A method for duplicate record detection by exploration and exploitation of optimization algorithm

Deepa Karunakaran (Corresponding author), Rangarajan Rangaswamy

Associate Professor, Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, Tamilnadu, India. Email: deepkarun@rediffmail.com
Principal, Indus College of Engineering, Coimbatore, Tamilnadu, India.

Abstract: The duplicate detection is the process of identifying duplicate or redundant information from a set of documents or datasets. A wide variety of methodologies for the identification of duplicate records were projected by numerous researchers. Recently, different optimization algorithms are used for identifying the duplicate records. The optimizing algorithms such as Particle Swarm Optimization, Genetic Algorithm and Artificial Bee Colony provided satisfactory results for duplicate detection process. In this paper, modified algorithms of PSO and ABC are used for the duplicate detection process. The improvements to the algorithms are incorporated by the process of exploration and exploitation. The algorithms are evaluated by their performance on different conditions. The experimentation is conducted based on two datasets namely CORA and RESTAURANT datasets.
[Deepa Karunakaran, Rangarajan Rangaswamy. **A method for duplicate record detection by exploration and exploitation of optimization algorithm.** *Life Sci J* 2013;10(7s):645-653] (ISSN:1097-8135). http://www.lifesciencesite.com. 102

Key words: Duplicate detection, Genetic Algorithm, Particle Swarm Optimization, Artificial Bee Colony, Exploration and Exploitation.

1. Introduction

The increasing volume of information available in digital media has become a challenging problem for data administrators. Usually, data repositories such as those used by digital libraries and e-commerce brokers may present records with disparate structure (de Carvalho, G *et al*, 2011). Today it is possible to say that the ability of an organization to provide valuable services to its users is proportional to the quality of the data available in its systems. In this environment, keeping the repositories with "dirty" data affects the overall speed or performance of data management systems. The solutions available to address this problem requires more technical efforts (de Carvalho. G *et al*, 2011, Koudas. N *et al*, 2006). According to recent statistics, duplicate detection is considered to be the most-impact storage technology and it is estimated to be applied to 75% of all backups in the next few years (Dorward. S *et al*, 2002).

Data duplicate detection strategies can be categorized based on the data units they handle. In this aspect, there are two main data duplicate detection strategies: File-level duplicate detection, in which only a single copy of each file is stored. Two or more files are said to be identical if they have the same hash value. This is a very popular type of service offered in multiple products (Harnik *et al*,2010, Douceur. J. R *et al*, 2002, Gunawi. H. S. *et al*, 2005). Block-level duplicate detection segments files into blocks and stores only a single copy of each block. The system may possibly use fixed-sized blocks or variable-sized chunks (Muthitacharoen.A *et al*, 2001, Vrable. M *et al*, 2009). In terms of the architecture of the duplicate detection solution there are two basic approaches. Target based approach handles duplicate detection by targeting the data-storage device or service, while the client is unaware of any duplicate detection that might occur during the process. Source based duplicate detection occurs at the client side before it transferred to some other source. The client software communicates with the backup server to check the existence of files or blocks (Harnik *et al*,2010). There are two well-known source of de-duplication methods, first is source local chunk-level de-duplication (Tan *et al*, 2010) and source global chunk-level de-duplication (Bhagwat. D *et al*, 2009, Lillibridge. M *et al*, 2009, Ye Qingwei *et al*, 2010) have been proposed in the past to address the above challenges by removing the redundant data chunks before sending them to the remote backup destination.

The recent studies revealed that ABC algorithm is a well performed optimization algorithm. However there is quiet inefficiency in ABC algorithm regarding the solution search equation, which is used to generate new candidate solutions based on the information of previous solutions. It is known that both the exploration and exploitation are necessary for a population based optimization algorithm. In exercise, the exploration and exploitation contradicts to each other. In order to achieve good performance on their optimization problem, both exploration and exploitation should be well balanced.

The recent researches have given many methods for the duplicate detection purposes with many

distinct features by their own.. In this paper, improved algorithms of PSO and ABC are used for the duplicate detection process. The improvements to the algorithms are incorporated by the process of exploration and exploitation. The algorithms are evaluated by their performance on different conditions. The experimentation is conducted based on two datasets namely CORA and RESTAURENT datasets. The performance evaluation has shown that the accuracy of the modified algorithms is better than the original PSO and ABC algorithms.

