On Scheduling of Fuzzing Test

Hongbo Zhang
Supervisors: Steve Blackburn, Tony Hosking, Shane Magrath

Australian National University

In order to improve the efficiency of fuzzing test, we study the problem of scheduling policy in the context of multi-armed bandit problem.
We investigate a new and more practical crash model by simulation in which finite unique crashes could be potentially triggered.
The results figure that the scheduling problem in fuzzing test is complicated that the optimal policy will depend on the accurate modelling the nature of crashes.

Fuzzing Test and Our Focus

Fuzzing (Fuzz test) is a testing technique by providing lots of random input data into tested programs, so that the flaws could be found and located. By the way, Miller found more than 25% programs in UNIX could be crashed by fuzzing [1].
In order to improve the efficiency of fuzzing, there are lots of open questions such as fuzzing techniques, seed selection policy, scheduling policy and so on. In this work, we will focus on **scheduling policy** of fuzzing test.
Scheduling policy can be studied by various approach, including theory, simulation and experiment. We will study this problem by **simulation**.

Crash Models

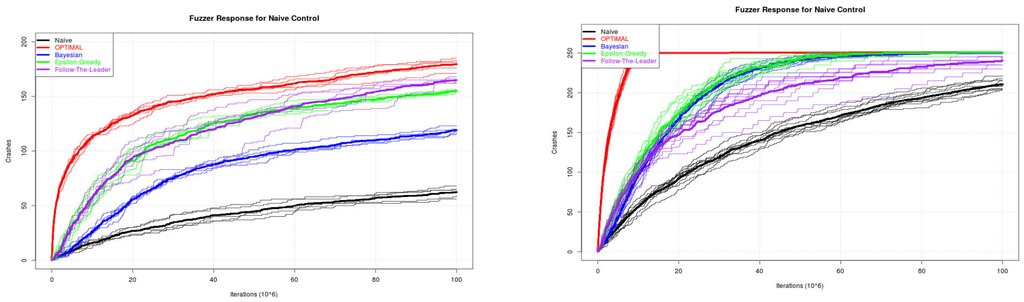
We need to model the flaws mathematically in the study of simulation. The accuracy of model will be important. However, it is too difficult to build a bug model. Consequently, we will focus on crashes. **Crash model** will be built and studied in this work.

- Bernoulli Model
  This is the most naïve crash model that any seed in a fuzzing will cause infinite number of crashes with the same probability.
- Limited Crashes Model
  This model improves the naïve model in two aspects.
  1. The probability of triggering a further crash will decrease once it has already found one.
  2. The number of crashes triggered by a given seed will be finite.

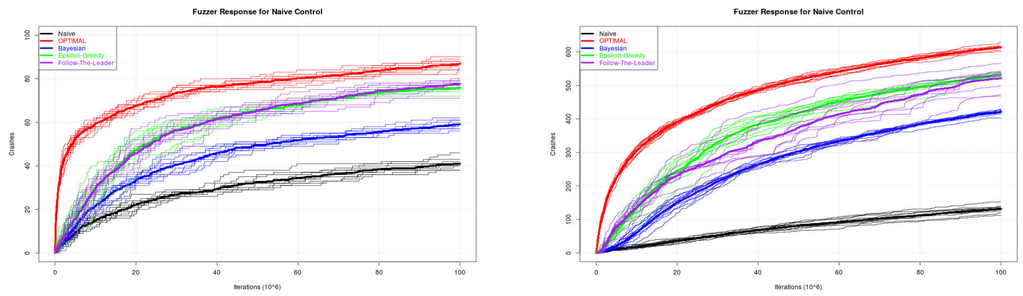
Result: α-Upper Confidential Bound I

The mean term and variation term compete in UCB1. Therefore, the α presents the competing between exploration and exploitation

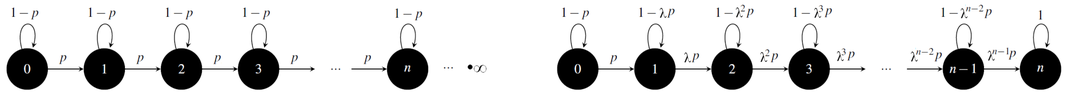
- For Bernoulli case, follow the leader is the optimal
- For limited crashes case, it is not.



