Journal of Computer Science 8 (9): 1487-1495, 2012
ISSN 1549-3636

Optimization Techniques To Record Deduplication

[1]Deepa Karunakaran and [2]Rangarajan Rangaswamy
[1]Department of Information Technology,
Sri Ramakrishna Engineering College, Anna University-Coimbatore, India
[2]Indus College of Engineering, Coimbatore, India

Abstract: Duplicate record detection is important for data preprocessing and cleaning. Artificial Bee Colony (ABC) is one of the most recently introduced algorithms based on the intelligent foraging behavior of a honey bee swarm. Our approach to duplicate detection is the use of ABC algorithm for generating the optimal similarity measure to decide whether the data is duplicate or not. In the training phase, ABC algorithm is used to generate the optimal similarity measure. Once the optimal similarity measure obtained, the deduplication of remaining datasets is done with the help of optimal similarity measure generated from the ABC algorithm. We have used Restaurant and Cora datasets to analyze the proposed algorithm and the performance of the proposed algorithm is compared against the genetic programming technique with the help of evaluation metrics.

Key words: Data preprocessing, genetic programming, remaining datasets, similarity measure obtained, evaluation metrics, proposed algorithm, Artificial Bee Colony (ABC)

INTRODUCTION

Normally, organizations become conscious of practical precise disparities or inconsistencies while integrating data from diverse sources to implement a data warehouse. Such problems belong to the category called data heterogeneity (Ahmed *et al*., 2007).

With the increase in size of the database the problem intensifies taking into account the huge amount of computational resource required for examination and removal of duplicate records (Haidarian *et al*., 2006). Duplicates can occur out of numerous scenarios, for instance when a large database is updated by an external source and registry numbers are not accessible or are in error (Winkler, 2001).

File systems often contain superfluous copies of information: identical files or sub-file regions, perhaps stored on a single host, on a shared storage cluster, or backed-up to secondary storage. Deduplicating storage systems take advantage of this redundancy to decrease the essential space needed to contain the file systems (or backup images thereof). Deduplication can work at either the sub-file (Dutch and Bolosky, 2011; Dubnicki *et al*., 2009; Ungureanu *et al*., 2010) or whole-file (Bolosky *et al*., 2000) level.

Data deduplication policies can be classified according to the basic data units they handle. In this context, manily two main data deduplication strategies can be defined: File-level deduplication, in which only a single copy of each file is stored. Two or more files are known as identical if they have the same hash value. This is a very popular service imbibed in multiple products (Harnik *et al*., 2010; Gunawi *et al*., 2005; Douceur *et al*., 2002); Block-level deduplication, which breaks files into blocks and stores only a single copy of each block. The system could either use fixed-sized blocks (Quinlan and Dorward, 2002) or variable-sized chunks (Muthitacharoen *et al*., 2001; Vrable *et al*., 2009). The architecture of the deduplication solution is modeled by two basic approaches. In the target-based approach deduplication is managed by the target data-storage perhipherals or service, while the client is ignorant of any deduplication that might occur. Source based deduplication is performed on the data at the client side only before it is transferred. Particularly, there is communication between client software and the backup server to check for the presence of files or blocks (Harnik *et al*., 2010). Two well-known source de-duplication methods, source local chunk-level de-duplication (Tan *et al*., 2010) and source global chunk-level de-duplication have been proposed in the past to address the above mentioned problem by erasing the redundant data chunks before transfering them to the remote backup destination.

The studies (Zhu *et al*., 2008; Rhea *et al*., 2008; Lillibridge *et al*., 2009; Bhagwat *et al*., 2009) expose that, due to the out-of-memory fingerprint accesses to massive backed-up data, chunk-level de-duplication has an inherent latency and suffers throughput problem that affects the backup performance. In source global

Corresponding Author: Deepa Karunakaran, Department of Information Technology, Sri Ramakrishna Engineering College, Anna University-Coimbatore, India

chunk-level de-duplication, this overhead of massive disk accesses will regulates the deduplication process and which will result in increase of backup window. While in source local chunk-level de-duplication, the overhead is reduced by searching the duplicate chunks at the same client. This alleviated overhead, however, limits the compression ratio, which results in increases of backup window due to the increased data transmission cost. Therefore, there is immediate need to achieve a balance between de-duplication efficiency and deduplication overhead for the maintainance of a shorter backup window than existing solutions. There are several other methods that have been proposed for the deduplication purpose which are having efficiency and accuracy. The methods are deduplication using genetic algorithm, semantic methods, cloud services. Above mentioned problems have been solved by the deduplication methods which have been modeled using GA. This research has been done to find the optimization techniques that are having some performance superiority over these existing methods.

