TEST DATA GENERATION: A HYBRID APPROACH USING CUCKOO AND TABU SEARCH

Authors:-

Krish Perumal (Undergraduate)

Jagan Mohan Ungati (Undergraduate)

Gaurav Kumar (Undergraduate)

Nitish Jain (Undergraduate)

Raj Gaurav (Undergraduate)

Praveen Ranjan Srivastava (Lecturer, Dept. of Comp. Sci. & Information Systems)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

ROAD MAP

- Motivation
- Existing Approaches
- Proposed Approach
- Experimental Analysis
- Discussion
- Conclusion

MOTIVATION

- Software Testing – Key Part of SDLC
- SQS Compact Paper 2011: > 50 billion euros on testing
- NIST's National Impact Estimates (2002-03): $60 billion
- Cost can be drastically reduced, by tackling the inefficiencies of the existing approaches.

TEST DATA GENERATION

- The generation of test cases during the testing phase of software development
- Why automate it?
  - To reduce the cost of resources, both human and material
  - To reduce the enormous amount of time spent in identifying the test cases
  - To increase the reliability of testing

EXISTING APPROACHES

- Random Testing
  - Low efficiency in case of large codes
- Concolic Testing
  - Inability to test nondeterministic programs
- Search Based (Optimization) Testing
  - By far, the most successful
  - But suffer from time and cost optimality issues
- Adaptive and Heuristics Based Approaches
  - Ant Colony Optimization
  - GA – Problems of blind search and slow convergence
  - GA with Tabu Search – Combines local + global optimization

    - The best in terms of node coverage and time optimality till now

PROPOSED APPROACH

A Meta-Heuristic Approach Inspired by Nature

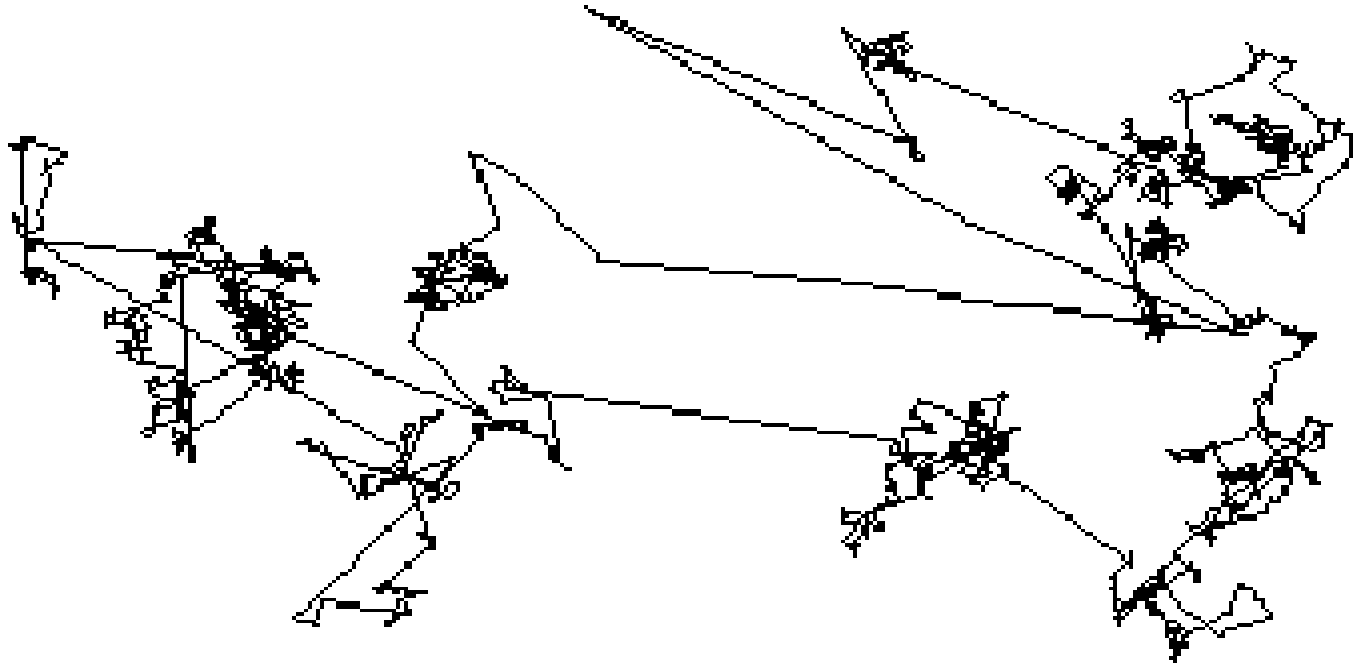
- Cuckoo Search
  - Found to be more generic and robust for many optimization problems
- Levy Flights
  - Ensures best overall coverage of search space
- A Variation of Tabu Search
  - Ensures movement from locally optimal to globally optimal solutions

