CHAPTER 5

PERFORMANCE OF ODD AND EVEN POINT CROSSOVER BASED TABU GA

5.1 ODD AND EVEN POINT CROSSOVER BASED TABU GA

The previous chapters explain the GA and Tabu Search. These two tools are optimization/search tools, but the fundamental difference made these two as different entities. The first one is a global search or optimization tool (Sanjay et al 2011), but the second one is a local search tool. The first one GA may or may not work with history, but the Tabu search entirely depends on the history. Without history one can’t find an optimal solution. In GA, history reduces the convergence rate and the search may struck with local optima (Wengdong Wang and Susan Bridges 2000). In other words the global search tool becomes a local search tool. The local search tool combined with the global tool to form a hybrid tool and the hybrid tool is more effective (Asha Gowda Karegowda et al 2011, Cho Sung-Bae 2002, Juang 2004). Hence, a hybrid search or optimization tool has been proposed with a powerful exploration operator.

The odd and even point crossover based GA has a powerful exploration mechanism. That powerful mechanism is an uncontrolled one. This uncontrolled nature may leads to a random search. Hence, it demands a control mechanism. The controlling action can be carried over with the use of some local search tools (Darrell Whitley et al 2001, Edmundo Bonilla Huerta
et al 2006). The local search tool merged with global search mechanism and forms a new hybrid search/optimization tool (Andal Jayalakshmi et al 2001, Carvalho and Freitas 2004, Dong Hwa Kim et al 2007). So, hybrid search tool becomes a combination of both local and global optimization mechanism (Ashwani Dhingra and Pankaj Chandna 1998). In our case, Tabu search is the local search which merged with a best global optimization tool namely GA (Panduro 2009). The inherent property of Tabu search controls the powerful exploration mechanism of our proposed odd and even point crossover based GA.

Tabu is a local search mechanism which entirely depends on Tabu tenure period. The conventional Tabu search algorithm has been modified in accordance with the global search mechanism to give a new hybrid search mechanism which can be tested over data fusion problem in information retrieval. The algorithm used for finding an optimal solution in data fusion is given Algorithm 5.1.

The algorithm is designed to find an optimal solution in the data fusion problem for information retrieval. The focus of this algorithm used to find a best fusion function, best retrieval strategies with appropriate weights. Sample solution and the fitness function are already described in the previous chapter. The genetic algorithm parameters used in this algorithm are same as that of previous chapter. The Tabu list is divided into three parts. The first portion devoted for data fusion function, and the second one for information retrieval strategies. The last one contains the weights for the retrieval strategies.
Initialize the number of Generations $G$
Initialize the population size $N$
Initialize the Tabu list
Initialize the Tabu tenure period
Randomly generate the initial Population $P(g)$
Evaluate $P(g)$
While (non termination condition)
do
{
    \text{g = g +1}
    \text{Select n chromosomes for $P(g)$ from $P(g-1)$}
    \text{While (}$P_c > 0.6$$\text{)}
    \text{Do}
    {
    \text{Selects two random parents}
    \text{While (two parents not in the list)}
    \text{If ($P_{odd} > 0.5$)}
    \text{Select a random location in odd group}
    \text{Recombine the two parents.}
    \text{Else}
    {
    \text{Select a random location in Even group}
    \text{Recombine the two parents}
    \text{}}
    \text{While ($P_m < 0.01$)}
    {
    \text{Select a random string}
    \text{While (a random string not in Tabu list)}
    \text{Select a random location}
    \text{}}
}
5.2 TABU TENURE PERIOD

The tenure period influences the optimal solutions. The tenure period is based on the Tabu list. As the Tabu list contains three parts, the tenure period also divided into three portions. The tenure period has to be selected carefully. In order to do so, the impact of model solution’s vital component is analyzed. In the sample solution, it is identified that, the fusion functions has more importance. Hence, it is planned to assign higher tenure period to fusion function. The next important component of the solution is retrieval strategies.

The retrieval strategies’ importance are not that much vital as that of fusion functions. Hence, it deserves lesser tenure period in compression with fusion function. The last component of the sample solution is weights of the retrieval strategies.

The weights have to operate over the selected fusion function and the retrieval strategies. It can’t act alone. Hence, the weights are placed in the last position based on the level of importance.

Based on these assumption retrieval strategies demands the lowest tenure period of all the three. The tenure periods are selected accordingly. Two options are available. The first one is higher tenure period. Second one is the lower one. A higher tenure period is opted then the exploration become more controlled. So we opted for a lower tenure period. The selected tenure period for the algorithm is given in the following Table 5.1.
Table 5.1  Tabu tenure period

<table>
<thead>
<tr>
<th>S.No</th>
<th>Name</th>
<th>Tabu tenure we period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fusion Function</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Retrieval Strategies</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Weights</td>
<td>0</td>
</tr>
</tbody>
</table>

The Tabu tenure period is the restriction imposed over the individual components. During this period, the components are neither deleted nor included in the solution. The Tabu list and tenure periods are used as the exploitation tool.

5.3  EXPERIMENTS AND RESULTS

The experiments are conducted over the same data sets which are used in our previous experiments. The fusion functions retrieval strategies, and their weights are also same.

The experiments are divided into two half. The first one is 12 bit encoding, and the second one has 16 bit encoding. The results obtained for the first case is given in the Table 5.2.