The recent researches have given many methods for the deduplication purposes with many distinct features by their own. In this study, we tried to propose a better method for the deduplication approach. The techniques we proposed is, the ABC algorithm can provide better performance and accuracy than the genetic algorithm based techniques, which is presented in the recent times (Moises *et al*., 2011). The proposed algorithm is used the restaurant and cora data to evaluate their performance against the genetic programming based technique. The results from the evaluation of the proposed approach are satisfactory as compared to the results provided by the genetic algorithm for the same set of input data.

The distinct features we considered here fitness function of the ABC algorithm. ABC algorithm is based on the food processing of the bee colony and has three phases namely, employed bee phase, onlooker bee phase and scout bee phase as its characteristics.

Review of related works: Several researches have been done in field of deduplication. Recently, deduplication in distributed manner has fascinated lots of researchers due to the demand of scalability and efficiency. Here, we reviewed the recently done works in the literature for deduplication and the different approaches used for it. Moises *et al*. (2011) have proposed a genetic programming based approach to record deduplication that prepares a deduplication function to identify the replicate pairs on the basis of evidence extracted from the data content. They have also shown experimentally that their approach better than existing state of-the-art method which has been proposed in literature. Moreover, the devised functions take less time computationally because they used less evidence. In addition, the genetic programming approach was capable of automatically limiting these functions to a given fixed replica identification boundary, which frees the user from the load of choice and tunning this parameter.

Karaboga and Ozturk (2010) used Artificial Bee Colony algorithm to fuzzy clustering of medical data which are widely used benchmark problems. The results of ABC algorithm are compared with Fuzzy C-Means (FCM) algorithm and the experiments showed that the Artificial Bee Colony algorithm is very successful on optimization of fuzzy clustering.

Ektefa *et al*. (2011) have proposed another method which was based on a threshold whose criteria was to takes string and semantic similarity measures for comparing record pairs into consideration. This method was experimented on a real world dataset of Restaurant and several standard evaluation metrics have been used to judge it. As experimental results indicate, method which was based on the combination of string and semantic similarity measures were efficient than the individual similarity measures in Restaurant dataset. Therefore, based on experimental results, string similarity, semantic similarity should be considered in order to detect duplicate records more effectively.

Kumbhar and Krishnan (2011) have presented an ABC based methodology, which maximizes its accuracy and minimizes the number of connections of an ANN by evolving at the same time the synaptic weights, the ANN's architecture and the transfer functions of each neuron. The methodology is tested with several pattern recognition.

Elhadi and Al-Tobi (2009) have proposed method that reports on experiments performed to investigate the use of a combined Part of Speech (POS) and an improved Longest Common Subsequence (LCS) in the analysis and calculation of similarity between texts. The text's syntactical structures are the main elements and were used for representation ofr documents. A better LCS algorithm was applied to representation to compare and rank the documents according to the similarity of their representative string. The approach was applied in detecting duplicate documents within a corpus and in the filtering of search engine results. Results obtained were encouraging. Qingwei *et al.* (2010) have proposed an algorithm using PSO algorithm to search the optimized partial contents. For PSO algorithm, it gives the encoded particles. For string similarity a new related coefficient of strings was defined for strings similarity. An evaluation function of PSO was devised on the basis of related coefficient function. And the searching of partial contents is done by using the hybrid mutation PSO algorithm. And the effectiveness of

algorithm is proved by making a simulation experiments which can search the similar partial contents in two documents successfully.

Samanta and Chakraborty (2011) proposed algorithm using, artificial bee colony to search out the optimal combinations of different operating parameters for three widely used Non-Traditional Machining (NTM) processes i.e., electrochemical machining, electrochemical discharge machining and electrochemical micromachining processes. Both the single and multi-objective optimization problems for the considered NTM processes are solved using this algorithm.

Kumar and Govindarajulu (2009) have conducted a survey on Duplicated web pages that are having identical structure but different data. These type of pages can be regarded as clones. To identify similar or near-duplicate pairs in a large collection is a challangeing problem with wide-spread applications. The problem has been deliberated for diverse data types (e.g., textual documents, spatial points and relational records) in diverse settings. Another contemporary materialization of the problem was the efficient identification of near-duplicate Web pages. This was undoubtedly a demanding in the web-scale due to the voluminous data and high dimensionalities of the documents. This main intention behind this survey paper is to prepare an up-to-date review of the existing literature in duplicate and near duplicate detection of general documents and web documents in web crawling.

