AmbieGen at the SBFT 2024 Tool Competition - CPS-UAV Track

Dmytro Humeniuk
dmytro.humeniuk@polymtl.ca
Polytechnique Montréal
Montréal, Canada

Foutse Khomh
foutse.khomh@polymtl.ca
Polytechnique Montréal
Montréal, Canada

ABSTRACT

Simulation based testing of autonomous systems prior to their deployment in the real world is of big importance. AmbieGen is a tool for automatic generation of virtual test cases for autonomous cyber-physical systems (CPS). In the context of SBFT 2024 CPS-UAV tool competition, we adopted it to the generation of scenes with obstacles for testing an unmanned aerial vehicle (UAV) obstacle avoidance system. AmbieGen leverages a genetic algorithm guided by a path planner to prioritize test scenarios to be evaluated in the simulation. It could reveal some critical failures in all the 6 use cases considered in the competition.

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**; • **Computing methodologies → Search methodologies**.

KEYWORDS

test cases, UAVs, tool competition, genetic algorithm, path planning

ACM Reference Format:
Dmytro Humeniuk and Foutse Khomh. 2024. AmbieGen at the SBFT 2024 Tool Competition - CPS-UAV Track. In *2024 ACM/IEEE International Workshop on Search-Based and Fuzz Testing (SBFT '24), April 14, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3643659.3648552

1 INTRODUCTION

Autonomous unmanned aerial vehicle (UAV) are complex cyber-physical systems (CPS), that heavily rely on machine learning (ML) components for perception and navigation. ML models are typically trained on a limited amount of data, giving rise to a risk of poor generalization in the novel environments. It is thus critical to perform simulation based testing of UAVs to ensure their robustness.

One of the challenges posed by running the tests in the simulation is high computational cost and execution time. For instance, one test scenario for a UAV can take around 2 minutes to execute. In our previous research work, we proposed AmbieGen tool [1], that relies on surrogate functions approximating the system behavior to guide the evolutionary search. Only the test cases achieving a high surrogate function score are evaluated with a computationally expensive system model in a simulator (we also refer to it as the full system model). This year, a CPS-UAV testing competition is organized within the SBFT 2024 workshop [4]. The goal of the competition is to generate virtual test scenarios falsifying an autonomous UAV navigating in a designated area with obstacles. In this paper, we describe the AmbieGen tool we submitted to SBFT 2024 CPS-UAV testing competition. AmbieGen leverages a genetic algorithm, where the test scenarios are first evaluated with a surrogate fitness function, represented by an RRT* planner. If the length of the path returned by the planner exceeds a defined threshold, the simulation based system model is used to evaluate the test scenario. An additional fitness value is assigned if the test scenario results in a failure.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SBFT '24, April 14, 2024, Lisbon, Portugal
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0562-5/24/04
https://doi.org/10.1145/3643659.3648552

2 TEST SUBJECT

The test subject in the SBFT-2024 CPS-UAV competition is an Iris [1] drone running PX4 [2] control firmware in Gazebo simulator [5]. This testing pipeline is based on the work of Khatiri et al. [2], [3]. The drone is tasked to safely navigate in a closed space with box-shaped obstacles. An obstacle is defined by its size (length, width, height) and position in the simulation environment (x,y,z) in meters and its rotation angle in degrees. At maximum four obstacles are allowed in the scene. The drone is expected to follow a given trajectory autonomously, avoiding the obstacles. The goal of test generation is to force the UAV to crash or significantly divert from its intended path. The test case is considered as failed if the drone approaches the obstacle at an unsafe distance, which is set to be 1.5 m in the competition rules. Examples of a failed and passing test cases are shown in Figure 1. In the figures you can see the top view on the scene with the obstacles and the UAV trajectory (blue line). In the next section, we provide more details about AmbieGen tool implementation.

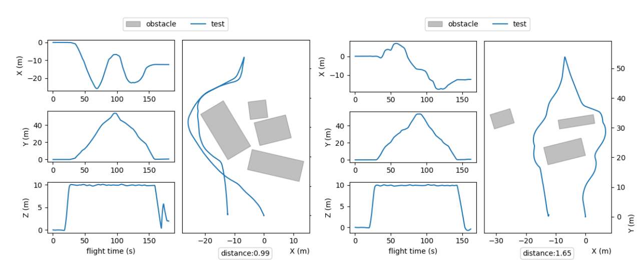

Figure 1: Examples of a failed (a) and passing (b) test scenarios for the UAV

[1]https://www.arducopter.co.uk/iris-quadcopter-uav.html
[2]https://docs.px4.io/v1.12/en/

3 AMBIEGEN TOOL DESCRIPTION

The operation of the AmbieGen tool is presented in Algorithm 1. The hyperparameter values are listed in Table 1.

