Test Case Optimization and Prioritization Using Improved Cuckoo Search and Particle Swarm Optimization Algorithm

Chandrasekhara Reddy T1*, Srivani V2, A. Mallikarjuna Reddy1, G. Vishnu Murthy1

1Department of CSE, Anurag Group of Institutions, Hyderabad, India
2Department of CSE, MLR Institute of Technology, Hyderabad, India
*Corresponding author E-mail: chandu.tummuru@gmail.com

Abstract

For minimized t-way test suite generation (t indicates more strength of interaction) recently many meta-heuristic, hybrid and hyper-heuristic algorithms are proposed which includes Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Genetic Algorithms (GA), Simulated Annealing (SA), Cuckoo Search (CS), Harmony Elements Algorithm (HE), Exponential Monte Carlo with counter (EMCQ), Particle Swarm Optimization (PSO), and Choice Function (CF). Although useful strategies are required specific domain knowledge to allow effective tuning before good quality solutions can be obtained. In our proposed technique test cases are optimized by utilizing Improved Cuckoo Algorithm (ICSA). At that point, the advanced experiments are organized or prioritized by utilizing Particle Swarm Optimization algorithm (PSO). The Particle Swarm Optimization and Improved Cuckoo Algorithm (PSOICS) estimation is a blend of Improved Cuckoo Search Algorithm (ICSA) and Particle Swarm Optimization (PSO). PSOICS could be utilized to advance the test suite, and coordinate both ICSA and PSO for a superior outcome, when contrasted with their individual execution as far as experiment improvement.

Keywords: Particle Swarm Optimization, Improved Cuckoo Search, Quantitative Measure, Test case generation, Prioritization, Software Testing

1. Introduction

For achieving brilliant product or project and testing is one of the major strategies. Testing is set up to distinguish nearness of mistakes is cause improper programming outcome. Programming testing is one of the essential methods used to accomplish high caliber in Software. Testing is a tedious and expensive assignment it utilizes roughly half of the product framework improvement assets [1] [2]. Testing can likewise be characterized as the way toward confirming and assessing it to ensure that product meets the specialized and business prerequisite [3]. The upkeep stage in a Software Development Life Cycle includes regression testing to be done widely. It is important to retest the current test suite at whatever point any adjustments are done to the product. Regression testing is the wonder of re-running the experiments from the test suite to guarantee bug free product. It ensures that adjustments in the product have not impacted its useful attributes. Confirmation is done to guarantee that the product meets detail and is near basic testing while approval is done on code itself. The lesser the changes we made, the lesser disruptions we find while testing. Streamlining strategies have been adequately utilized as a part of experiment era and prioritization as of late. Although, various enhancement strategies had been proposed and great outcomes has been acquired. For example, multifaceted nature in powerful informational indexes, and higher time utilization for joining exists in the conventional streamlining methods. In this way, there is an extent of change for the change of advancement comes about. This exploration work concentrates on utilizing the proper improvement procedures, for experiment prioritization which gives the ideal outcomes.

This approach uses intelligence based techniques for test case optimization and test case prioritization. Several swarm intelligence approaches have been observed to produce meaningful results in terms of accuracy, convergence behavior, and time taken. This research uses a couple of recent intelligence approaches called as the Improved Cuckoo Search Algorithm and Particle Swarm Algorithm (PSO) for Test Case Optimization and Prioritization. Cuckoo search algorithm is more efficient than Genetic Algorithm (GA) another artificial life based algorithms. Compare to GA, and PSO Cuckoo Search (CS) offers lightweight computation and relying only on three parameters: nest size, max generation and probability pa [10].

2. Background Study

Hyper-heuristics are alternative to meta-heuristics. Hyper-heuristics is an abnormal state approach which plays out an inquiry over the space shaped by an arrangement of low level heuristics which work on the issue space. Unlike typical meta-heuristics, there is a logical separation between the problem domain and the prominent level hyper-heuristic [4] [7]. Apart from increasing the level of generality, hyper-heuristics can also be competitive with bespoke meta-heuristics. Hong Me et al [11] have proposed a way to deal with organizing experiments without scope data that working on Java programs tried under the JUnit structure an inexorably well-known class of frameworks. Utilizing the CIT approach, the experiments could be diminished drastically.
this experiment does not uncover the correct item because as it can be found in the compelled list the calculation sort SCC requires the inquiry sort to be DFS, while in the produced test, it is BFS. Henceforth, this experiment is not legitimate as it abuses the oblige [6] [9].