Motivating algorithm: The main problems caused by duplicates in the data repository is inefficient memory usage, high execution time. So as a remedy for this problem, we are using the algorithms for finding and separating the near replicas in a data repository. Moises *et al.* (2011) had proposed an approach for record deduplication by applying the genetic programming. The GA approach to record deduplication is to combine l different pieces of evidence extracted from the data content And to devise a deduplication function that will be able to identify whether two entries in a data store are replicas or not. In reference to the above mentioned algorithm, we have proposed an deduplication approach based on optimization algorithm like Artificial Bee Colony (ABC) algorithm.

MATERIALS AND METHODS

Proposed methodology: The proposed approach has two phases such as training phase and duplicate detection phase. These two phases are explained with the four different steps.

Step 1: Similarity computation for all pair of records: In this step, the similarity computation is carried out by finding the similarity functions on each record field. Each function compares the similarity of each field with other record fields and assigns a similarity value for each field. Accurate similarity functions are very important to calculate the distance between the records for better duplicate detection. Levenshtein distance and cosine similarity are the two similarity measures used in our proposed approach. Here, the input records are partitioned into two parts and the two measures are computed for the two parts of record pairs. This operation provides the four similarity values (a, b, c, d) for the record pair. (1) *Levenshtein distance:* The chosen name fields of the records are "record 1" and "Record 2". The "Levenshtein distance" is computed by calculating the minimum number of operations that has to be made to transform one string to the other, usually these operations are: replace, insert or deletion of a character. The levenshtein distances between the records are found out by considering the record as a whole. *2) Cosine similarity:* The cosine similarity between the two records name field "Record 1" and "Record 2" are calculated as follows: First, the dimension of both strings are obtained by taking the union of two string elements in the "record 1" and "record 2" as (word1 , word2, …….word N) and then the frequency of occurrence vectors of the two elements are calculated i.e., "record 1" = (<vector value1>, <vector value2>,……<>) and "record 2"= (<vector value1>, <vector value2>,……<>) . After that, we obtain the dot Product and magnitude of both strings.

Step 2: Computing feature vectors: Feature vectors represent the set of elements that is required for the detection of duplicate elements from the data repository. The vectors can be obtained from the processing of the two similarity measure values. In general, the usual similarity functions may fail to find the similarity correctly, because the computation of similarity between fields can vary significantly depending on the domain and specific field under consideration. Therefore, it is necessary to adapt similarity measures for each field of the database with respect to the particular data domain for attaining accurate similarity computations. Consequently, we combine these similarity values obtained from different similarity measures to compute the distance between any two records. Here, we can represent similarity between any pair of records by a feature vector in which each component has the similarity value between two records of anyone of the similarity measure. When considering a database D that contains records

composed of n different fields and a set of m distance metrics, we can represent similarity between any pair of records by a 4-length vector. Each component of the vector represents the computed similarity value between two records that is calculated using one of the m distance metrics.

Step 3: New similarity formulae generation using optimization algorithm: In this step, we consider the optimization algorithm for the extraction of the feature vectors. An expression derived to calculate the fitness of the corresponding data. In order to find more precise output, i.e., to find the near duplicates better, we process a number of expressions. These expression, that we subject to process are used for the calculation of duplicates. A set of similar expression are supplied as input to the optimization algorithms for the find better among the supplied inputs. The optimization algorithms find the best among the input expressions, which is capable of providing better solution for the problem.

Step 4: Duplicate detection using the new similarity formulae: Once the optimal similarity formulae are generated from the optimization algorithms, the generated formulae is used to find the duplicate or non-duplicate records. Here, we fix the threshold, T to find the margin between duplicate and non-duplicate pairs.

Algorithm:

Artificial bee colony based deduplication: The ABC algorithm is one of the newly introduced optimization algorithm, the algorithm is introduced in 2005 by Karaboga and Ozturk (2010). The ABC algorithm is characterized by optimizing a number of solutions according to the foraging feature of the bees. The typical mathematical methods used in the ABC algorithm give extra hand for the ABC to differ from other optimization algorithms. The main features of ABC algorithm are the Employed bees, Onlooker bees and scout bees, which are processing elements for the optimization process. The ABC algorithms is processed in terms of cycles, in each cycles new employed bees, onlooker bees and scout bees are generated. The proposed approach, the input initially is considered as the employed bees. The processing of the bees or the input is done in three phases, they are.