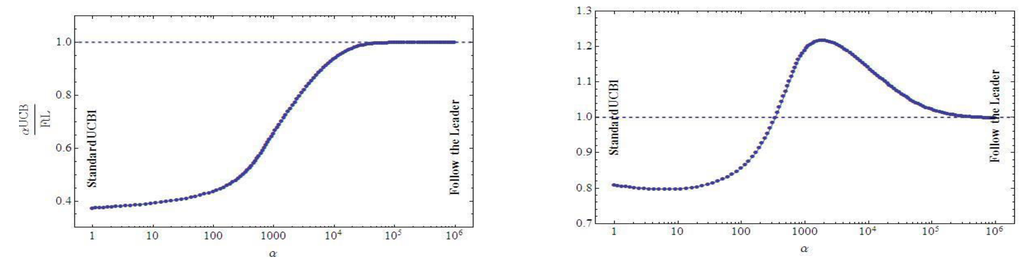
Left: a fuzzing simulation with a crash model with infinite number of crashes. Right: a simulation with limited crash model triggering 5 crashes at most



Left: a fuzzing simulation with decay factor λ=0.3. Right: a simulation with decay factor λ=0.9



Left: naïve Bernoulli crash model. Right: a more practical limited crashes model, where λ is decay factor.



Left: . α-UCB1 with a crash model with infinite number of crashes. Right: a simulation with limited crashed model triggering 5 crashes at most.

Result: Limited Crashes

As the work by Gittins [2], follow the leader is the optimal policy in Bernoulli model. However, in the limited crashes model, it is not the case.

- n>n*, being similar to the naive Bernoulli crash model, follow the leader policy works best among all the policies used in the simulation.
- n<n*, the follow the leader doesn't perform the optimal.

where n* is maximun expected number of crashes found in a fuzzing campaign.

Result: Decay Factor

For small decay factor, follow the leader wins. However, for large decay factor, it is not the case.
The smaller decay factor will favor the policy which finds crashes earlier in fuzzing. In this case, exploration is more important than exploitation. For large decay factor, the situation changes.
If decay factor is 1, it decays to Bernoulli's case.

Discussion

- **More exploitation, less exploration.** In the more realistic limited crashes model, the total number of crashes expected in a fuzzing campaign will be low due to both the low probability of triggering a crash and limited number of crashes potentially triggered by a seed. Therefore, the balance in exploration and exploitation changes. Finally, it affects the optimal scheduling policy, and can explain all the results above.
- **Mortal bandit**. This problem can be studied in the context of multi-arm bandit problem, especially, mortal bandit problem [3], in which the bandit has finite lifetime.
- **Bug model**. However, crash model is different from bug model. A bug model should be built for further studies.
- The results figure that the scheduling problem in fuzzing test is complicated that the optimal policy will depend on the accurate modelling the nature of crashes.

FOR FURTHER INFORMATION
Hongbo Zhang
u6170245@anu.edu.au

References
[1] Miller B P, Fredrikson L, So B. An empirical study of the reliablity of UNIX utilities. Communications of the ACM, 1990,33(2):32.
[2] Gittins J, Jones, D, A dynamic allocation index for the discounted multarmed bandit problem, in Progress in Statistics, Amsterdam, Netherland: North-Holland, vol.9, pp. 241-266, 1974.
[3] Chakrabarti D, Kumar R, Filip R, Eli U, Mortal Multi-Armed Bandits, in Advances in Neural Information Processing Systems 21, Curran Associates, Inc., pp. 2730280, 2009

DATA 61 CSIRO