Table 5.2  Tabu GA results for 12 bit encoding

<table>
<thead>
<tr>
<th>Collection</th>
<th>Precision</th>
<th>% of Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>0.3812</td>
<td>6.809</td>
</tr>
<tr>
<td>CISI</td>
<td>0.1923</td>
<td>5.66</td>
</tr>
<tr>
<td>CRAN</td>
<td>0.1437</td>
<td>4.74</td>
</tr>
</tbody>
</table>
The second half of the experiment has more bits than the first one. This means the weights of the retrieval strategies have been increased. The GA parameters and the Tabu list are same as that of the first case. The result for the second case is given in the Table 5.3.

Table 5.3 Tabu GA results for 16 bit encoding

<table>
<thead>
<tr>
<th>Collection</th>
<th>Precision</th>
<th>% improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>0.3817</td>
<td>6.95</td>
</tr>
<tr>
<td>CISI</td>
<td>0.1928</td>
<td>5.93</td>
</tr>
<tr>
<td>CRAN</td>
<td>0.1442</td>
<td>5.10</td>
</tr>
</tbody>
</table>

The above two Tables 5.2 and 5.3 give the consolidate results. The precision value given at the tables are recorded at the 100% recall level. These two tables are used to measure the performance of data fusion. It shows the improvement in performance for the Tabu GA with respect to data fusion problem. It can be confirmed by comparing the tables 5.2 and 5.3 with 3.4 and 3.5 respectively. In order to understand the impact of Tabu GA, it is necessary to analyze the 11 point precision.

The 11 point precision used to analyze the precision value at various recall levels. The following graph shows the values for all three data collection and it compares the conventional GA’s performance with that of Tabu GA. From the two Tables 5.2 and 5.3, it is confirmed that, Tabu GA’s overall result is much better than the conventional GA. But it needs to be analyzed, the Results on point to point basis. It is given in the following Figures 5.1 to 5.6.
Figure 5.1  Performance comparison of Tabu GA and conventional GA for 12 bit encoding over ADI data set

Figure 5.2  Performance comparison of Tabu GA and conventional GA for 12 bit encoding over CISI data set
Figure 5.3  Performance comparison of Tabu GA and conventional GA for 12 bit encoding over CRAN data set

Figure 5.4  Performance comparison of Tabu GA and conventional GA for 16 bit encoding over ADI data set
Figure 5.5  Performance comparison of Tabu GA and conventional GA for 16 bit encoding over CISI data set

Figure 5.6  Performance comparison of Tabu GA and conventional GA for 16 bit encoding over CRAN data set
5.3.1 Performance Comparison of 12 Bit and 16 Bit Encoded String in Tabu GA

The results chapter proffers the result and comparison of Tabu GA and conventional GA. It doesn’t shows the performance comparison of 12 bit and 16 bit encoded string over Tabu GA. This subsection gives the overall performance of Tabu GA and their comparison at 11-recall points. The comparison has been given in following Figures 5.7 to 5.9.

Figure 5.7 Performance comparison of Tabu GA for 12 and 16 bit encoding over ADI

Figure 5.8 Performance comparison of Tabu GA for 12 and 16 bit encoding over CISI
Figure 5.9 Performance comparison of Tabu GA for 12 and 16 bit encoding over CRAN

The graphs show the precision at 11 point recall level. At each levels, there is no significance difference. The performance of both 16 bit and 12 bit seems to be the same.

The main difference between the 12 pt and 16pt is, the number of bits used for representing the weights of retrieval strategy. The increased weights for the retrieval strategies doesn’t produce significant improvement in overall performance. Hence, it is identified that, the performance of the data fusion problem depends on the fusion function and retrieval strategies. Hence, it concludes that, 12 bit encoded string is more than sufficient to carry out the experiment. The final conclusion at this point is, the role of weights for retrieval strategies are not significant as that of the other parameters like fusion function, retrieval strategies.

5.3.2 Statistical Analysis

The Tables 4.1, 5.2 and 5.3 gives the overall performance of both conventional GA and Tabu GA. From the table itself, it could be confirmed. the superiority of Tabu GA. Even though Tabu GA seems to be better. The
same is to be confirmed. For this purpose, the Student-T test is used. The performance of the conventional GA and the Tabu GA for both 12 and 16 bit representation analyzed using student-T test.

The hypothesis used for t test is given below,

\[ H_0: \mu_0 = \mu_1 \] - There is no significant variation in performance between conventional GA and Tabu GA.

\[ H_0: \mu_0 \neq \mu_1 \] - There is a significance difference in performance between conventional GA and Tabu GA.

The following Table 5.4 gives the T value over all three data sets.

**Table 5.4 Calculated T-value**

<table>
<thead>
<tr>
<th>Collection</th>
<th>T - Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12 bit</td>
</tr>
<tr>
<td>ADI</td>
<td>5.63</td>
</tr>
<tr>
<td>CISI</td>
<td>4.87</td>
</tr>
<tr>
<td>CRAN</td>
<td>7.07</td>
</tr>
</tbody>
</table>

The null hypothesis rejected at 1% confidence level and claim proved to be correct using the T-Test.

5.4 CONCLUSION

This chapter analyzed the performance of Tabu GA. This hybrid algorithm proves to be better. The precision as a performance indicator indicates a significant improvement for Tabu GA over the conventional GA. The characteristic of Tabu GA is not yet explored. Next chapter devoted for this purpose.