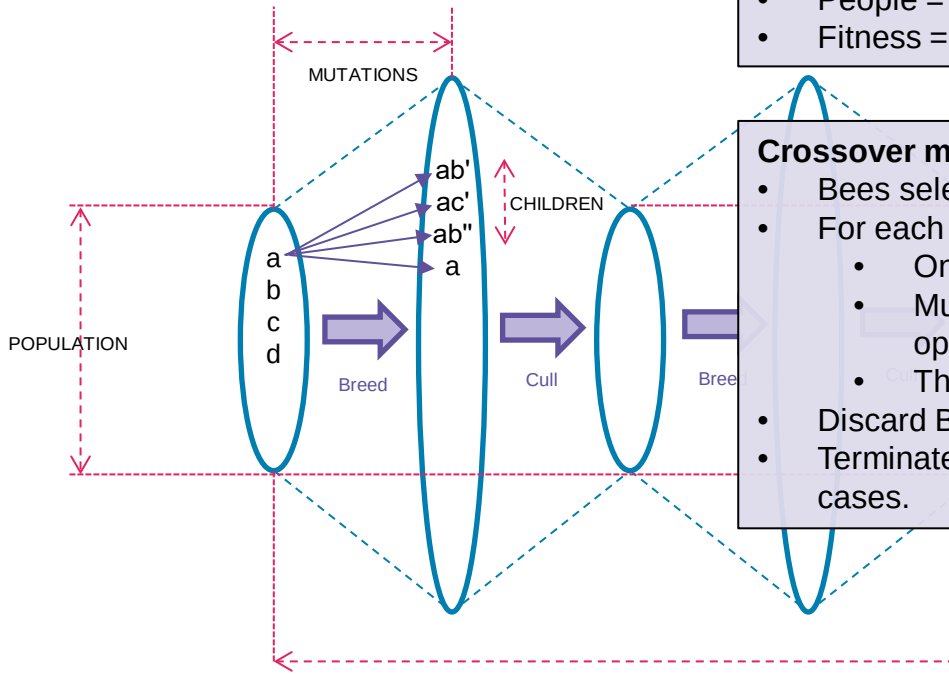
Algorithm:
- For GENERATIONS
  - Breed
  - Cull

Breed:
- Initialize Bees & Food Sources
- For each Bee (Primary Parent) in population:
  - Randomly assign food source to Bee
  - For CHILDREN
    - Randomly pick a mate (Other parent)
    - For MUTATIONS
      - Child := Crossover mutate(Primary Parent, Other parent)
    - Add Child to New population

Cull:
- Reduce population to POPULATION fittest people

ALTRAN

Genetic Algorithm for Test Case Prioritization – Specifics



Instantiation:

- People = Bees
- Fitness = Most Test Failures with less execution time

Crossover mutation:

- Bees selection
- For each bee:
  - One point crossover with another bee
  - Mutation of the data generated by crossover using OR operation.
  - The operation increases or decreases the fitness
- Discard Bees with low fitness (cull)
- Terminate on exit criteria(maximizing failures) or exhausting the test cases.

ALTRAN

Test Case Prioritization Solution

Data Preparation
- Read, impute and aggregate data for processing

Data Understanding
- Univariate, Multivariate and Time Series Analysis

Swarming and Genetic Algorithms based optimization
- Evaluate Feature Test Cases to be promoted to Regression Tests
- Rank Regression Tests in order of order of effectiveness to uncover faults

Average Percentage Faults Detected (APFD) metric
- Measure the effectiveness of the Regression Test Case Prioritization by calculating the APFD metric.

Third Party Tool Adapters

Test Management System

Test Management System

Test Automation System

ALTRAN

Automated Test Case Prioritization using Artificial Intelligence

Tier-1 networking equipment OEM – Router Product Line

Issue

Weekly regression taking 10-15 days (43% to 112% delay)

Frequent delay to close the test cycle due to late discovery of failures in the automated weekly regression cycle (5-7 days)

Solution

6 months historical data used for AI based prioritization

Test execution records used as the input to the prioritization algorithm based on **Hybrid Artificial Bee Colony** and **Genetic Algorithm**

Benefits

80% of issues discovered on day 1 of weekly regression

Of the defects are now discovered in the 1st day (for automated test run across 5 days)

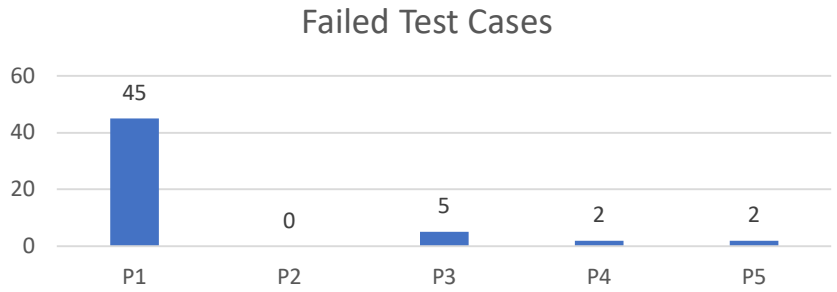
Test prioritization also key to manage emergency release (test-to-time-budget)

ALTRAN

Validation of Test Case Prioritization Model Using Last Regression Cycle

Train : Cycle 1 to Cycle 6

Test : Cycle 7 (last regression cycle)



Regression Start : 15-Jun-18				
Priority	**Total Failed**	**Cumulative Fail%**	**Total Passed**	**Total Tests(Pass+Fail)**
P1	45	83.33%	153	198
P2	0	83.33%	246	246
P3	5	92.59%	229	234
P4	2	96.30%	183	185
P5	2	100.00%	162	164
Total	54			

P1: Highest priority level**, P5:** lowest priority level

ALTRAN

Automated Test Case Prioritization using Artificial Intelligence

Project Execution

	T0	Test case prioritization functionality available as a part of ATLAS framework.
Business Understanding	**3 days**	Identified the issue that Regression Testing Cycle is not closing in time. Proposal of automated Test Case Prioritization as solution.
Data Understanding	**1 week**	Selection of test beds from which data is to be collected. 6 month test case execution data available. 1,547 test cases in current test suite.
Data Preparation	**2 week**	Extraction of test case execution data from proprietary Test Management tool using APIs. Preparing the data in the format required for automated Test Case Prioritization.
Modeling	**3 week**	Test Case Prioritization using ATLAS framework. Last regression test cycles used as test data.
Evaluation	**4 week**	Validated the results across multiple test beds
Deployment	**Final**	Demonstration

Team

- 1 Test Lead, Customer
- 3 Data Scientists, Altran

ALTRAN

04.

Altran Group in AI/ML

altran

Altran AI / ML Services Spectrum

Strategic Consultancy

Identifying opportunities for use of AI.

Designing roadmaps for AI adoption.

Advising on organisational & infrastructure change to enable AI.

Novel Research

Applying theoretical academic ideas into practice

Novel exploratory uses of machine learning

Proof of concepts of new ideas.

AI Implementation

AI system architecture and design

Design and implementation of robust validation processes

End-to-end system integration and long-term support service provision.

Caffe2 TensorFlow PyTorch Keras Natural Language ToolKit python hadoop Spark cassandra

ALTRAN

Questions?