NATURE INSPIRED COMPUTING BASED ON PARTICLE SWARM APPROACH USING VARIATIONS IN INERTIA WEIGHTS FOR AUTOMATIC TEST DATA GENERATION APPROACH WITH SPANNING TREE CONCEPTS

Sanjay Singla¹, Dr. Raj Kumar² & Dr. Dharminder Kumar³

Abstract-The weight is allotted to all or any edges of diagram as per a consistent arrangement and concentrate is on the leaves hubs of the Spanning Tree of control Flow Graph of Program. On the off likelihood that every one leaves hubs of Spanning tree area unit secured by the experiments, at that time variety of experiments needed or testing the program may be minimized to extraordinary degree. Assist the spreading over tree depends on the thought of greatest crossing tree of CFG that helps find the fundamental ways of CFG. the chances of discovering mistakes in basic area unit a lot of as distinction with the standard means. Covering of the considerable variety of leaves hubs of Spanning Tree offer guarantee for the scope of crucial means too. Assist the paper to extra demonstrates that variety of experiments needed if there ought to be a happening of Spanning tree is a smaller amount as distinction with the Dominance tree plan. PSO (Particle Swarm Optimization) - a Nature roused calculation by applying distinct dormancy weights in PSO, one of a kind, dependable and effective experiments are created to cover the leaf hubs of traversing tree. Latency weights like settled dormancy weight (FIW), worldwide nearby best (GLbestIW), Time-Dependent weight (TDW), and proposed GLbestRandIW weights are utilized with PSO to explore the impact of idleness weights on the execution of PSO regarding number of age required, rate scope , add up to test cases created to test the product under thought.

Keywords: Testing, Particle Swarm Optimization (PSO), Inertia Weight, Spanning Tree, Dominance Tree

1. INTRODUCTION

The most difficult and tedious method in programming life cycle that devours price of programming advancement method making an attempt in programming. The foremost difficult issue in take a look at is to find a considerable and solid test info methodology. To fulfill a given testing criteria, varied information test should be made that needed bunches of your time and endeavours. the foremost basic enterprise in programming testing method is that the outlining of experiments for the fulfillment of given testing criteria [26]. they're distinctive type of procedures or ways purposed by varied appearance into/others each once in a very whereas to make take a look at info for programming testing [5]-[8], [10]-[13], [30]. The cost, time and endeavours for designing various take a look at in packaging for testing the merchandise is slashed to substantial broaden if the manner toward testing is completed naturally while not inclusion people. There are completely different code Computing Technique like Swarm Intelligence, Genetic calculation, Evolutionary Programming and then on that has demonstrated their impact and impact within the field of programming testing [25]-[26]. The execution of these code Computing algorithmic program is noteworthy to expansive degree nevertheless time taken by these calculations is usually ache for investigation and abusing the promising reasons within the inquiry house [31].

Hereditary Algorithm (GA) depends on Darwin's hypothesis of Natural Evolution indicated in the starting point of flavours. It was purposed by Holland in 960. It depends on the idea of populace of chromosome where is chromosome speaks to a comparing arrangement. These arrangements/chromosome experience different operations to create people to come. The operations utilized as a part of Genetic Algorithm are hybrid that change administrator that causes the GA advance through age to age. Hereditary calculation is broadly utilized as a part of many Engineering and Optimization issue and it has demonstrated its reality and impact yet it is caught in nearby improvement and it is tedious. Kennedy and Eberhart have suggested another populace implied improvement system named as (PSO) Partial Swarm Optimization in 1995. PSO is energized by collective conduct of winged animals running and Schooling of fish [17]. That fresh calculation have many remarkable component which improves this calculation more method as contrast with the Genetic calculation. In the 1st
place, PSO utilized plan of memory that is not accessible in Genetic rule. It keeps the info of each molecule best position, that they accomplished from the start stage of their journey and also the worldwide best position among each one of the particles. owing to this memory section, PSO moves speedier towards the target. this type of knowledge is not place away anywhere in Genetic calculation. Second, it's simple and usage PSO as distinction with Genetic calculation. Third, there's information sharing among the particles in PSO through productive participation that is deficient in Genetic calculation [33]. Later PSO execution betters as distinction with the Genetic calculation [22]. There square measure distinctive variety of analysis papers that demonstrate that execution of PSO for unraveling numerous reasonably programming testing problems is desirable or is equally nice over the execution of Genetic calculation. This paper deals with a profitable PSO calculation for programming testing and analyzed the impacts of weight of dormancy varieties. The proposed procedure gives the results better rate scope and less number of ages.