CUCKOO SEARCH

- Imitation of the behaviour of cuckoos in nature
  - Cuckoos dump their eggs in other birds' nests
  - So we basically tried to put this evil behaviour to good use!
- Rules, as described by Yang and Deb, the proponents of Cuckoo Search:-
- Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.
- The best nests with high quality of eggs will carry over to the next generations.
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a certain probability. In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest.

LEVY FLIGHT

- A random walk where the random step length is drawn from a levy distribution.
- The random walk via levy flight is more efficient in exploring the search space as its step length is much longer in the long run.



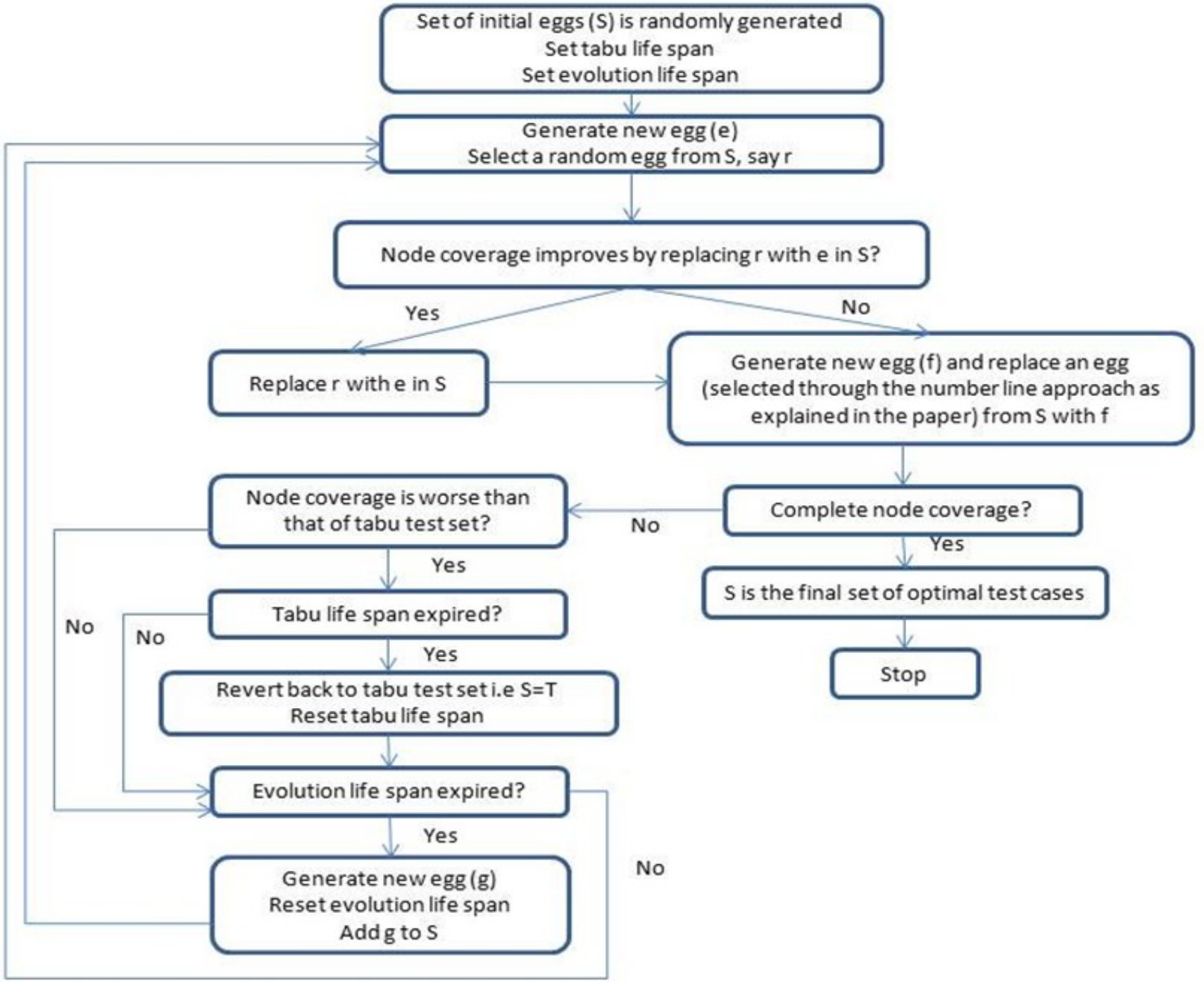
LEVY FLIGHT (CONTD…)

