CHAPTER 1

INTRODUCTION

In this chapter a humble attempt is made to analyze the basic concepts of information retrieval, data fusion, Genetic Algorithm, and Tabu search. The problem description and the objective of our proposed work are given at the end of this chapter. The last part of this chapter gives the overall outline of this thesis.

1.1 INFORMATION RETRIEVAL

Digital era has made the publishing process easier than never before. The digital media contains text and multimedia (Michael Lew et al 2006). The text data exist from ancient days. This leads to the development of library science. The concept of library science has been borrowed and modified for the autonomous system, which can handle the digital text data. The focus towards the autonomous system leads to new area called Information Retrieval (IR) (Amit Singhal 2003, Amanda Spink Tefko Saracevic 1997, Ed Greengrass 2000, Jin and Rahmat-Samii 2007).

The process of retrieving the required relevant documents from the collection (corpus) is termed as Information Retrieval (Christopher Manning 2009, Salton,G. and Mc Gill 1983, Yates and Neto 1999). The retrieval schemes aid the process by defining the various stages involved. Indexing and matching are the two main stages. The indexed words are matched against the user query for testing the relevance. The matching process uses a separate

Indexing and matching have been vividly described by various IR models. The models are equipped with a unique indexing mechanism and they are having various matching/similarity measures. The models and matching process are loaded with advantages and drawbacks.

1.1.1 Information Retrieval Models

Various models have been proposed to effectively carry out the process of information storage and retrieval (Cheng Xiang Zhai 2008). These models explain the method of storing the keywords or index terms (Donald Metzler and Bruce Croft 2006). These processes transform the documents into a suitable format such that storage and retrieval becomes easy (Peter Ingwersen 1992, Rabia Nuray and Fazli Can 2006). As a result of this, method of identifying the documents becomes easier. Model is a set of premise and an algorithm for ranking documents with regard to a user query. More formally, an IR model is a quadruple \([D, Q, F, R(q_i, d_j)]\), where \(D\) and \(Q\) is a set of logical views of documents and queries and \(R(q_i, d_j)\) is a ranking function which associates a numeric ranking to the query \(q_i\) and the document \(d_j\) and \(F\) is the framework for modeling document and queries. Strategy or scheme is synonymous with rank \(R(q_i, d_j)\).

System refers to the physical implementation of an IR algorithm which can have various operational modes or various settings of parameters. Therefore the same IR system may be used to execute different IR schemes by adjusting the various parameters.
Number of models are available in the literature. Out of these, Vector Space Model (VSM), and P-norm model play an important role. Hence, the next few pages are dedicated for the VSM and P-norm model explanation.

**Vector Space Model**

VSM is the most popular model used in the IR process. VSM not only explains the process of retrieving relevant documents but also explains the assignment of rank to the documents. In VSM, the objects of IR, such as term, document, and query are treated as a multidimensional linearly dependent vectors in the vector space.

In the vector space model, the weight \( \omega_{td} \) associated with the index term ‘t’ in a document ‘d’ is positive and non-binary. Further, the index terms in the query are also weighted. Let \( \omega_{tq} \) be the weight associated with the pair [t,q], where \( \omega_{tq} \geq 0 \). Then the query vector \( q \) is defined in Equation (1.1).

\[
q = (w_{1,q}, w_{2,q}, w_{3,q}, \ldots, w_{t,q}, \ldots, w_{n,q}) \tag{1.1}
\]

where,

\[
w_{t,q} = \text{weight of the index term ‘t’ in query ‘q’}.
\]
where,

\[ N = \text{total number of documents in the corpus and} \]

\[ n = \text{total number of index terms in the corpus.} \]

The scalar product between the query and document vectors is used to calculate the relevance (similarity) of the document with respect to the query. In the vector space \( V \), if there exist two vectors \( x \) and \( y \) such that \( x, y \in V \), then the scalar product is defined in Equation (1.2).

\[
S(x, y) = |x| |y| \cos \theta
\]  
(1.2)

where,

\[ |x|, |y| = \text{magnitude of the vectors,} \]

\[
|x| = \sqrt{\sum_{i=1}^{n} x_i^2}
\]

\[
|y| = \sqrt{\sum_{i=1}^{n} y_i^2}
\]

and

\[ \theta = \text{angle between the two vectors.} \]

The vector space that considers only the scalar product is termed as the Euclidean space. The scalar product is used as one of the method to calculate the correlation between the query and the document vectors. There are various methods available to calculate the correlation (similarity) value. Based on the value of the correlation between the query and the document, the relevance of the document is justified. The retrieved documents arranged in descending order based on the value of their similarity.
Similarity Measure of VSM

The similarity between the document and the query is measured using the similarity measures (Zobel, J. and Moffat, A. 1988). There are various similarity measures available in VSM and few of them shown in Equations (1.3) - (1.6).