Table 1: AmbieGen algorithm hyperparameters

Parameter	Description	Value
N_{pop}	Population size	50
N_{off}	Number of offsprings	25
$CROS$	Crossover rate	0.9
MUT	Mutation rate	0.4
ϕ	Surrogate fitness threshold	30
ϕ_{sim}	Simulation fitness threshold	1.5

The objective of the algorithm is to maximize the F_1 surrogate fitness function value. The algorithm starts by generating an initial population of solutions. We represent the individuals as one-dimensional vectors of size 25x1. The first element corresponds to the number n of box-shaped obstacles in the scene. The remaining 24 elements describe the obstacles that are present in the scene (up to 4). Each obstacle is described by 6 elements: x and y coordinates of the obstacle center position, obstacle length l, width w, height h in meters, and rotation r in degrees. For the initial population, we generate the vectors randomly by sampling from the allowed value ranges, which are listed in Table 2.

Table 2: Allowed ranges for the parameters

n	x	y	l	w	h	r
$1-4$	$-40-30$	$10-40$	$2-20$	$2-20$	$15-25$	$0-90$

Algorithm 1 AmbieGen algorithm for the CSP-UAV competition

Require: Termination criterion T
Require: Surrogate fitness F_1 threshold ϕ
Require: Simulation failure threshold ϕ_{sim}
Require: hyperparameters N_{pop}, N_{off}, CROS, MUT.

```
Set hyperparameters.
Initialize population P.
Evaluate the objective F_1 over population P.
while not (T) do
    P' ← TournamentSelection(P)
    P' ← Crossover(CROS, P')
    P' ← Mutation(MUT, P')
    F_1 ← SurrogateFitnessEval(P')
    for i = 0 to N_pop do
        if F_1i ≥ φ then
            f_sim ← ExecuteSimulation(P'_i)
            if f_sim ≤ φ_sim then
                F_1i ← UpdateFitness(P'_i)
            end if
        end if
    end for
    P ← SurvivorSelection(P, P')
end while
```

The selected individuals undergo the crossover and mutation operations with the defined probabilities. We use a one-point crossover as well as a customized mutation operator, where an obstacle is chosen at a random index and replaced by a randomly generated obstacle. We then evaluate the surrogate fitness of each individual. We define it as a length of the path returned by a well-known path planning algorithm RRT* [6]. For simplicity, we used a fixed target trajectory as the input for the RRT*, which was a straight line going through the middle of the defined test scenario zone. However, the input trajectory can be adjusted depending on the defined flight mission. Intuitively, the longer the path produced by the planner, the more deviations from the shortest path the UAV will have to make, increasing the risk of its failure. If the length of the path exceeds ϕ relative units, we evaluate the test in the simulation environment provided by the competition. If a failure is revealed in the simulator, i.e., the distance that the UAV approached an obstacle is less than ϕ_{sim}, additional fitness is assigned to an individual as shown in Equation 1:

$$F_{1i} = l + f_{sim_1} + f_{sim_2}, \quad (1)$$

where l is the length of path returned by a path planner, $f\_sim_1 = 10$ is the additional fitness for the failed test case in the simulation and $f\_sim_2 = 10 \times \frac{1}{n}$ is an additional fitness for the number of obstacles. The fewer obstacles were used to cause the failure, the more additional fitness is assigned. The individuals with the highest fitness F_1 are selected for the next generation. The algorithm terminates when the number of executions in the simulator reaches the allowed budget, which was 200 simulations in this competition.

4 CONCLUSIONS

In this paper, we present AmbieGen tool adopted to the generation of test scenarios for an autonomous UAV. The tool leverages evolutionary search, where the test cases are prioritized to be executed in the simulator based on the output of a path planning algorithm. AmbieGen has proven to be effective at generating diverse and challenging test scenarios by revealing failures in all the 6 case studies introduced in the competition, More detailed information about the tool performance can be found in the competition report [4].

REFERENCES

[1] Dmytro Humeniuk, Foutse Khomh, and Giuliano Antoniol. 2023. AmbieGen: A search-based framework for autonomous systems testingImage 1. *Science of Computer Programming* 230 (2023), 102990. https://doi.org/10.1016/j.scico.2023.102990

[2] Sajad Khatiri, Sebastiano Panichella, and Paolo Tonella. 2023. Simulation-based test case generation for unmanned aerial vehicles in the neighborhood of real flights. In *2023 16th IEEE International Conference on Software Testing, Verification and Validation (ICST)*.

[3] Sajad Khatiri, Sebastiano Panichella, and Paolo Tonella. 2024. Simulation-based Testing of Unmanned Aerial Vehicles with Aerialist. In *International Conference on Software Engineering (ICSE)*.

[4] Sajad Khatiri, Prasun Saurabh, Timothy Zimmermann, Charith Munasinghe, Christian Birchler, and Sebastiano Panichella. 2024. SBFT Tool Competition 2024 - CPS-UAV Test Case Generation Track. In *IEEE/ACM International Workshop on Search-Based and Fuzz Testing, SBFT@ICSE 2024*.

[5] Nathan Koenig and Andrew Howard. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, Vol. 3. IEEE, 2149–2154.

[6] Steven LaValle. 1998. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811* (1998).