CHAPTER 5

PERFORMANCE COMPARISON OF PROPOSED
ECBCA AND HHCGA ALGORITHMS

5.1 INTRODUCTION

Matching schema of data tables is one of the main challenges in information integration. The purely manual solution to the schema matching problem includes heavy labor intensive. Hence there is a need for automated techniques which will speed up this process by either automatically discovering good mappings, or by proposing likely matches that are then verified by some human expert. In this thesis, two proposed schema matching techniques such as an Efficient Centroid Based Clustering Algorithm (ECBCA) and Hybrid Hill Climbing with Genetic Algorithm (HHCGA) approach to improve the schema matching automation. In this chapter the performances of the proposed ECBCA and HHCGA algorithms are compared with the existing schema matching algorithms such as Kang Naughton Entropy only (KNE), Kang Naughton Mutual Information (KNMI), K-Means Clustering algorithm and the modified Kang Naughton Genetic Algorithm (KNGA). The forthcoming sections will compare the performances of the proposed algorithms using the performance metrics such as Precision, Recall, Computation Time, Sensitivity and Specificity for the Census and Medical lab real datasets.
5.2 EXPERIMENTS AND DATASETS

The proposed schema matching automation algorithms such as ECBCA and HHCGA are implemented in the JAVA platform. Experiments were performed on an Intel Core i5 @ 2.6 GHz machine with 8GB of memory, running Windows XP Professional.

Two real-world data namely Medical laboratory data and census data were used for the study. The medical laboratory data set contains the result of investigation patients. The table contains five years worth of patient’s records, which is 60K records, and each records consists of 36 test types attributes. A particular portion of the table is empty and the column data types is mostly numeric. Basic experimental technique with the medical lab data was to range partition the original table based on exam dates (column 1) into two sub-tables and to use these two sub-tables. Assume these two sub tables as two different tables for which the schemas need to be matched. Obviously, the correct answer for the mapping is well known; but the mapping algorithm did not give the same.

The second data set taken is the census data. There are 160 attributes in the table. The experiments were done on a randomly chosen set of 24 attributes. The original census data files have some duplicate columns and two of them are in the 30 attributes randomly chosen for our experiments. Therefore the accuracy of matching was low because of duplicate columns.

The experiments were done by increasing the number of attributes in the input tables to be matched. For each table width, two to 24, the measurement was iterated 30 times with randomly chosen subsets of attributes. Then the results are averaged.
In the proposed HHCGA algorithm, the performance of a GA heavily depends on the choice of its main control parameters population size, mutation rate and crossover rate. Genetic Algorithm in the proposed HHCGA algorithm uses the Mutation rate of 0.01, Cross over rate of 0.8 and Population size of 5000.

The mutation rate which prevents the population from stagnating at local optima is selected based on the experiments carried out by the various mutation values varying from 1% to 10%. Our results showed optimal performance and convergence improvement with a mutation rate 0.01.

Experiment also carried out for various population sizes ranging from 1000 to 5000 with the maximum of 30,000 generations. There is a gradual increase in the fitness function with increase of population size up to 5000. Further increase in population size does not show any significant improvement in the fitness function. Hence the population size is fixed at 5000.

Similarly the crossover rate was chosen. Our GA system got stagnated when we tried with low Crossover rate. However, if the Crossover rate is too high, unstable solutions may be quickly substituted into the population for individuals with better fitness. From extensive experiments, we found the proposed GA algorithm work well with crossover rate of 0.8.

The ordered Crossover type and the Rank selection procedures are used in the proposed work.
5.3 PERFORMANCE METRICS

The metrics that are evaluated for the proposed ECBCA and HHCGA schema matching algorithms are Precision and Recall, Computation time, Sensitivity, and Specificity. These terms are defined in the following.

- **Precision**

  For the schema matching process, the Precision is the fraction of derived matches of schema attributes that are relevant to the schema of table instances. Precision is given in Equation 5.1.

  \[
  P = \frac{|(relevant \ match) \cap (derived \ match)|}{|(relevant \ match)|} \quad (5.1)
  \]

- **Recall**

  For the schema matching process, the Recall is the fraction of relevant matches that are derived to the schema of table instances. It is given in Equation 5.2.

  \[
  R = \frac{|(relevant \ match) \cap (derived \ match)|}{|(derived \ match)|} \quad (5.2)
  \]

- **Computation Time**

  This is time taken for executing the schema matching algorithms.