Cosine similarity measure
\[
S(q,d) = \frac{\sum_{i \in s \cap d} (w_{q,i} * w_{d,i})}{\sqrt{\sum w_{q,i}^2} \sqrt{\sum w_{d,i}^2}}
\] (1.3)

Inner product
\[
S(q,d) = \sum_{i \in s \cap d} w_{q,i} * w_{d,i}
\] (1.4)

Dice coefficient
\[
S(q,d) = \frac{2 \sum_{i \in s \cap d} w_{q,i} * w_{d,i}}{\sum w_{q,i} + \sum w_{d,i}}
\] (1.5)

Jaccard coefficient
\[
S(q,d) = \frac{\sum_{i \in s \cap d} (w_{q,i} * w_{d,i})}{\sum w_{q,i} + \sum w_{d,i} - \sum (w_{q,i} * w_{d,i})}
\] (1.6)

where,

\[S(q,d)\] = Similarity score of document ‘d’ with respect to query ‘q’.

\[w(q,t)\] = Weight of the term ‘t’ in the query ‘q’.

\[w(d,t)\] = Weight of the term ‘t’ in the document ‘d’.

\[W_q\] = Weight of the query ‘q’ and,

\[W_d\] = Weight of the document ‘d’. 
**Extended Boolean Model (P-Norm Model)**

Boolean model is very simple and it operates on the principle of Boolean algebra. It retrieves documents based on word matching function. Since the decision space in Boolean Model is binary, documents are judged as either relevant or irrelevant. Thus the number of documents retrieved as a result of the Boolean nature of the model is either vast or too small. Also there is no provision for the ranking the documents. These limitations are eliminated by extending the Boolean Model with the functionality of partial matching and term weighting. This extended model combines the advantages of the Boolean model and the VSM.

The Extended Boolean Model (EBM) was introduced by Salton, G. in 1983. In EBM, the weights assigned to the term lie between zero and one. Maximum normalization method is frequently used and the normalized weights are assigned to the index terms. The function used for maximum normalization is given in Equation (1.7).

\[
\text{Normalized } w_{i,j} = \frac{\text{Unnormalized } w_{i,j}}{w_l} \quad (1.7)
\]

where,

\[w_{i,j} = \text{weight of the term } i \text{ in } j^{\text{th}} \text{ document and}
\]

\[w_l = \text{maximum weight of the generic index term } l \text{ in the corpus.}
\]

The weight assignment techniques in the EBM are same as that of VSM with the only difference being that, the weights are normalized. The matching function or similarity measure is adapted from the Boolean Model. In the Extended Boolean Model, a query is represented in one of the following forms:
1) Conjunctive form

2) Disjunctive form and

3) Combination of both conjunctive and disjunctive form.

Let us consider a query, which contains two terms. The two terms may possess different weight. After the maximum normalization, the maximum weight is ‘one’. As the query consists of two terms, it’s maximum and minimums weights are (1,1) and (0,0) respectively. If the document contains the two terms its weight is (1,1) and it is more relevant to the query. If not, the document weight is (0,0) and it is irrelevant to the query. Hence in the disjunctive form, the distance is calculated from (1,1) and in the case of conjunctive form, it is measured from (0,0). The distance measure is not restricted to Euclidean distance but generalized to any value from 1 to $\infty$. Since the EBM depends on the value of p (distance) to calculate the similarity value, it is also referred to as P-norm model. The generalized form of the query in conjunctive and disjunctive form is represented in Equations (1.8) and (1.9) respectively.

$$q_{or} = w_1 \lor^p w_2 \lor^p \ldots \lor^p w_m$$  \hspace{1cm} (1.8)

$$q_{and} = w_1 \land^p w_2 \land^p \ldots \land^p w_m$$  \hspace{1cm} (1.9)

Where

- $m$ - No of terms
- $p$ - The distance value
Similarity Measures of P-Norm Model

The similarity measure between the document and the query in the P-norm model is given in Equations (1.10) and (1.11).

\[
Sim(q_{pe}, d_j) = \left( \frac{w_1^p + w_2^p + \ldots + w_m^p}{m} \right)^{1/p}
\]  

(1.10)

\[
Sim(q_{end}, d_j) = 1 - \left( \frac{(1-w_1^p) + (1-w_2^p) + \ldots + (1-w_m^p)}{m} \right)^{1/p}
\]  

(1.11)

where,

\(w_m = \) weight of the index term, and \(1 \leq p \leq \infty\).

