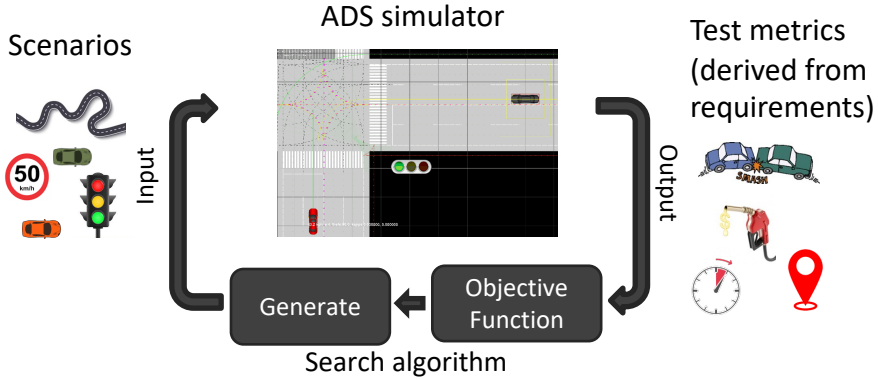
Smart Testing of Automated Driving Systems

Application of Search-based Test Generation Techniques for Automated Driving Systems

Paolo Arcaini, Fuyuki Ishikawa, Xiao-Yi Zhang

Research content



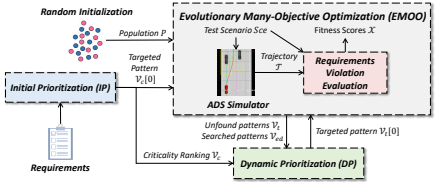
What do we solve ?

Scenario generation [ASE'21] Which driving scenarios to generate? The ADS considers different *requirements* and, sometimes, it sacrifices the satisfaction of non-critical ones to guarantee the critical ones. We generate scenarios in which these trade-offs happen.

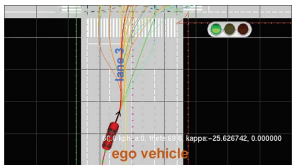
Scenario simplification [ICST'22] How to *simplify* a scenario so that it contains only the important elements? If too many elements are in a scenario, it is difficult to do debugging (understand why a failure occurred). We generate the *minimal* scenario that shows the same failure.

Scenario generation (Luo et al. Targeting Requirements Violations of Autonomous Driving Systems by Dynamic Evolutionary Search. In ASE'21)

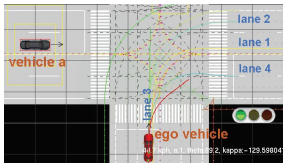
- **Pattern**: specific combination of satisfaction/violation of requirements
- **IP**: patterns are sorted by criticality
- **EMOO**: search algorithm that finds a scenario exposing a pattern
- **DP**: the list of patterns is sorted considering likelihood to occur



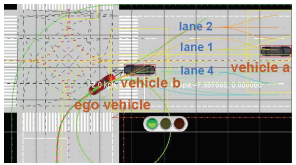
(1) acceleration to turn



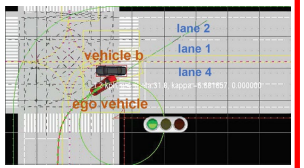
(2) vehicle-a appears suddenly



(3) critical curvature degree reached



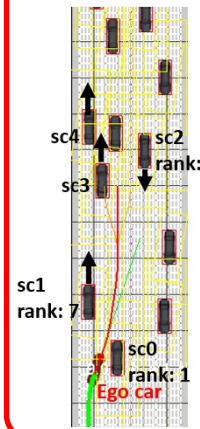
(4) collision



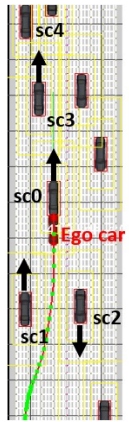
Scenario simplification (Arcaini, Zhang, Ishikawa. Less is More: Simplification of Test Scenarios for Autonomous Driving System Testing. In ICST'22)

Test scenario

start

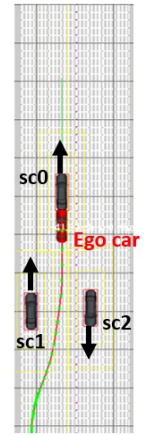


end



- *Why* does the ego car collide?
- Why does it not apply an evading manoeuvre?
- Our approach removes all the cars that are not necessary for the collision

Simplified scenario



- The ego car does not go in the opposite lane because of *sc2*
- But, it also does not try to go to the left lane: the reason of this behaviour must be investigated by ADS engineers