Employee-bee phase: In the proposed approach, the expressions that are used to determine the duplicates are used as the input. There expressions are initially considered as the employed bees:

Employed bee

$(a+b)^2+(c-d)^{-2}$

$((a+b)*(c-d))^{-2}$

$(a^2+b^2)*(c^2-d^{-2})$

$c(a+b)-d(a-b)$

The above shown is an example of the set of data, which are given as input. The whole data is considered as an employed bee. Like PSO algorithm, initially we find the fitness of the employed bees. The bees with best fitness value are stayed with the population and rests are rejected. The main objective of the employed bee phase is to generate the best solution.

Fitness function: In the proposed approach, we find the fitness values for the expressions generated for determining the duplicates. In the current scenario, we are selecting the expression, which determine duplicates, for evaluating the fitness. The fitness function that we used in the proposed approach is composed of three factors. These factors are the same factors which are used in the PSO algorithm. Here the fitness function is different, i.e. we are using the fitness function defined by the ABC algorithm itself:

$$\text{fitness} = \begin{cases} \frac{1}{1+f_i}, f_i > 0 \\ 1+\text{abs}(f_i), f < 0 \end{cases}$$

The values *recall*, *precision* and *fmeasure* are used for the calculation of f value in order to calculate the fitness-s value. In the current scenario, the f values are generated by the fmeasure value of each expression in the employed bees. The next phase of the algorithm will proceeds according to the fitness value obtained from the calculations. In this ABC algorithm, the expression which possesses the best fitness is stayed and rests are rejected. The replacing of the expression will be done in the onlooker phase.

Onlooker-bee phase: This phases is the replacement of new population generation phase of the PSO algorithm. In the ABC algorithm, we select the employed bee and process it to generate a new set of bees. This new phase generate a set of new bee with different position value. In the onlooker bee phase, we select one expression from the employed bee and new solution for that bee is calculated using the following formulae:

$$v_i = v^0 + \phi(v^0 - v_k)$$

The object *v* represents the new solution for the existing solution v^0. The value for φ is a random number ranges in [0, 1]. The value of k is also randomly generated. The f_i value for the particular solution is calculated and then the fitness. If the calculated fitness value of the new solution is better than that of the old solution, then the new solution replaces the old. This process continues up to the last cycle. The new solutions are called improved solutions, according to the ABC algorithm, if there is no improved solution in a particular cycle that solution is considered as abandon solution.

Scout bee phase: The problem with the abandon solution is solved with the scout bee phase. When an abandon solution is discovered, then that solution is replaced with a randomly generated solution. The newly introduced solution is called scout bee. The scout bee is then becomes anemployed bee and the process continues as described in the prior sections. A scout bee is introduced at the end of each cycle, if there exists an abandon phase.

Termination phase: A termination criterion of the ABC algorithm is also fixed by the user itself. The termination criteria set in accordance with nature result that has to be produced by the ABC algorithm. Usually the number of cycles, to which the program has to execute, is set as the termination criteria. Once the criteria are met, the program stops execution and produce the result as per the ABC algorithm.

Example.1 Evaluation function = (a-b)/(c+d) Consider the employed bee as:

$$\begin{matrix} a & b & - \\ c & d & + \end{matrix}$$

The onlooker can be derived as:

$$(a-b)/(c+d) \rightarrow '+ \rightarrow '(a+b)/(c+d)$$

So new onlooker bee → (a+b)/(c+d)

RESULTS AND DISCUSSION

The method we proposed includes optimization based algorithms such as GA and ABC. The performance of the proposed approach is evaluated under different evaluation criteria. All algorithms are implemented in MATLAB and executed on a core i5 processor, 2.1MHZ, 4 GB RAM computer.

Dataset description: In the experiment we have selected datasets from the Riddle data repository Riddle dataset and the datasets used is Restaurant dataset. The datasets, which we are used in our proposed approach, is detailed below.

Dataset1 (Restaurant): This dataset Riddle dataset contains four files of 500 records (400 originals and 100 duplicates), with a maximum of five duplicates based on one original record (using a Poisson distribution of duplicate records) and with a maximum of two modifications in a single attribute and in the full record.

Dataset2 (CORA): The Cora dataset Riddle dataset consists of duplicate and non-duplicates data records and the Cora data includes 13 attributes.