- Some of the new solutions should be generated by Levy walk around the best solution obtained so far, thus speeding up the local search.
- However, a substantial fraction of the new solutions should be far enough from the current best solution, thus ensuring that the system will not be trapped in a local optimum.

TABU SEARCH

- A mathematical optimization method
- Improves the performance of a local search by marking the previously visited solutions as “tabu”.
- It ensures that the algorithm does not get stuck in a locally optimum solution, thereby enabling faster convergence to the globally optimal solution.

METHODOLOGY



NOTATIONS USED

- M = Total number of nodes in Control Flow Graph
- N = Current number of input test cases
- I = Current input test case set
- t = Tabu life-span
- e = Evolution life-span
- T = Tabu test case set
- n(S) = Number of nodes covered by a set of test cases S/Node Coverage
- E_i= Individual test case (A cuckoo egg)

THE ALGORITHM

- Before initiating the algorithm, it needs some input data for assistance.
- The user needs to input values for N, M, t and e
- These values are used by the algorithm, and might vary from program to program.
- With the initial input values, a test case that satisfies the constraints set by input values is randomly generated.

THE ALGORITHM (CONTD…)

- Say, the initial test case consists of 3 eggs - E1, E2 & E3. (Generated based on initial parameters set by user) Also, we initially set T=I i.e. good tabu list is initial test case set
- Generate a new random egg Ei using levy flight.
- Select at random, an egg from I (say E2).
- If n ({E1, E2, E3}) < n ({E1, Ei, E3})

  Replace E2 by Ei in I

  Else

  Nothing

  EndIf

THE ALGORITHM (CONTD…)

- Generate a new egg, say Ej, using levy flight.
- Replace one of the eggs from I with Ej.

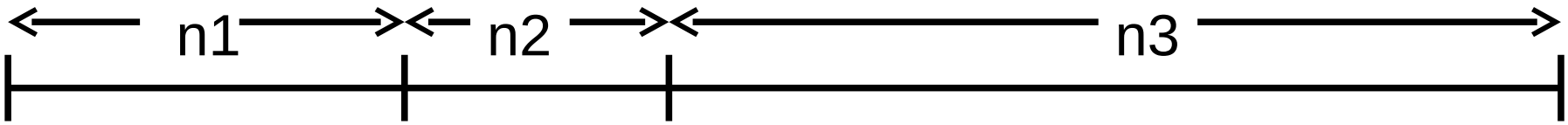
  (The probability of a particular egg from I to be replaced depends on the number of test cases covered due to the egg's presence in I)

Let

n({Ej, Ei, E3}) = n1

n({E1, Ej, E3}) = n2

n({E1, Ei, Ej}) = n3



Generate a random number in the range (0, n1+n2+n3).
According to the range in which the random number falls, select the corresponding test case for the next step.