3. Implementation

Programming measurements are proposed to work out the product quality in view of the Improved Cuckoo Search Algorithm is the principle intension of the recommended technique. Figure 1. shows the Block Diagram of the proposed method. At first the experiments are delivered from the application program. From that point forward, the quality based components are expelled from the experiments the elements are Fault and Execution Time [10]. Next, we utilize Particle Swarm Optimization (PSO) calculation to streamline the evacuated highlights. After that later the components are removed from that cutting-edge parts they are Cohesion, Coupling, if regard, by then these components are again streamlined [5].

3.1. Improved Cuckoo Search Algorithm

Improved Cuckoo Search algorithm is motivated by the rearing conduct of the cuckoos and mitigates to execute. There are a couple of nests in cuckoo search. The new and better arrangement is supplanting the most terrible arrangement in the nests [11]. The consequent portrayal plot is chosen by Improved Cuckoo Search calculation. The following are the estimation rules amid the pursuit procedure [12] [13].

(1) Each cuckoo lays one egg (arrangement) at once and dumps its egg in an arbitrarily picked nest.
(2) The best nest with a great egg (better arrangement) will be extended to next iteration.
(3) The number of accessible host nest is settling. A host winged animal can find an outsider egg with a likelihood.
(4) Through the arbitrary end instrument in improved cuckoo search mode has both position and velocity. velocity in the cuckoo seek mode is not changed, and the present velocity of the individual is the velocity resulted by Particle Swarm Optimization search mode.
(5) An individual is updated its position and velocity by utilizing the procedure of Particle Swarm Optimization Algorithm.
(6) An individual in ICSA has no velocity updating formula, whereas an individual in Particle Swarm Optimization search mode has both position and velocity. velocity in the cuckoo seek mode is not changed, and the present velocity of the individual is the velocity resulted by Particle Swarm Optimization search mode.
(7) For this situation, the host fledgling can either discard the egg or destroy the nest and assemble a totally new nest.

For this situation, the host fledgling can either discard the egg or destroy the nest and assemble a totally new nest.

3.2. PSO General Algorithm.

This feature prevents PSO from having operators like cross-over and mutation genetic algorithm (GA) or other algorithms, which are reliant on natural evolution to form innovative solutions. In the beginning to searches in the global space by taking all the particles in the swarm as a nearer node and sharing information many particles. The particle takes in the experience from other partner particles when they abuse data. This will locate a promising locale in the global space that the calculation examines more enhanced arrangement toward the end. Because of the distinctive natures of the applications since its first rise until the point when PSO has experienced diverse improvements and advancements [4] [5].

Hybrid PSOICSA Algorithm

The PSO has good understanding, straightforward operation, and fast seeking. Be that as it may, in taking care of a substantial complex issue, PSO turns out to be effectively caught in nearby ideal. This shortcoming must be overcome to expand the practicability of PSO. ICSA has points of interest, for example, few control parameters and high proficiency, yet it additionally has a few deformities, for example, moderate meeting velocity and low exactness. To enhance the execution of ICSA, PSO is presented in the refresh procedure of ICSA. In this manner, a PSOICSA cross breed calculation is created. PSOICSA first uses particle hunt space to inquiry, and after that it utilizes the position of the PSO refresh mode to quicken the particles to the ideal arrangement merging. In the meantime, the arbitrary disposal component of ICSA can effectively escape nearby optima, in this way enhancing the execution of scanning for the ideal arrangement. Figure 2 shows the flow diagram of the PSOICSA.

PSOICSA Algorithm steps are defined as follows.

(1) Identify the Population Size. The populace is made of a specific number of people or individual particles.
(2) Fitness is a list of individual quality. As a rule, a substantial fitness esteem relates to a decent outcome, and the other way around.
(3) Identify minimum and maximum values in search space for the optimization problem.
(4) Velocity is restricted as the calculation plays out an inquiry.
(5) An individual is updated its position and velocity by utilizing the procedure of Particle Swarm Optimization Algorithm.
(6) An individual in ICSA has no velocity updating formula, whereas an individual in Particle Swarm Optimization search mode has both position and velocity. velocity in the cuckoo seek mode is not changed, and the present velocity of the individual is the velocity resulted by Particle Swarm Optimization search mode.
(7) The arbitrary end instrument in improved cuckoo search algorithm, the host has likelihood of finding foreign eggs.