1.1.2 Performance Measures

Effectiveness of the retrieval systems are calculated by using number of binary performance measures. They are i) Precision, ii) Recall, iii) Fallout, and iv) Generality. Among these measures Precision and Recall are most widely used.

Precision and Recall

Consider a retrieval system, which proffers the germane articles from a test collection (corpus). Let \(R_1\) be the set of relevant documents and \(R_2\) be the set of retrieved documents from the corpus and the scenario is explained with the help of venn diagram shown below. From the Figure 1.1 Precision and Recall may defined as follows:
Figure 1.1 Precision and recall for a given query

**Precision**

It is defined as the proportion of the documents that are relevant.

\[
\text{Precision} = \frac{R_w}{R_f}
\]

**Recall**

It is defined as the proportion of the relevant documents that are retrieved.

\[
\text{Recall} = \frac{R_w}{R_e}
\]

**Precision and Recall Curve**

The value of Precision and recall becomes a point in the graph when the retrieval system continuously retrieving the document for the given query. The graph is termed as precision recall graph and it gives the information about the underlying retrieval system. The Figure 1.2 shows the sample graph. The precision and Recall values are bounded to 100%.
The graph shows the curve of two retrieval systems $S_1$ and $S_2$. The performance of $S_1$ is higher than $S_2$ at lower recall level and at higher recall level the $S_2$ dominates $S_1$ (Salton, G. and Mc Gill, M.J.1983).

**Interpolated Precision**

Figure 1.2 is not a smooth one, and it gives the precision value only at the corresponding recall level. The interpolated precision gives the smooth curve as the precision values are interpolated at 11 standard recall points. The standard points are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 and the curve is called as 11-pt interpolated precision.

**1.2 FUSION**

Fusion is the methodology of combining retrieval strategies associated with the retrieval task followed by an assignment of relevance score or rank to documents on the basis of the score returned by the fused strategies (Rabia Nuray and Fazli Can 2005, Gendreau et al 1999). Fusion
techniques can be implemented with or without using the training data. By fusing more number of sources, it taps the merits of all participating members and diminishes their drawbacks (Clement et al 1993, Hugh Durrant-Whyte 2001). The sources are combined in its weighted and non-weighted forms in a centralized or distributed environment (Girija et al 2000), Hasan Demirel and Gholamreza Anbarjafari 2000).

1.2.1 Types of Fusion

Fusion methods are broadly classified into: (i) Data fusion and (ii) Collection fusion (Kokar et al 1999). A detailed classification of fusion techniques is shown in Figure 1.3 (Hall D.L. et al 1991, Jens Bleiholder and Felix Naumann 2008).

![Figure 1.3 Types of fusion techniques](image)

Collection Fusion

In Collection fusion the same query is operating on the various document collections. The relevant documents returned from the multiple corpuses are merged together to give the final relevant document list.
**Data Fusion**

The data fusion approaches combines the results obtained from various retrieval strategies over the same document collection or corpus (Daniel Dailey 1996, Franklin White 1991). The data fusion may further classified into

1) Representation Fusion,
2) Query Fusion,
3) Method Fusion and,
4) System Fusion.

In IR the same document and the query can be represented by different weighting scheme. If the fusion operation merges the result of the various document representations then it is called as representation fusion and if the various query forms are fused then it is called as query fusion. There are various methods, which are used to retrieve relevant document and if these methods are fused together then it is called as method fusion. The results from the multiple systems are merged then it is termed as system fusion. Previous results show that the fusion methods have some positive impact on the effectiveness of the retrieval system. It also yields consistent results over all test document collections.

**Meta Search**

The on-line information have tremendous impact. The search engines are used to find out the necessary information from the internet (Kushchu 2005). The merging of results from more than one search engine prove to be effective and it is termed as “Meta Search”.

The meta search engine have well defined architecture and the performance of the such engines depend on the selection and the number of underlying engines (Don Koks and Subhash Challa 2002).

There are various selection methods available. The selected search scores are merged together. The scores are combined in non-weighted and weighted forms. The meta search not only used for the text retrieval but also used to retrieve multimedia information’s from the web. The proper display of the results along with the merging dominates the whole meta search process. For the past few years meta search comes in to the limelight of researchers.