Experimental results: The experimentation starts from selecting the datasets as the input of the similarity computation by the similarity computation factors, listed in the above sections, such as Levenshtein distance method and cosine similarity method. The similarity factors produce feature vectors on regard with the elements in the dataset. The feature vectors produced are represented with variables <a, b, c, d>. The expressions are created from the feature vectors produced by the similarity vectors. The populations are the starting point of the two optimization algorithm. The next step is the fitness evaluation:

$$\text{Example}: \; < a, b, c, d > \text{®} (a+b) + (c+d)$$

The above shown is an element in the population of the optimization algorithms. The processing on this population defines the relevance of the population. The fitness function defines the importance of the elements in a particular population. The fitness function for ABC algorithm is defined based on the different bee phases. The dataset is processed with the algorithms for a number of iterations and the best results of each algorithm for the particular iteration are listed in Table 1.

Table 2, the solution or expressions are arranged based on the best fitness value of a particular expression. In the table we can see that, the proposed algorithm has the upper hand over the existing genetic algorithm.

Table 1: Best fit solutions

Iterations	Algorithm	Fitness values	Expression
1	ABC	0.812	'(a+b)+(c+d)'
	GA	0.790	'(a-b)+(c+d)'
10	ABC	0.842	'(a+b)+(c+d)'
	GA	0.792	'(a-b)+(c+d)'
50	ABC	0.842	'(a+b)+(c+d)'
	GA	0.792	'(a-b)+(c+d)'
100	ABC	0.842	'(a+b)+(c+d)'
	GA	0.792	'(a-b)+(c+d)'

Table 2: Top obtained solutions sorted based on the fitness value

GA	ABC
'(a+b)+(c+d)'	'(b+a)+(d *c)'
'(a+b)-(c+d)'	'(b+a)-(d *c)'
'(a-b)+(c+d)'	'(b*a)+(d *c)'
'(a+b)+(c-d)'	'(b+a)+(d *c)'
'(a+b)-(c-d)'	'(b-a)-(d-c)'
'(a-b)-(c+d)'	'(b*a)-(d-c)'
'(a-b)+(c+d)'	'(b-a)-(d*c)'
'(a-b)*(c+d)'	'(b+a)-(d *c)'
'(a*b)+(c+d)'	'(b*a)-(d *c)'
'(a-b)+(c*d)'	'(b+a)-(d *c)'
'(a*b)-(c+d)'	'(b+a)-(d /c)'
'(a*b)-(c+d)'	'(b/a)-(d *c)'
'(a*b)+(c*d)'	'(b+a)-(d /c)'
'(a-b)+(c+d)'	'(b-a)*(d-c)'
'(a*b)+(c*d)'	'(b*a)*(d-c)'