- **Sensitivity and Specificity**

  Sensitivity and Specificity are given in Equations 5.3 and 5.4 respectively.
where,

true match = a matching attribute pair identified as a schema match
true non-match = two attributes that do not match identified as not a schema match
false schema match = two attributes that do not match identified as a schema match
false non-match = a matching attribute pair identified as not a schema match

**F-measure** is where precision P and recall R are equally weighted.

\[
F\text{-measure} = \frac{2\times P \times R}{P + R}
\]  

(5.5)

A good clustering solution has the F-measure value closer to one.

Based on these evaluation metrics, the performances of the proposed ECBCA and HHCGA based schema matching automation algorithms are evaluated and compared along with the existing algorithms such as KNE, KNMI, K-Means and modified KNGA.

5.4 PERFORMANCE COMPARISON

In this section, performances of the proposed ECBCA and HHCGA algorithms for schema matching technique are compared with the existing schema matching techniques in respect of the performance metrics such as Precision, Recall, Computation Time, Sensitivity and Specificity.
5.4.1 Precision Performance Comparison

The Precision performance of the proposed ECBCA and the HHCGA schema matching algorithms by varying the number of attributes from 2 to 24 for both the Census data set and the medical lab data set are compared and presented in the Figure 5.1. From the Figure 5.1(a), it is inferred that the proposed ECBCA and HHCGA algorithms outperform the other existing schema matching automation algorithms. The graph shows that for the Census dataset, the proposed ECBCA and HHCGA improve the precision by 54.02 % and 57.86 % over the existing KNMI algorithm. Similarly, when compared to the existing KNGA and K-Means algorithms, the proposed HHCGA improves precision by 11.93 % and 17.41 % respectively. In the case of proposed ECBCA, 14.56 % improvement in precision over the existing K-Means algorithm for the census data. In general if the number of attributes increases then the precision value will be decreased. This is due to the ratio of the number of correct matches in the produced match results to the number of matches produced by a schema matching algorithm will be decreased. As the proposed ECBCA enhances centroid section in forming the clusters, the global optimal schema matching is considerably increased when compared to the existing other algorithms. Similarly, in the proposed HHCGA, the improvement is due to improvising the searching automation technique by combining Genetic Algorithm by hill climbing approach which will find the optimal matches of the schemas. Among the proposed ECBCA and HHCGA algorithms, HHCGA outperforms precision by 2.49 % over the ECBCA. Similar performance is received when using the proposed ECBCA and HHCGA algorithms for the Medical lab data set which is experienced from the Figure 5.5 (b).
Figure 5.1 (a) Overall Precision comparison for the Census dataset

Figure 5.1 (b) Overall Precision Comparison for the Medical Lab dataset
5.4.2 Recall Performance Comparison

Figure 5.2 illustrates the performance of the proposed schema matching algorithms for both the Census and Medical lab dataset. From the graphs, it is inferred that when the number of attributes increase, the recall also increases. This is due to the ratio of the number of correct matches in the produced match results to the total number of true matches in two input schemas will be increased. In Figure 5.2 (a), the recall value of the proposed ECBCA and HHCGA algorithms are compared with the existing KNE, KNMI, K-Means and modified KNGA algorithms by increasing the number of attributes from 2 to 24. Among the various algorithms compared, the proposed HHCGA outperforms the other existing and proposed algorithms. The improvement is due to the hybridisation of HC with GA. For the Census data, the recall performance improvement is about 51.26 %, 29.85 %, 20.27 % and 9.06 % over the existing KNE, KNMI K-Means and the modified KNGA algorithms and 3.25 % over the proposed ECBCA.

![Recall performance comparison](image)

Figure 5.2 (a) Overall Recall Comparison for the Census dataset
For the Medical Lab dataset shown in Figure 5.2 (b), the recall performance improvement of the proposed HHCGA is about 45.42 %, 28.44 % and 9.05% over the existing KNE, KNMI, K-Means and modified KNGA algorithms when using 10 K sample data. Similarly, HHCGA shows improved recall performance of 4.5 % over the other proposed ECBCA algorithm for schema matching.

![Recall performance comparison](image)

**Figure 5.2 (b) Overall Recall Comparison for the Medical Lab dataset**

### 5.4.3 Computation Time Performance Comparison

The graph presents in Figure 5.3 compares the performance in respect of computation time for the proposed ECBCA and HHCGA algorithms with the other existing algorithms such as KNE, KNMI, K-Means and modified KNGA algorithms when varying the number of attributes from 2 to 24 for both the Census and Medical lab datasets. From the figure, it is inferred that when the number of attributes increases, the computation time also increases because of the increase in search space. Though the computation time increases, among the various algorithms compared, the proposed HHCGA and ECBCA outperform because of their efficient
integration of hill climbing with genetic algorithm and formation of clusters. Among the four algorithms compared, the proposed HHCGA algorithm shows improvement by 26.87 %, 69.47 % and 19.26 % over the existing KNE, KNMI and modified KNGA algorithms. In the proposed ECBCA, the achieved improvement in computation time for the Census dataset is about 18.91 %, 66.15 % and 32.65 % over the existing KNE, KNMI and K-Means algorithms. The results for the Census data set are plotted in the graph shown in Figure 5.3 (a). Among the proposed HHCGA and ECBCA, HHCGA improves computation time by 9.81 % over the ECBCA.