1.2.2 Fusion Functions

The fusion functions assign final relevance score to the documents based on the participating sources (Luciano Alparone et al 2008). The assignment of final relevance score by the functions are based on basic set theoretic operations like union, intersection and arithmetic operations. The linear combination method assigns weights to the individual strategies (Roland Soong and Michelle de Montigny 2001). The final relevance score of a document assigned by weighted linear combination method is given in Equation (1.12).

\[ R(q,d) = \sum_{i=1}^{k} \theta_i E_i(q,d) \]  

(1.12)

where,

- \( \theta_i \) = Weight of the \( i^{th} \) retrieval strategy,
- \( E_i(q,d) \) = Relevance score returned by the \( i^{th} \) retrieval strategy and
- \( k \) = Number of retrieval strategies to be fused.
The weighted linear combination method has the limitation of requiring prior knowledge about the retrieval systems to assign the weights (Umakishore Ramachandran et al. 2000). This limitation is eliminated in Comb functions by treating all strategies equally.

The Comb functions for combining scores by treating all strategies equally have been proposed by Fox, E.A and Shaw, J.A 1994. The various Comb Functions used for combining scores is shown in Table 1.1.

<table>
<thead>
<tr>
<th>Function</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CombMIN</td>
<td>Minimum of Individual Similarities</td>
</tr>
<tr>
<td>CombMAX</td>
<td>Maximum of Individual Similarities</td>
</tr>
<tr>
<td>CombSUM</td>
<td>Summation of Individual Similarities</td>
</tr>
<tr>
<td>CombANZ</td>
<td>CombSUM % Number of non zero Similarities</td>
</tr>
<tr>
<td>CombMNZ</td>
<td>CombSUM X Number of non zero Similarities</td>
</tr>
</tbody>
</table>

1.3 **GENETIC ALGORITHM**

Engineering design and techniques are subject to certain constraints (Aman Aggarwal and Hari Singh 2005). There is a trade-off in all engineering design and techniques. The success of these methods or techniques lie in the selection of best possible solution (Abraham Evonf 2002). In other words, an optimal solution have to be selected if the best solution (Jin and Rahmat-Samii 2007). Can not be selected the solution is a function of variables. Those variables are called as decision variables. The solution may be a function of single variable or multi variables. The variable may be continuous or discrete. These bottle neck made this process as complex as ever before. There is a need for a mechanism to find an optimal
solution subject to these constraints. The method or mechanism which assist us to find the optimization solution is termed as the search/optimization methods (Eberhart, R. and Shi, Y. 2001).

The first ever optimization method depends on some mathematical concepts. The objective function processed using some well known mathematical principles and finally the optimal solution was found. The format of the objective function is given below.

\[
O(v_1, v_2, \ldots, v_n) = \max \text{ or } \min f(v_1, v_2, \ldots, v_n)
\]

where

- \(v\) - Decision variable
- \(n\) - Total number of decision variable.

In an optimization problem, if the next move is calculated based on the calculated gradient value (\(\nabla\)) of the fitness function, then the method is called gradient based method. This kind of optimization method has some limitation. The limitation of these kinds of optimization mechanisms are

1. It needs concrete mathematical knowledge
2. The goal has to be converted into a function
3. It is operating directly over the variable
4. Time to find the optimal solution depends on the nature of the variable

The aforesaid limitations demand some alternative optimization/search mechanisms. As the demand for optimization/search mechanisms are immense, few non-traditional optimization/search mechanisms surfaced out. These non-traditional mechanisms are not based on mathematical concepts.
Hence, it attracts every one. These non traditional mechanisms are inspired by some real life principles. As these things are inspired, they are good at some aspects only (Yoshida et al 2000, Zhao Ping et al 2006). Hence, inspired non-traditional optimization techniques are classified into local and global search algorithm (Torn and Zilinskas 1989, Thomas Weise 2009). Depends up on our need, an appropriate method has to be selected from these available global search algorithm. Apart from this, the GA handles the multivariable and multi objective optimization problems effectively. (Abdullah Konaka 2006, Abramson and Abela 1992, Donald Metzler and Bruce Croft 2006, David Gold Berge 1989).

Genetic Algorithm (GA) inspired from the Darwin’s theory (Andrey Popov Hamburg 2005, Cao, Y.J. and Wu, Q.H 1999). It is inspired from the life. The survival of the fittest is it’s operating principle (Cheng-Lung Huang and Chieh-Jen Wang 2006). As it is inspired from life, there is no need for mathematical method to find the optimal solution except the fitness function. The fitness function derived from the objective function. The GA has following merits.