2. BACKGROUND
This unit depicts the essential idea which must be utilized in paper all over.

2.1 (CFG) Control Flow Graph
CFG is a coordinated chart of program's. In CFG, each hub speaks to a gathering of sequential proclamations, which together speak to an essential piece and each edge speaks to development of organize stream among the vertices. To fabricate a CFG initially construct essential squares and afterward we include edges that speak to control stream between these fundamental pieces. Figure 1 whose CFG is appeared in figure 2 defines the program.

```c
#include<stdio.h>
void main ()
{
   int a, b , c;
   a= b= 0;
   scanf("%d", &c);
   while (c <> 0)
   {
      if ( c %2 = = 0)
      {
         a = a + 1
      }
      else
      {
         b = b +1 ;
      }
      scanf("%d", &c);
   }
   printf  ("%d    %d",  a , b);
}
```

Fig. 1. Program 1

A plotted diagram or Digraph is spoken to as G (V, E) whereas V is the arrangement of vertices define by V= \{v1, v2 , ……\}, and E is the arrangement of edges given by E = \{e1, e2 ,… .\}, and a mapping that maps every edge onto several requested match of vertices (vi, vj).
2.2 Weighted Graph
A diagram $G$ is alleged to be a weighted chart if its edges are assigned a weight. Weight is parcelled out to edges of the chart. Within the 1st place, range of Level is tallied and then weight is started as $2n$, where $n$ is the amount of Levels. Within the figure 2, there are five levels thus weight is started as twenty five = thirty two. This weight bifurcate on the selection hub, seventy fifth percent of alternative hub weight moves towards the branch that underpins the real choice of predicate hub whereas twenty fifth moves towards the inverse facet. The burden for the CFG of Figure a pair of is appeared as in fig 3.

2.3 Adjacency Matrix of a Diagraph
Adjacency matrix provides one of the ways to represent any graph having weighted edges. When two vertices have been involved to form an edge then that entry can be reflected as "1" in the matrix, otherwise the entry can be represented as "0". All the entries representing an edge of the matrix carry some weight. This weight may represent some kind of distance, cost, value etc. as per the application.

If $[x_{ij}] = w$

Above equation shows the existence of an edge between the vertices $i$ and $j$ and carrying the weight as "$w$". Figure 4 in the following text representing the adjacency matrix of Fig 3.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4: Adjacency matrix

2.4 Spanning Tree
The major concept behind crossing tree is that, it is a sub graph of any graph $G$; and it must represent all the vertices of main graph $G$. A crossing tree may be a quite skeleton of distinctive Graph $G$. Therefore Spanning tree is once in an exceedingly whereas alluded to as a skeleton or framework of $G$. A crossing tree with the smallest weight within the weighted diagram is thought as a most restricted spreading over tree or briefest separation traversing tree or negligible crossing tree. Associate insignificant spreading over tree in associated weighted diagram may be a traversing tree that has the smallest conceivable combination of weights of its edges. During this paper the modified Prim formula is employed to accumulate negligible traversing tree, but instead of insignificant spreading over tree, most extreme crossing tree (spreading over tree having most astounding weight) is needed. As Prim formula has two constraints, 1st it is not acceptable if weights square measure negative and second it's used only for rudderless chart but not for coordinated diagram. For greatest Spanning tree, 1st nearness framework is duplicated with - one