![Computation Time performance comparison](image)

**Figure 5.3(a) Overall Computation Time comparison for the Census dataset**

For the Medical lab data set, the performances of the proposed ECBCA and HHCGA are compared and presented in Figure 5.3 (b). The proposed HHCGA outperform the other existing algorithms such as KNE, KNMI, K-Means, modified KNGA and the proposed ECBCA by 20.77 %, 66.8 %, 16.26 %, 10.71 % and 4.21 % respectively. This improvement is due to the effective integration of GA with HC and GA minimises the the number of hill climbing iterations considerably. Among the existing KNE and KNMI,
KNE shows improved performance in computation time metric when compared to KNMI because of its simple procedure. Though the KNE shows improved computation time, it could not improved the Precision, recall, error rate etc.

![Computation Time performance comparison](image)

**Figure 5.3(b) Overall Computation Time comparison for the Medical lab dataset**

Average computation time performance of the proposed and existing algorithms are presented in the graph shown in Figure 5.4 (a) and 5.4 (b) for both the census and medical lab dataset. From the figure, it is inferred that the proposed HHCGA and ECBCA outperform the other existing algorithms. Average computation time for the proposed ECBCA and HHCGA are 7886 ms and 7113 ms; but the existing algorithms such as KNE, KNMI, K-Means and modified KNGA show about 9725 ms, 23298 ms 11709 ms and 8810 ms respectively for the census dataset. Similarly for the medical lab dataset, the proposed HHCGA, ECBCA and existing KNE, KNMI, K-Means and modified KNGA show about 8423 ms, 8793 ms, 10631 ms, 25372 ms, 10059 ms and 9433 ms respectively.
5.4.4 Sensitivity Performance Comparison

Figure 5.5 compares the sensitivity performances of the proposed HHCGA and ECBCA algorithms with respect to varying the number of
attributes from 2 to 24 for both the Census and medical lab data sample set of 10 K tuples. Among the various algorithms compared, the proposed HHCHA outperforms the others, since it has high sensitivity, i.e., high precision of finding attribute matches. As the proposed HHCGA algorithm uses both hill climbing to find the local matches and GA for finding the global matches using the local matches, it considerably improves the sensitivity performance. From the experimental results, the HHCGA improves sensitivity by 13.77 %, 9.11 %, 7.33 %, 2.81 % and 1 % over the existing KNE, KNMI, K-Means, modified KNGA algorithms and the proposed ECBCA algorithm for Census data set which is shown in Figure 5.5 (a).

![Sensitivity performance comparison](image)

**Figure 5.5 (a) Overall Sensitivity comparison for the Census dataset**

Figure 5.5 (b) presents the experimental results of the proposed schema matching algorithms with respect sensitivity performance metric for the medical lab dataset. The results plotted in the graph ensures that the proposed HHCGA and ECBCA outperform the other existing schema matching algorithms. The improvement of the proposed HHCGA over the existing
KNE, KNMI, K-Means, modified KNGA and proposed ECBCA by 18.13 %, 10.51 %, 8.71 %, 5.70 % and 3.25 % respectively.

Figure 5.5 (b) Overall Sensitivity comparison for the Medical lab dataset

5.4.5 Specificity Performance Comparison

Figure 5.6 compares the performances of the proposed HHCGA and ECBCA algorithms with respect to specificity performance metric when varying the number of attributes from 2 to 24. For the experiment, 10 K tuples data sample is taken and the experiment is carried out multiple times and the average is used to represent a single attribute. When increasing the number of attributes, the specificity is also increasing because of increase in search space. For the Census dataset, the proposed HHCGA and ECBCA shows about 47.81 % and 39.80 % improvement over the existing KNMI algorithm and for the Medical lab dataset it is about 48.81 % and 41.46 %. Similarly the proposed algorithms show improvement in specificity over the existing K-Means, KNGA and KNE algorithms also for both the two real datasets used. For both the datasets, the ECBCA and HHCGA achieve the doubles the improvement in specificity. This is illustrated through the Figures 5.6 (a) and
5.6 (b). The proposed HHCGA shows high specificity, i.e., high precision of rejecting non-matching attributes. Optimising the matching schemas and nodes using hill climbing algorithm with GA, the proposed HHCGA will significantly improve the specificity compared to others.