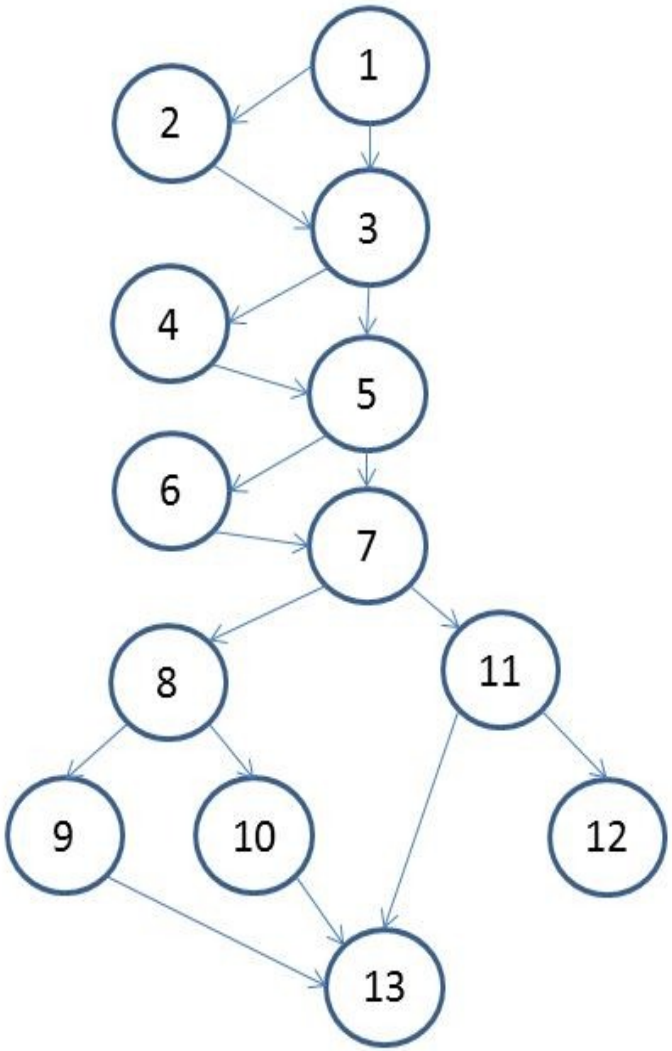
```
If e==0
        I=T
        N=N+1
        Reset e
EndIf
If n(I) <= n(T)
       If t==0
                I=T
                Reset t
                    Else
                T=I
                 EndIf
EndIf
```

```
If n (I) == M
        Output I
        Stop
Else
        Continue
EndIf
```