Adjacency Matrix $X_1 = (\cdot 1) \times X$ is represented as:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-8</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-18</td>
<td>-6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-18</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>-24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

After that altered Prim calculation is connected. As Prime calculations begins from the primary vertex $v1$ and interface it to
its nearest neighbor (i.e., to the vertex that has the smallest passage in push one of the table), say \(v_k\) currently think about \(v_1\) and \(v_k\) together sub-diagram and associate this sub-chart to its wardrobe neighbor (i.e., to the vertex aside from \(v_1\) and \(v_k\) that has the smallest section among each one of the sections among columns one and \(k\)). Rehash this procedure until all \(n\) hubs or vertices are associated by \(n-1\) edges.

The highlighted price is chosen by applying Prim algorithm is shown as:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & -32 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -24 & 0 & 0 & 0 & -8 \\
3 & 0 & 0 & 0 & -18 & -6 & 0 & 0 \\
X1 = & 4 & 0 & 0 & 0 & 0 & 0 & -18 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & -6 & 0 \\
6 & 0 & -24 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Other than the bold entries of \(X1\), all the entries becomes zero.

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & -32 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -24 & 0 & 0 & 0 & -8 \\
3 & 0 & 0 & 0 & -18 & -6 & 0 & 0 \\
X1 = & 4 & 0 & 0 & 0 & 0 & 0 & -18 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

The rows are leave nodes with zero values, shows the main issue of this research. Minimal and maximum spanning tree have been represented by Fig 5 and fig 6 respectively.
2.5 Critical Path
Basic way has a noteworthy part in programming testing in light of the fact that a great deal of blunder falls in the basic way. On the off chance that basic way is tried appropriately then the odds of happening of blunders in programming can be decreased to expansive degree in light of the fact that the basic way has more opportunities to trap mistakes amid programming testing. In the most extreme crossing tree, the way having most astounding weight is basic way. These ways must be broke down amid testing. The route is 1-2-3-4-6 which is appeared via dim mark.

2.6 Reduce the Test Cases
The primary point of research is to include all announcement of method and discover most extreme mistakes with least number of experiments. This should be possible by including a subset of articulations that ensures the scope of all announcements of the tried program [29]. the basic concern is that the entire arrangement of leaves hubs of greatest spreading over tree as every arrangement of manner that spreads it likewise covers all hubs in tree. The arrangement of leave hubs for figure half-dozen is as $X = \{6, 5, 7\}$. The manner of each single element of set $X$ is appeared as
- Path (6) = 1,2,3,4,6 - Critical path
- Path (5) = 1,2,3,5
- Path (7) = 1,2,7

Advance the covering of Path (6) covers consequently the fundamental approach that is additional inclined to blunder. All hubs of the CFG (1, 2, 3, 4, 5, 6, seven hubs of CFG appeared in fig 2) may be secured by covering the approach of each single element from the arrangement of leaf hub of most extreme crossing tree. Covering of each last hub of CFG implies covering of all announcements of program and this is often principle purpose behind testing of programming

The predominance tree of CFG of fig two is appeared in fig seven. The amount of leaves hubs for Dominance Tree in figure five is four i.e. $\{4, 5, 6, 7\}$ whereas for traversing tree, the amount of leaves hubs is three $\{5, 6, 7\}$. any as distinction with the testing of leave hubs of strength tree planned by A S Ghiduk [29], the amount of experiments needs is a smaller amount in utilizing most extreme spreading over tree, on these lines decrease the push to an enormous degree.