---

**Pseudocode for PSOICSA Algorithm**

```plaintext
Begin
Initialize particle
While (t<max Generation) or (stop criterion)
Get a cuckoo randomly
Evaluate its quality/Fitness
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pBest) in history
Set current value as the new pBest
End
While maximum iterations or minimum error criteria is not attained
Choose a nest among n (say, j) randomly;
if (F_{i}>F_{j})
Replace j by the new solution;
End
If a fraction (Pa) of the worse nests are abandoned and new ones are built;
Keep the best solutions/nests;
Rank the solutions/nests and find the current best
End
For each particle
Find in the particle neighbourhood, the particle with best fitness
Calculate particle velocity according to the velocity equation
Apply the velocity constriction
Update particle position according to the position equation
Apply the position constriction
End
End
```

**Fig.1: Block Diagram of Proposed method**
Colony Optimization), PSABC (Particle Swarm Optimization and Artificial Bee Colony), and PSOICSA (Particle Swarm Optimization and Improved Cuckoo Algorithm) and proved that PSOICSA performs better and efficient than other optimization methods.

4. Experimental Result

The Existing review works are compared in this section with the proposed work to show that our proposed work is better than the state-of-art works. We can establish that our proposed work helps to attain very good accuracy for the estimation of software quality database using Improved cuckoo search algorithm. And, we can establish this proposed accuracy outcome by comparing other existing work. In our proposed study introduced this can be a software quality estimation centered on Improved Cuckoo search, Particle swarm optimization algorithm. Hence the efficiency methods computation revealed which our Proposed method is successful than the Existing method.

The analysis is executed in MATLAB. The experiment optimization technique essential assessment is to have most extreme number of issues secured and explanation secured with least number of testcases required ad identified the required testcases to execute in less time by prioritizing the testcases. In this approach, time taken to execute each testcase is analysed. In this case, there are test suites formed with number of test cases (TS) = |TS1, TS2, TS3, TS4, TS5| and the faults covered are (FC) = |FC1, FC2, FC3|. So also, the announcements secured by the experiments are signified as Statements Covered (SC) = |SC1, SC2, SC3, SC4|.

Below figure illustrated the comparison of the proposed PSOICS approach with the other advancement methodologies, like for example: PSO, ACO, PSABC, CS Comparison is for statement coverage (SC) and fault coverage (FC). Table 1, depicts the comparisons of the PSO (Particle Swarm Optimization), ACO (Ant-Colony Optimization), PSABC (Particle Swarm Optimization and Artificial Bee Colony), and PSOICSA (Particle Swarm Optimization and Improved Cuckoo Algorithm) and proved that PSOICSA performs better and efficient than other optimization methods.

5. Conclusion

Testing guarantees that the product meets the client conditions and necessities. Elements like exertion, time and cost of the testing are elements impacting test cases generation, optimization and prioritization. Several research works have been proposed in the writing for experiment prioritization. The fundamental go for prioritization of experiments is to limit the cost and time of relapse testing. Aim of this research work is to attain test case optimization and prioritization results using PSOICSA. PSOICSA has the upsides of quick union speed, solid looking capacity, and the capacity to take care of the issue of multi-dimensional constant space advancement by utilizing test capacities. A test illustration demonstrates that PSOICS can viably take care of the upkeep time frame advancement issue in view of the proposed streamlining model.

Table 1: % of test cases/test data in terms of maximum fitness value

<table>
<thead>
<tr>
<th>Fitness Value Range</th>
<th>% of Test-cases (PSO)</th>
<th>% of Test-cases (ACO)</th>
<th>% of Test-cases (PSABC)</th>
<th>% of Test-cases (PSOICSA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0≤(f(x)&lt;0.3</td>
<td>20</td>
<td>15</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>0.3≤(f(x)&lt;0.7</td>
<td>30</td>
<td>25</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>0.7≤(f(x)&lt;1.0</td>
<td>40</td>
<td>35</td>
<td>17</td>
<td>14</td>
</tr>
</tbody>
</table>

References