EXPERIMENTAL RESULTS

TRIANGLE PROBLEM

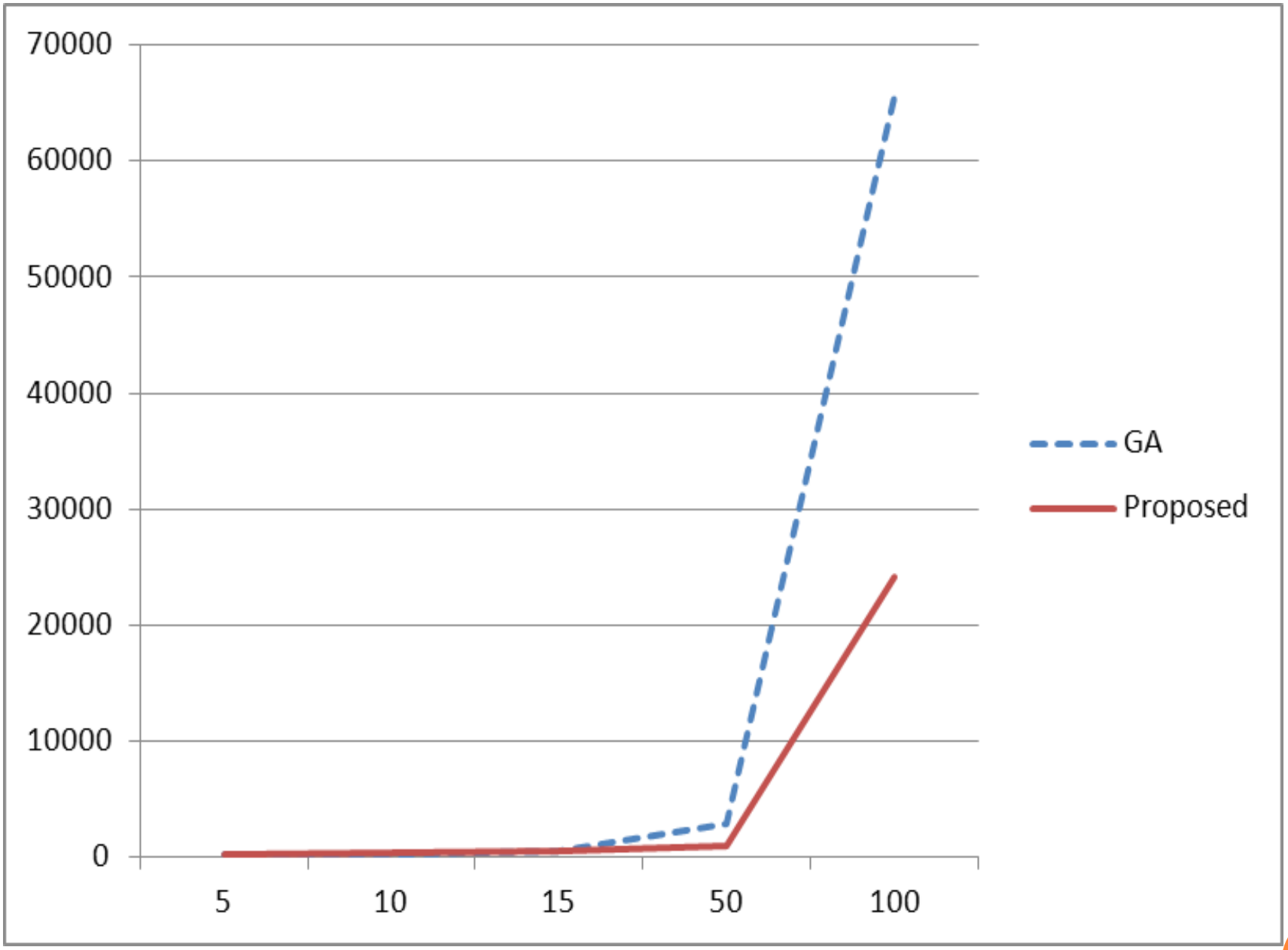
```
int triType(int a, int b, int c){
1 int type=PLAIN;
1 if(a<b)
2         swap(a,b);
3 if(a<c)
4         swap(a,c);
5 if(b<c)
6         swap(b,c);
7 if(a==b){
8         if(b==c)
9                   type=EQUILATERAL;
10        else
                    type=ISOSCELES;
    }
11 else if(b==c)
12                  type=ISOSCELES;
13 return type;
}
```



RESULTS

	GENETIC ALGORITHM		PROPOSED APPROACH	
INPUT TEST CASE RANGE	AVG NO OF TESTS	AVG NO OF UNCOVERED NODES	AVG NO OF TESTS	AVG NO OF UNCOVERED NODES
0-5	290	NONE	215	NONE
0-10	295	NONE	379	NONE
0-15	605	NONE	582	NONE
0-50	2860	NONE	1013	NONE
0-100	65535	1	24233	NONE

*Shen X., Wang Q., Wang P., Zhou B., “Automatic generation of test case based on GATS algorithm”, Granular Computing, GRC‟09. IEEE International Conference, pp. 496-500, 2009.



DISCUSSION

- The Cuckoo Search part of the algorithm is a close resemblance to the GA model
- The reproduction, crossover and mutation are modeled through Cuckoo search
- However, the implementation is considerably different, thus giving a completely different perspective towards optimization
- The variation of Tabu search, in combination with Levy flights, ensures that iterations are not wasted in traversing the local space for more than a certain amount of time
- This ensures faster convergence to the optimal solution
- This also ensures 100% node coverage in the CFG

DISCUSSION CONT…

- Currently, there is no global selection criterion for evolution and tabu lifespan
- However, t=20 and e=1000 have been shown to work for the problems our work concentrated on

CONCLUSION

- The hybrid approach involving Cuckoo and Tabu Search has been shown to work better than GA as well as the combination of GA and Tabu Search.
- A change of perspective of software testing researchers towards optimizations through Cuckoo Search may result in drastic reduction in costs and time involved in this phase of SDLC