3. PARTICLE SWARM OPTIMIZATION (PSO)
Kennedy and Eberhart developed an algorithm called Particle Swarm Optimization [14] that replicates the natural tendency of birds or fishes during their food discovery or new appropriate habitation. Consider a d-dimensional search space in the basic PSO technique. Every member is considered as a particle. Each particle is shown by d-dimensional positional vector along with description as $X_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$

1. A Population is an ordered set of particles in the swarm which is expressed as $pop = [x_1, x_2, \ldots, x_d]$.
2. $pBest$ is considered as the previously best value of every particle. This is expressed as $PB_i = [pb_{i1}, pb_{i2}, \ldots, pb_{id}]$
3. $gBest$ is considered as global best for each particle and can be calculated as $GB_i = [gb_{i1}, gb_{i2}, \ldots, gb_{id}]$
4. The term Velocity defines the change in the position of each particle and is expressed as $V_i = [v_{i1}, v_{i2}, \ldots, v_{id}]$
When number of iterations is "k" then the velocity of \( i \)th particle is expressed as:

\[
v_{id}(k+1) = wv_{id}(k) + c_{1}r_{1}(pb_{id}(k) - x_{id}(k)) + c_{2}r_{2}(gb_{id}(k) - x_{id}(k))
\]

(1)

Where \( i \) varies from 1 to \( n \). Here, \( n \) is the size of each population, inertia weight is denoted by \( w \), \( c_{1} \) along with \( c_{2} \) are constants. \( r_{1} \) and \( r_{2} \) represents random variables having scope \([0,1]\).

5. The position of particles are expressed with the help of following equation:

\[
x_{id}(k+1) = x_{id}(k) + v_{id}(k+1)
\]

(2)

3.1 Considerations for Inertia Weight

3.1.1 Fixed Inertia Weight (FIW)

The usual PSO algorithm at first utilized a steady or constant inertia weight.

3.1.2 Time Dependent Weights (TDW)

Remembering the true objective to improve the current framework, the time-varying idleness weight was suggested [34]. This latency weight straightforwardly lessens with respect to time. Generally, to start periods of the interest methodology, substantial inactivity weight to overhaul the worldwide investigation (looking new area) is recommended while, for end part, low idleness weight is proposed for neighborhood examination. Global-Local best Inertia Weight (GLbestIW)

The GLbestIW strategy is projected in [35]. It is considered as a function of local and global estimations of the particles in various generations. The equation for the same is given as:

\[
GLbestRandIW Wis= (1.1 - (gbesti/pbesti))
\]

(3)

3.1.3 Proposed global-local best random inertia weight (GLbestRandIW)

The changes in inertia factors may enhance the performance of this optimization technique (PSO). Here, inertia weight is considered as a function of pbest and gbest with random factor values of the particles in each generation.

\[
GLbestRandIW Wi= (1.1 - (gbesti/pbesti)) *z(Rand) + 0.5*(Rand)
\]

where \( Z = 4* (Rand) *(1 - Rand) \)

(4)

4. FITNESS FUNCTION

A welfare work is that the proportion of the {number|the amount} of secured hubs of the method of the target hub to the mixture number of hubs of the method of the target hub. every experiment is nourish to the program for execution and each one in every of the hubs secured by this experiment square measure recorded that is termed execPath. The wellness esteem \( f_{t} \) for every Individual/chromosome/molecule (\( I = 1, \ldots, S \)) in a populace having size 'S' is computed as: conclude path \( p(n) \): represents the reach of the target node.

1. Compute \((p(n) - \text{exec}_\text{Path})\) to identify the nodes not covered.
2. Compute \((p(n) - \text{execPath})'\) to identify the covered nodes/node
3. Find \(|p(n) - \text{execPath}'|\) to count the number of nodes covered.
4. Find \(|p(n)|\) to compute the number of nodes up to target node
5. Then

\[
f_{t}(vi) = \frac{|\text{pat h}(n)-\text{exec pat h}|}{|\text{pat h}(n)|}
\]

(5)

Every single chromosome is delineated by test suite and that must be optimum if its fitness worth is represented as "1". [29]. The fitness worth is barely technique of feedback to improvement algorithms like GA, SIMBO, PSO etc.