Figure 5.6 (a) Overall Specificity comparison for the Census dataset

Figure 5.6 (b) Overall Specificity comparison for the Medical lab dataset
5.4.6 Computation Time Performance Comparison for Various Sample Size

Figure 5.7 compares the computation time performance metric of the proposed ECBCA and HHCGA algorithms for the various sample sizes. For example, if the experiment is carried out for the 1K tuples of sample data then the proposed ECBCA is represented as ECBCA-1K and the proposed HHCGA is represented as HHCGA-1K. In Figure 5.7 (a), the performance of the proposed HHCGA and ECBCA algorithms in respect of computation time for the various sample size from 1K to 30K by varying the number of attributes from 2 to 24. When the sample size is increased and the number of attributes are also increased, the computation time also increased. This is due to the increase in search space. Among the various samples based ECBCA and HHCGA performances are compared, HHCGA and ECBCA for 1K, 5K and 10K tuples show almost similar performance. Hence in the thesis, almost in all the experiments, 10K tuples sample data is used. Similar performance is experienced for medical lab dataset also and it is given in Figure 5.7 (b).

Figure 5.7(a) Overall Computation Time comparison using various sample data size for the Census dataset
Figure 5.7(b) Overall Computation Time comparison using various sample data size for the Medical lab dataset

5.4.7 Precision Performance Comparison for Various Sample Size

Figure 5.8 compares the precision rate of the proposed ECBCA and HHCGA algorithms with various sample size such as 5 K, 10 K and 15 K tuples by varying the number of attributes from 2 to 24. When the number of sample size is increased, the average precision rate also increased because of the more sample data available for matching. In Figure 5.8 (a), precision rate of the proposed ECBCA and HHCGA algorithms with various sample size such as 5 K, 10 K and 15 K tuples by varying the number of attributes from 2 to 24 is compared for the census data set and from the results it is inferred that the proposed ECBCA and HHCGA with 5 K and 10 K tuples show the similar performance when compared to the 15 K tuple data size. Similar performance is experienced for the medical lab dataset also. The experimental results are presented in the graph shown in Figure 5.8 (b).
Figure 5.8(a) Overall Precision comparison using various sample data size for the Census dataset

Figure 5.8(b) Overall Precision comparison using various sample data size for the Medical lab dataset
Figure 5.9 compares the average precision rate of the proposed ECBCA and HHCGA algorithms with 5 K, 10 K and 15 K tuples. Among the various data size compared, for 5 K and 10 K tuples the proposed ECBCA algorithm shows 89% and 90% precision and the proposed HHCGA shows 93.87% and 93.93% precision. Hence in the thesis, with the consideration of computation time, 10 K tuples sample data size is used for the performance evaluation. Figure 5.9 (a) and 5.9 (b) compares the average precision rate of the proposed ECBCA and HHCGA algorithms for census and medical lab datasets.

![Figure 5.9(a) Average Precision comparison of the proposed HHCGA and ECBCA using various sample data size for the Census dataset](image-url)
5.4.8 F-measure Comparison

Figure 5.10 (a) compares the F-measure performance metric for the Census dataset. Among the various schema matching algorithms compared, the proposed ECBCA and HHCGA performs better F-measure value due to their higher precision and recall values. When compared to the existing K-Means clustering algorithm, the proposed ECBCA shows 16.61 % improvement in F-measure value and HHCGA improves F-measure by 46.86 % over the existing KNMI algorithm. Among the proposed, the HHCGA shows improved performance over the ECBCA. Similar F-measure performance is achieved for the medical lab dataset also and it is presented in Figure 5.10 (b).
5.5 SUMMARY

The overall performance comparison of the proposed ECBCA and HHCGA schema matching automation algorithms were compared in respect of performance metrics such as Precision, Recall, Computation Time, Sensitivity and Specificity. Also, in this chapter, the performances of the proposed algorithms are compared with the existing KNE, KNMI, K-Means
and modified KNGA algorithms for schema matching. Among the various algorithms compared, experimental results ensured that the proposed ECBCA and HHCGA algorithms outperformed the others. Also among the proposed HHCGA and ECBCA, HHCGA shows improved performance when compared ECBCA, due to the efficient integration of hill climbing with Genetic algorithm.

The conclusion and future enhancement of the proposed research work towards the schema matching automation will be discussed in the forthcoming chapter of the thesis.