1. No need for mathematical background

2. It operates on the mapped variable rather than the variable itself.

3. The operation is more simple.

The GA uses three basic operators to find the optimal solution from the fitness function (Gang Wang et al 2006). The fitness function is derived from the objective function.

The objective function which involves with various decision variable may be a maximization or minimization function. The fitness
function of GA should be a maximization function. If the objective function is a maximization function then there won’t be any problem. If it is a minimization function then it is to be converted into a maximization function. The formal representation of the fitness function is

$$\text{fit}(v_1, v_2, \ldots, v_n) = \max f(v_1, v_2, \ldots, v_n)$$

The fitness function derived from the objective function (Ozgur Yeniay 2005). Hence, it may be a single variable or multi-variable function. The performance of the GA based on the fitness function (Ahmad Qasaimeh et al 2012, Mohammed Awad 2000). The fitness function and the decision variable and their range constitute the entire search space where GA has to operate. The GA has to explore the search space and it has to find the optimal solution (Poonam Garg 2009). In order to do so, GA equipped with three fundamental operators.

### 1.3.1 GA Operators

The GA has three basic operators namely

1. Reproduction
2. Crossover
3. Mutation

These three basic operators assist the GA to explore the search space to find an optimal solution (Osman Ahmed et al 2009). Exploration about these three basic operator are given in forth coming sub-sections as it seems vital.
Reproduction

The reproduction operator used to select the best individual present in a particular generation. It works on the principle of survival of the fittest. Hence, it selects the best solution. Consider the following example to understand the reproduction.

\[ O(x) = \begin{cases} \text{Max } (x)^2 \\ 0 \leq x \leq 10 \end{cases} \]

The objective function itself a maximization function. Hence there is no need for conversion to derive the fitness function. The fitness function is given as.

\[ f(x) = \begin{cases} \text{Max } (x)^2 \\ 0 \leq x \leq 10 \end{cases} \]

<table>
<thead>
<tr>
<th>S.No</th>
<th>X</th>
<th>( F(x) = x^2 )</th>
<th>( P(x) )</th>
<th>Expected Count</th>
<th>No of individual for the next generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>9</td>
<td>0.138</td>
<td>0.55</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0.0615</td>
<td>0.2461</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>16</td>
<td>0.2461</td>
<td>0.984</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>36</td>
<td>0.55</td>
<td>2.21</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ \sum f(x) = 65 \]
\[ \bar{x} = \frac{65}{4} = 16.25 \]

From the table, it is identified that, the highest fit individual has more representation. The above average individual has more representation and it will flourish further. It may lead to a scenario where a particular
generation filled with the multiple copies of the highly fit individual. It is the major draw back of the reproduction. In other words, the searching struck at the local optima and there are plenty of unexplored space left behind. It is the major draw back of reproduction. If reproduction alone is used, then GA becomes a local search. It won’t be a global search. There are various methods available to select an individual out of the particular generations. They are

1. Roulette wheel selection and
2. Tournament selection

To overcome the drawback of the reproduction, GA uses the crossover and mutation operators.

Crossover

The crossover is used as the exploration operator in GA. The crossover operates on the encoded string. In GA, the variable is not used in the direct format. The variables are represented in the encoded format. Three are various encoding mechanisms available. They are

1. Binary
2. Integer
3. Characters.

Out of these three formats, binary format is most famous. The variables are represented as a stream of ‘0’ and ‘1’. Consider our previous example, where there is one variable function.

$$fit(x) = \max(x^2)$$
If the 8 bit representation is chosen to use for encoding, then the variable \( x \) can be represented by using \( 2^8 \) combinations. After the representation, the real value with the encoded value has to be mapped. In our case the mapped value calculated as

\[
\text{Mapped value of } x = \text{binary equivalent value of } x \times \frac{x_{\text{max}} - x_{\text{min}}}{2^8 - 1}
\]

By using this convention, the mapped value of \( x \) can be calculated. Once, it is started operating in the mapped value, the crossover becomes much simple

In order to carry out the crossover operation, a parent has to be selected and two random string have to be selected and to be named as parent 1 and parent 2. In this example parent 1 and parent 2 are given below.