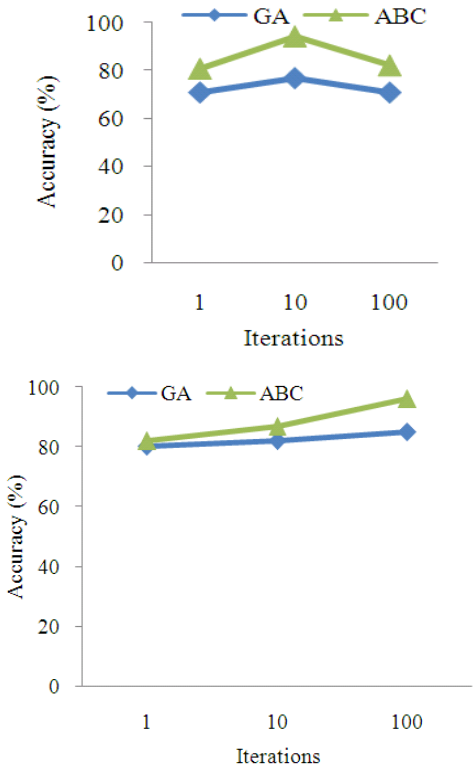

Fig. 1: Accuracy based on Threshold T1

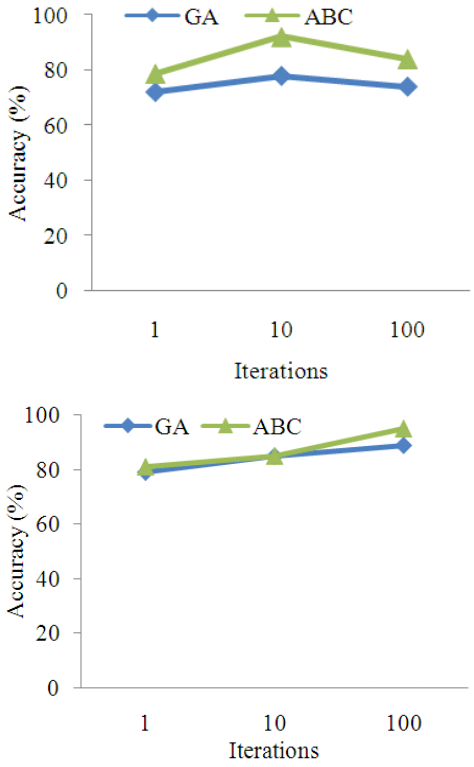

Fig. 2: Accuracy based on Threshold T2

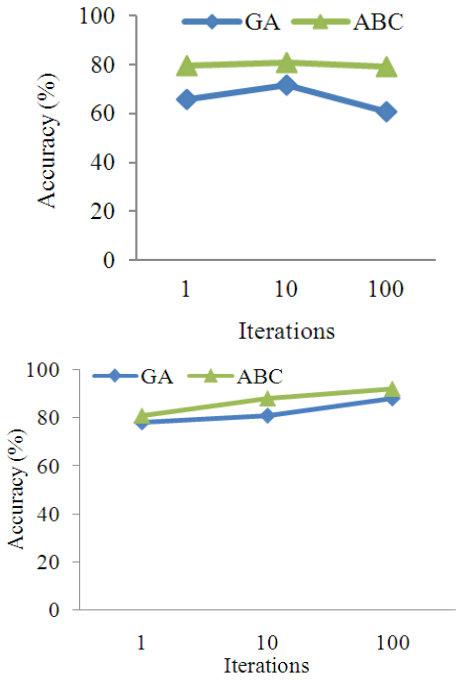

Fig. 3: Accuracy based on Threshold T3

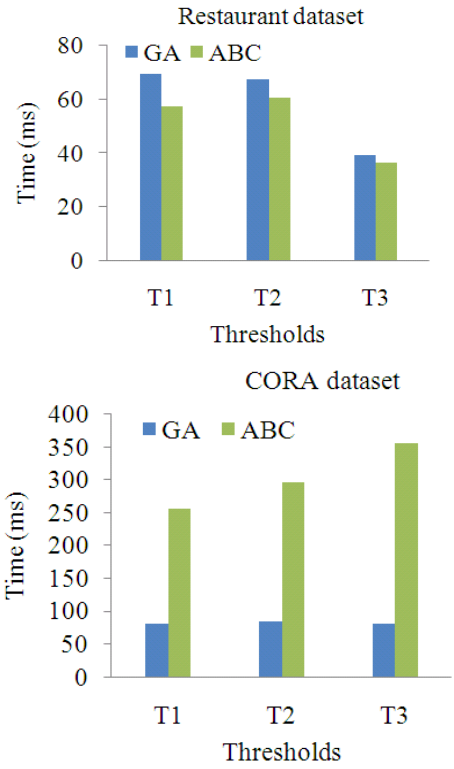

Fig. 4: Time of deduplication at single iteration

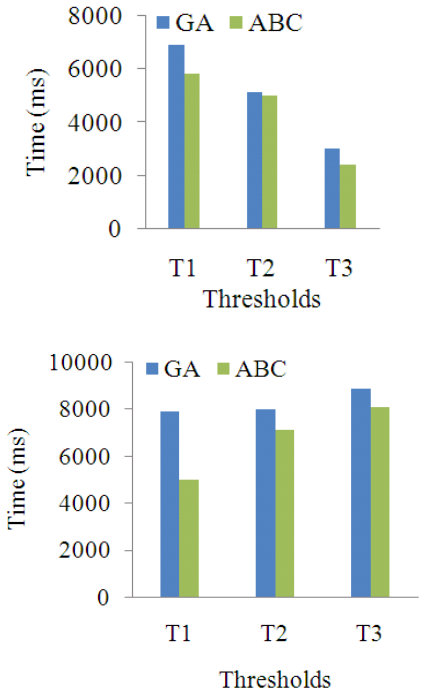

Fig. 5: Time of deduplication at 10 iterations

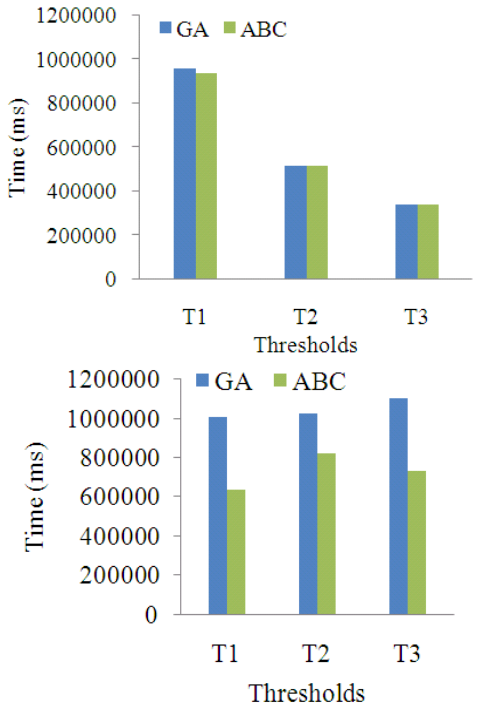

Fig. 6: Time of deduplication at 100 iterations

Comparative study: This section provides a comparative analysis of the proposed algorithm with the genetic programming method (Moises *et al*., 2011). The analysis is based on accuracy of the algorithm and the time for execution of the algorithm. The comparative study represents the responses of the proposed cosine similarity and concept similarity measure with different datasets, namely Restaurant and CORA.

Accuracy based analysis:

Analysis of the various algorithms are listed in the below graphs: The above plotted graphs are the accuracy percentage of the proposed algorithm and the genetic algorithm. The plot in Fig. 1 shows the accuracy of the two different algorithms on the basis of the number of iterations under threshold T1 (1.25). The plot in Fig. 2 and 3 represents the same for thresholds T2 (1.5) and T3 (1.75). In all the cases, it is evident that our proposed algorithms possess more accuracy on compared to the existing algorithm.

Time based analysis: The above analysis is based on the time taken for the deduplication proposed by the proposed algorithms and the genetic algorithm. The three Fig. 4-6 are plotted by varying the number of iterations under three threshold values. The analysis showed that the proposed algorithms, concept similarity based method and cosine similarity method consumes

less time for the deduplication purpose than the genetic algorithm.

CONCLUSION

The deduplication has been one of the most emerging techniques for data redundancy and duplication. The methodology we proposed to avoid the duplication is the ABC algorithm, which provides better performance and accuracy than the genetic algorithm based techniques. The experimentation of the proposed algorithms showed significant results. We used the Restaurant and Cora dataset to evaluate the performance of the two algorithms and the results showed that, the proposed ABC algorithm has better results than the genetic algorithm based technique. We have evaluated the dataset on the basis of accuracy and time consumed for the deduplication purposes.