5. EXPERIMENTAL OUTCOME AND CONCLUSION

Examination is performed on generally utilized PC programs for evaluating the execution of recently suggested technique utilizing test information age methods such as GA and PSO with various weight of Inertia for both predominance idea and spreading over Tree idea. The distinctive techniques are appeared in Table I.

Table 1. Test Data Generation Methods

<table>
<thead>
<tr>
<th>Method Description</th>
<th>Method name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic Algorithm</td>
<td>GA</td>
</tr>
<tr>
<td>PSO with Fixed Inertia Weight (PSOFIW)</td>
<td>PM1</td>
</tr>
<tr>
<td>PSO include Time Dependent Weights (PSOTDW)</td>
<td>PM2</td>
</tr>
</tbody>
</table>
The execution of each of the three strategies GA, PM1, PM2, PM3, and PM4 is assessed utilizing the two ideas predominance Tree ideas and in addition Spanning Tree idea.

From the table II and figure eight, plainly planned strategy PM4 is superior to something alternative in variety ancient taken by calculation that is specifically recommendation to the time taken by calculation to seem, therefore PM4 is superior to alternative. Facilitate it's clear the number of ages needed in utilizing crossing tree plan is a smaller amount as distinction with strength tree plan. once more here execution of PM4 is superior to alternative PSO techniques like PM1, PM2, PM3 and GA.

**Table 2. TO COMPARE IN TERM OF NUMBER OF GENERATION**

<table>
<thead>
<tr>
<th>Prog No.</th>
<th>No of Generation</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spanning Tree notion</td>
<td>Dominance Tree notion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>PM1</td>
<td>PM2</td>
<td>PM3</td>
<td>PM4</td>
<td>GA</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>14</td>
<td>13</td>
<td>11</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>

**Fig. 8 performance in term of number of Generations of GA and various methods of PSO**

From the Figure nine and Table III, it's clear the utilizing crossing tree concepts the scope rate proportion is handy one hundred or best as distinction with the concept of strength tree, promote the amount of focalize proportion of PM4 (Proposed) is healthier as distinction with alternative PSO and GA within the two concepts (Dominance Tree and Spanning Tree).

**Table 3.Compare In Term Of Coverage Ration Percentage**

<table>
<thead>
<tr>
<th>Prog No.</th>
<th>Coverage percentage</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spanning Tree notion</td>
<td>Dominance Tree notion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>PM1</td>
<td>PM2</td>
<td>PM3</td>
<td>PM4</td>
<td>GA</td>
<td>PM1</td>
<td>PM2</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>95</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Fig. 9 Evaluate in term of coverage ratio percentage by by GA and different methods of PSO

By examination of figure ten and Table IV, the number of experiments created in Spanning Tree plan is a smaller amount as distinction with Dominance tree. the number of experiments created in planned technique PM4 in each plan as distinction with alternative 2 sweetening strategy i.e. GA and alternative PSO. As PM4 accomplishes 100% scope in less variety of emphasis as distinction with GA and alternative PSO that reasons that experiments made by PM4 or outstanding as distinction with alternative PSO and GA. typically speaking, it will reasoned that Spanning tree concepts is superior to something Dominance tree altogether viewpoints i.e. as so much as time utilization, age of 1 of a sort experiments, scope proportion rate, variety of ages so forth. Further, it will be inferred that PM4 with planned ways performs fantastic as distinction with totally different techniques utilised as an area of PSO and GA within the 2 concepts.

Table 4 Compare in term of number of test cases generated

<table>
<thead>
<tr>
<th>Prog No.</th>
<th>No of Test cases</th>
<th>Spanning Tree notion</th>
<th>Dominance Tree notion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
<td>PM1</td>
<td>PM2</td>
</tr>
<tr>
<td>1</td>
<td>125</td>
<td>98</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>99</td>
<td>63</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>25</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
<td>72</td>
<td>63</td>
</tr>
<tr>
<td>6</td>
<td>180</td>
<td>140</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>72</td>
<td>45</td>
<td>44</td>
</tr>
</tbody>
</table>
6. REFERENCES