Parent 1 – 01001110
Parent 2 – 11110101

After selecting the parents, a crossover point has to be selected randomly. The selection of the crossover point is subject to a constraint. The selected crossover position should lies between 0 to \((l-1)\), where \( l \) is the total number of bits in the string. In this example, a position between 0 to 7 has to be selected. Assume that the position is selected as 3. After the crossover point selection, the bits between two parents have to be swapped with respect to the crossover point. As the bits are swapped it produces new children.

\[
\begin{align*}
\text{Parent} - 1 & \quad 0100 & \quad 1110 & \quad 01000101 - \text{Child 1} \\
\text{Parent} - 2 & \quad 1111 & \quad 0101 & \quad 11111110 - \text{Child 2}
\end{align*}
\]
This operation produces two children. If the structure of the children, is noticed one can easily identity one important thing. The structure of the two children’s are entirely different from the parent and it is a new one. Hence, a new point in the search space is successfully produced. These from of crossover operation is called as the single point crossover operation. This kind of crossover has one important drawback. These operations are inclined towards the end. Whatever may be the crossover position, the Least Significant Bit (LSB) will get disturbed. MSB remains unaffected. It is an important issue. The exploration entirely depends on the crossover operation. Stronger the disturbance more will be exploration. Hence a new kind of crossover operator developed. It is called as multi-point crossover. In the multi-point crossover, more than one crossover points are used. The bits which lies between the selected points are swapped to get a new children's. Consider our example. The crossover points are selected as 3 and 5.

```
Parent – 1  01  00  110
Parent - 2  11  11  0101
```

In this example, it is identified that, the drawback of single point crossover i.e. inclination towards the end has been eliminated.

The exploration is mainly depends on the crossover operation. Again, we have to analyze the relationship between the crossover and optimal solution. An optimal solution cannot be found without the exploration. If a vigorous exploration is then opted more time will be needed to find the optimal solution. This relationship is called as convergence. The convergence property is more vital for any search or optimization tool. The convergence analysis is used to measure the performance of the search tool. It can be analyzed by using the average fitness value and the difference between average fitness value among two successive generations. If it is in
the opposite direction, the convergence is not smooth and the exploration is more vigorous.

**Mutation**

During the convergence a single bit position may pose a threat. If the change the particular bit, then it may leads to a smooth convergence. This can be carried out using an operator called mutation. The mutation is a sudden change in the gene structure. This phenomenon used in GA. In GA the mutation is the sudden change in the single bit position. The mutation is carried out in the following method. First, a random string has to be selected from the pool of strings. Later the random position has to be selected. After that the content of that location has to be changed. In this example, the child - 1 produced by the single point crossover is selected for mutation.

Child – 1  01000101

After this selection random bit position has to be selected. Imagine that, the 5th position has been selected.

Child -1  01 [0] 00101.

As the content at the 5th position is 0, it has to be changed to 1. The final result after the mutation is

01100101

As the mutation is a rare phenomenon, this operator has to be used very rarely. In order to do so, some parameter should be used. These parameters are called GA parameters.
GA parameters

These parameters are used to control the overall GA operation. These parameters have some influence and it should properly set. The following parameters are used in GA. They are

1. Number of generations (G)
2. Number of individuals present in each generation (N)
3. Crossover Probability (p_c)
4. Mutation probability (p_m)

Out of these four parameters, the last two are probability value. These two values are lies between 0 and 1. The crossover probability value lies between 0.6 to 0.9. The mutation value should be very minimal. It's value varies from 0.001 to 0.1. As the mutation is rare phenomenon, the mutation probability is much minimal.

1.4 TABU SEARCH

Tabu search is the best local search method (Fermin Alfredo Tang Montane and Roberto Dieguez Galvao 2006) and it is also inspired from the real life. Taboo or Tabu means forbidden (Michel Gendreau et al 1999). It is inspired from the tribal people’s real life. It is invented by Fred Glover (Fred Glover 1995). It is the search mechanism, which uses memory for search operation (Somayeh Alizadeh et al 2002). The memory associated with the Tabu is a short term memory.

The simple Tabu search starts with the model solution. It directly operates over the decision variable, and the objective function. The Tabu search is a classical local search algorithm. The basic structure of the Tabu search consists of two parts. They are
1. Search space

2. Neighborhood structure.

1.4.1 Search Space

The search space is simply a space consists of all possible solution. The decision variable constitutes the search space. Every point in the search space represents the possible solution. The solution is made up of decision variables and they are subject to certain constraints. The solution in the search space may be a multi dimensional vector or a scalar. They form a set of solutions. The size of the set depends on the variable and their range and the constraints.