REFERENCES

Ahmed, K. Elmagarmid, Panagiotis G. Ipeirotis and Vassilios S. Verykios, 2007. Duplicate record detection: A survey. IEEE Trans. Knowl. Data Eng., 19: 1-16. DOI: 10.1109/TKDE.2007.250581

Bhagwat, D., K. Eshghi, D.D. Long and M. Lillibridge, 2009. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. Proceedings of the 17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, (MASCOTS '09), London, UK.

Bolosky, W.J., S. Corbin, D. Goebel and J.R. Douceur, 2000. Single instance storage in Windows® 2000. Proceedings of the 4th Conference on USENIX Windows Systems Symposium, (WSS '00), USENIX Association Berkeley, CA, USA, pp: 2-2.

Douceur, J.R., A. Adya, W.J. Bolosky, D. Simon and M. Theimer, 2002. Reclaiming space from duplicate files in a serverless distributed file system. Proceedings of the 22nd International Conference on Distributed Computing Systems, (ICDCS' 02), ACM, USA., pp: 617-617.

Dubnicki, C., L. Gryz, L. Heldt, M. Kaczmarczyk and W. Kilian *et al*., 2009. Hydrastor: A scalable secondary storage. Proccedings of the 7th Conference on File and Storage Technologies, (FST '09), pp: 197-210.

Dutch, T.M. and W.J. Bolosky, 2011. A study of practical deduplication. ACM Trans. Storage. DOI: 10.1145/2078861.2078864

Ektefa, M., F. Sidi, H. Ibrahim, M.A. Jabar and S. Memar *et al*., 2011. A threshold-based similarity measure for duplicate detection. Proceedings of the IEEE Conference on Open Systems, Sept. 25-28, IEEE Xplore Press, Langkawi, pp: 37-41. DOI: 10.1109/ICOS.2011.6079233

Elhadi, M. and A. Al-Tobi, 2009. Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures. Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology, Nov. 24-26, IEEE Xplore Press, Seoul, pp: 679-684. DOI: 10.1109/ICCIT.2009.235

Gunawi, H.S., N. Agrawal, A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau and J. Schindler, 2005. Deconstructing commodity storage clusters. Proceedings of the 32nd Annual International Symposium on Computer Architecture, Jun. 4-8, IEEE Xplore Press, pp: 60-71. DOI: 10.1109/ISCA.2005.20

Haidarian, S., C. Shahri, B. Lucas and N. Araabi, 2006. Identifying duplicate records by using estimation of distribution algorithms to learn the semantics. Proceedings of the 11th International CSI Computer Conference (CSICC '06), Tehran.