1.4.2 Neighborhood Space

The neighborhood space is closely related with the search space, while considering a model solution. The solution for an optimization problem is termed as the current solution. Let ‘S’ be the current solution. The point in the current solution may be surrounded by few more points. Such surrounding points may offer an optimal solution and those points can be reached from the current solution. The neighborhood point may be reached from the current solution by applying a single change. More formally, the neighborhood is defined as

\[ N(s) = \{ \text{solution obtained by applying a Single local transformation to } s \} \]

In general, for any specific problem, there are many possible neighborhood structure than the search space. This follows from the fact that there may be several plausible neighborhood structure for a given search space.
When the different definitions are considered for the search space, it leads to the different structure of the neighborhood. The neighborhood structure may be reached from the current solution by adding or dropping and swapping some variable from the current solution. During this process of addition, deletion and swapping, the neighborhood structure may struck with a cycle or loop. It prevents further exploration. In order to avoid this effect, one has to move towards a special structure, which is specifically designed for Tabu search. That special structure termed as ‘Tabus’.

1.4.3 Tabus

Tabus are one of the distinctive elements of Tabu search compared with any other local search. Actually Tabus are a short term memory which used to store some data during the search process. The main purposes of Tabus are to prevent the cycle or loop. The key realization here is that, when the cycling occurs, something needs to be done to prevent the search from tracing back its steps to where it came from. This is achieved by declaring Tabu moves that reverse the effect of recent moves. Tabus help us to avoid the already explored areas for effective exploration. The Tabus are stored as a list in the short term memory. That list is called as the Tabu List. The Tabu list may be a cyclic list and the entries are made and deleted in the cyclic i.e. first come first out basis. This kind of lists are more simple. Some times the problem demands a complex Tabu list. In such occasions multiple Tabu list has to be used for the effective operation. The separate list are used for the separate move.

The list which is used to store the neighborhood is called the Tabu list. The Tabu lists are stored in the temporary memory. It assist Tabu for finding the optimal solution. The size of the Tabu list depends on the type of
the problem and the number of variables involved in it. Standard Tabu lists are usually implemented as a circular list of fixed length. It has been proved that cyclic lists are not effective to prevent the cycling. Hence some researchers proposed the technique of varying the length during the search. Another solution is to generate Tabu tenure of each move with in some specified interval. If this method is adopted, a different technique has to be used to store the list in the memory. This is due to the fact that the conventional tag method is not effective for storing the list.

1.4.4 Tabu Tenure

The neighborhood are selected and stored in the list. Each entry in the list used to prevent the cycling. This can be achieved by imposing the restriction. The restriction is imposed in terms of prohibition for inclusion and exclusion. A variable once included in to the solution, it can’t be deleted for some prescribed number of iterations. Once it excluded from the solution, it should not be included for some fixed number of iteration. The fixed number of iterations for which the restriction imposed over the particular variable is called Tabu tenure period. Hence, the Tabu list consists of the list of variables along with their tenure periods. The value for the tenure period is more important. The value for the tenure period has to be selected in such a careful manner. If its value is higher, than it restrict the exploration. If it’s value is minimum then it restrict the exploitation. Hence, there should be a trade-off while selecting the tenure value. The exploration and exploitation are called as intensification and diversification in Tabu search.

1.4.5 Intensification

The idea behind the intensification is developed from the human life. In other words more importance is given to the recent past rather than
the long past. The short term memory is used. The same principles have been adopted for the intensification process. The intensification uses a separate short term memory. The short term memory used to store the recently visited solution. The recently visited solution are explored more vigorously and it is termed as the exploitation

1.4.6 Diversification

This used to explore the unexplored areas. This can be achieved with the help of long term memory. In other words diversification is an algorithm used to force the Tabu search to carry out the exploration in the unexplored areas. The long term memory used to store all visited solution from the 0th iteration. Those solution are exploited after a long period. The size of the memory and the period of the memory determines the validity of the diversification problem. If the size of the memory is too small then, then all solutions cannot be stored. If the period of memory is too small, it becomes intensification. Hence the size, and period is more important.

The intensification and the diversification can be achieved by using some alternative mechanism. The period of the tenure period indirectly assists the intensification and the diversification. Hence the Tabu tenure period fixation becomes critical. By properly selecting the value of tenure period avoid the usage of both long term and the short term memories can be avoided. If the long term and short term memory are to be avoided for intensification, and diversification then fixing the tenure period becomes a separate area.

1.4.7 Tabu Search Template

The general template for the Tabu search is given below with the notations and the search procedure. These search procedure operates over a
function $f(v)$. This function $f(v)$ may be a single variable or a multi variable function. The function also subject to some constraints. The variables also have some range. The solution has to be arrived at by considering all aspects.