Harnik, D., B. Pinkas and A. Shulman-Peleg, 2010. Side channels in cloud services: Deduplication in cloud storage, IEEE Security Privacy, 8: 40-47. DOI: 10.1109/MSP.2010.187

Karaboga, D. and C. Ozturk, 2010. Fuzzy clustering with artificial bee colony algorithm. Sci. Res. Essays, 5: 1899-1902.

Kumar, J.P. and P. Govindarajulu, 2009. Duplicate and near duplicate documents detection: A review. Eur. J. Sci. Res., 32: 514-527.

Kumbhar, P.Y. and P.S. Krishnan, 2011. Use of Artificial Bee Colony (ABC) algorithm in artificial neural network synthesis. Int. J. Adv. Eng. Sci. Technol., 11: 162-171.

Lillibridge, M., K. Eshghi, D. Bhagwat, V. Deolalikar and G. Trezise *et al*., 2009. Sparse indexing: Large scale, inline deduplication using sampling and locality. Proceedings of the 7th USENIX Conference on File and Storage Technologies, (FAST '09), USENIX Association, pp: 111-123.

Moises, G., D. Carvalho, H.F.A. Laender, M.A. Goncalves and A.S.D. Silva, 2011. A genetic programming approach to record deduplication. IEEE Trans. Knowl. Data Eng., 24: 399-412. DOI: 10.1109/TKDE.2010.234

Muthitacharoen, A., B. Chen and D. Mazieres, 2001. A low-bandwidth network file system. Proceedings of the 18th ACM Symposium on Operating Systems Principles, Oct. 21-24, ACM Press, Banff, Canada, pp: 174-187. DOI: 10.1145/502034.502052

Qingwei, Y., W. Dongxing, Z. Yu and W. Xiaodong, 2010. The duplicated of partial content detection based on PSO. Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, Sept. 23-26, IEEE Xplore Press, Changsha, pp: 350-353. DOI: 10.1109/BICTA.2010.5645302

Quinlan, S. and S. Dorward, 2002. Venti: A new approach to archival storage. Bell Labs, Lucent Technologies.

Rhea, S., R. Cox and A. Pesterev, 2008. Fast, inexpensive content-addressed storage in foundation. Proceedings of the Annual Technical Conference on Annual Technical Conference, (ATC' 08), ACM Press, USA., pp: 143-156.

Samanta, S. and S. Chakraborty, 2011. Parametric optimization of some non-traditional machining processes using artificial bee colony algorithm. Eng. Appli. Art. Intell., 24: 946-957. DOI: 10.1016/j.engappai.2011.03.009

Tan, Y., H. Jiang, D. Feng, L. Tian and Z. Yan *et al*., 2010. SAM: A semantic-aware multi-tiered source de-duplication framework for cloud backup. Proceedings of the 39th International Conference on Parallel Processing, Sept. 13-16, IEEE Xplore Press, San Diego, CA., pp: 614-623. DOI 10.1109/ICPP.2010.69

Ungureanu, C., B. Atkin, A. Aranya, S. Gokhale and S. Rago *et al*., 2010. HydraFS: a high-throughput file system for the HYDRAstor content-addressable storage system. Proceedings of the 8th USENIX Conference on File and Storage Technologies, (FST' 10), USENIX Association Berkeley, CA, USA., pp: 17-17.

Vrable, M., S. Savage and G.M. Voelker, 2009. Cumulus: Filesystem backup to the cloud. ACM Trans. Storage. DOI: 10.1145/1629080.1629084

Winkler, W.E., 2001 Record linkage software and methods for merging administrative lists. The Pennsylvania State University.

Zhu, B., K. Li and H. Patterson, 2008. Avoiding the disk bottleneck in the data domain deduplication file system. Proceedings of the 6th USENIX Conference on File and Storage Technologies, (FAST '08), USENIX Association Berkeley, USA.