**Notation**

- $S$ - The current solution
- $S^*$ - The best known solution
- $f^*$ - Value of $S^*$
- $N(s)$ - Neighborhood of $s.$
- $\tilde{N}(s)$ - Admissible subset of $N(s)$

**Initialization**

Set

- $s = s_0$
- $f^* = f(s_0)$
- $s^* = s_0$
- $T = \emptyset$

**Search**

While termination criteria not satisfied

- Select $s$ in $\arg\min [f(s')]$, $s' \subseteq \tilde{N}(s)$
- If $f(s) < f^*$ then set $f^* = f(s)$, $s^* = s$
- Record Tabu for the current move in $T$. 
1.4.8 Termination Criteria

The termination condition is vital. Different types of termination condition are considered for Tabu search. Each one has its own advantage and disadvantages. A termination condition has to be selected based on our need. The available termination conditions are

- After a fixed number of iterations
- After some number of iterations without an improvement in the objective function value
- When the objective reaches a pre-specified value.

1.5 PROBLEM DESCRIPTION

The main focus of this thesis is to develop an exploration tool for Genetic Algorithm. In order to develop a new exploration tool, the existing exploration tools were studied. They are having their own advantages and disadvantages. The single point and multipoint crossover are analyzed. Finally our new exploration tool namely “Odd and Even point Crossover” as proposed. In this proposed operator, the encoded string is divided into odd and even group. Apart from this, a probability measure called $P_{\text{odd}}$ is introduced. It used to control the probability at odd and even location.

The proposed crossover operator tested over the data fusion problem in Information Retrieval. This problem has been chosen because it has more combination and the search space is more complex. In order to find an optimal solution, the search space has to be explored completely. More ever the weights to the retrieval function can be increased to increases. The complexity of the overall problem. Two stages were selected the experiment is conducted with 12 bit and 16 bit encoding. The analysis tends to measure
the performance of the retrieval system. It produced a better result. After this confirmation, the convergence property is analyzed of the proposed crossover operator. The convergence is not smooth. It raised a doubt about the unexplored areas in the search space. Hence, the exploration is need to be controlled.

It is believed that, merging a best local search with our proposed crossover operator may control the exploration. For this purpose the Tabu Search is selected. The Tabu search merged with GA to give away a hybrid search tool (Kim and Abraham 2007). As the our proposed crossover operator is used, it is named as “Odd and Even Point Crossover based GA”.

The performance of the hybrid search tool is tested with the stand alone search tool. There is an improvement in the performance. The convergence analysis shows some improvement and it seems the exploration is controllable.

The controllability of the exploration tool has been assisted by the Tabu list and Tabu tenure period. Impact of Tabu tenure period over the convergence has been analyzed. It shows that there is a relationship prevailing between convergence and tenure period. The tenure period is varied to check the claim, and it has been proved. The variation in the tenure period influences the convergence of hybrid GA.

In our experiment, equal importance is maintained to all objects present in the model solution. Some objects (variables) may have more influence over others. The discrimination among the objects can be achieved by applying different tenure periods. The important objects are blessed with the higher tenure period. If a object has higher tenure period then it is exploited.
The tenure period assignment remains a challenge. So far the rule of thumb is followed for fixing the tenure period. For this purpose an experiment is conducted to test the impact of tenure period. Based on the result, a mechanism is developed for assigning the tenure period. The optimal tenure period produces the best result and the best convergence. The end results of this work seem promising.

### 1.5.1 Outcome of This Work

1. Odd and even point crossover operator.
2. Method to exploit odd and even group more vigorously based on the probability $P_{\text{odd}}$.
3. A new hybrid search/optimization tool odd and even point crossover based GA.
4. Method to assign optimal tenure period.

### Organization of the Thesis

This thesis is organized as follows. Chapter 2 gives the earlier work in the field of IR, Data fusion, GA, and Tabu. Chapter 3 gives the performance of odd and even point crossover based GA. Chapter 4 analyzes the convergence property of odd and even point crossover based GA. Chapter 5 gives the overall concepts of Tabu GA. Performance and convergence analysis of odd and even crossover based Tabu GA has been carried out in Chapter 6. Impact of tenure period over the system performance and convergence given in Chapter 7. The method to fix the optimal tenure period has been given in Chapter 8. The chapter 8 also gives the performance analysis based on optimal tenure period. The chapter 9 concludes with the future direction of our research.