The rest of the paper is planned as follows. Section 2 provides a review of some related works regarding duplicate detection. Section 3 contains Motivational algorithms behind this research. 4th, 5thand 6th sections give details of the proposed approach with mathematical models. 7th section gives the results and discussion about the proposed approach and with the 8th section gives the conclusion of the research work.

2. Reviews of related works

In recent times, duplicate detection is in distributed manner has attracted researchers significantly due to the demand of scalability and efficiency. Here, the recent works available in the literature for duplicate detection and the different techniques used for it are reviewed.

Guopu Zhu, Sam Kwong (Zhu *et al*, 2003) proposed an artificial bee colony (ABC) algorithm invented recently by Karaboga is a biological-inspired optimization algorithm, which has been shown to be competitive with some original biological-inspired algorithms, such as differential evolution (DE), genetic algorithm (GA) and particle swarm optimization (PSO). However, there is still an insufficiency in ABC algorithm concerning its solution search equation, which is good at exploration but poor at exploitation. Inspired by PSO, they proposed a modified ABC algorithm called gbest-guided ABC (GABC) algorithm by incorporating the information of global best (gbest) solution into the solution search equation to improve the exploitation. Mariem Gzara, Abdelbasset Essabri (Mariem Gzara *et al*, 2011) proposed a most parallel evolutionary algorithm for single and multi-objective optimization is motivated by the reduction of the computation time and the resolution of larger problem. Another promising alternative is to create new distributed schemes that improve the behaviour of the search process of such algorithm. In multi-objective optimization problem, more exploration of the search space is required to obtain the whole or the best approximation of the Pareto Optimal Front. In their paper, they presented a new clustering based parallel multi-objective evolutionary algorithm that balances exploration and exploitation of the search space.

De Carvalho, G *et al* (2011) have proposed a genetic programming approach to record duplicate detection that combines several different parts of evidence extracted from the data content to find a duplicate detection function that is able to identify whether two entries in a repository are replicas or not. This is due to the fact that clean and replica-free repositories not only allow the retrieval of higher-quality information but also lead to more concise data and to potential savings in computational time and resources to process this data. (Luís Leitão *et al*, 2011) have proposed an argument, that structure can indeed have a significant impact on the process of duplicate detection. The proposed method automatically restructures database objects in order to take full advantage of the relations between its attributes. The new structure reflects the relative importance of the attributes in the database and avoids the need to perform a manual selection. Experiments done on several datasets show that, using the new learned structure, they consistently outperform both the results obtained with the original database structure and those obtained by letting a knowledgeable user manually choose the attributes to compare.

Ektefa M *et al* (2011) have proposed a threshold-based method which takes into account both string and semantic similarity measures for comparing record pairs. This method is tried on a real world dataset, namely Restaurant and several standard evaluation metrics were used to measure its effectiveness. The result proved that, the proposed method which is based on the combination of string and semantic similarity measures outperforms the individual similarity measures with the F-measure of 99.1% in Restaurant dataset. (Elhadi.M *et al*, 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 syntactical structures were used as a representation for the documents and such a representation compares and ranks the documents according to the similarity of their representative string using an improved LCS algorithm. Their approach was applied in the filtering of search engine results and in detecting duplicate documents within a corpus.

Karaboga, D. *et al* (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 outperforms fuzzy clustering.

Ye Qingwei Wu *et al* (2010) have proposed a method using hybrid mutation PSO algorithm to examine the most similar partial contents quickly and accurately. The simulation indicate that, the algorithm search the most similar partial contents in the two documents effectively. (Prasanna Kumar *et al*, 2009) have conducted a survey on duplicated web pages that consist of identical structure with different data can be regarded as clones.

3. Motivation

The existence of duplicates is one of the most discussed area in the field of data mining because high quality services like digital libraries and e-commerce brokers may be affected by the existence of duplicates, near-duplicate or quasi-replicas entries in their repositories. Different duplicate detection methods have been introduced to limit amount of occurrence of duplicates. The major problems faced by these algorithms are the lack of optimizing the final outcome. The optimization techniques are based on exploitation and exploration of the solutions obtained from the different duplicate detection algorithms. Recently, (Mariemgzara *et al*, 2011) have proposed a balanced explore – exploit clustering based distributed evolutionary algorithm. The method illustrates incorporating a balanced exploitation and exploration behaviour to the optimization algorithm. Similar way, (Guopu Zhu *et al*, 2010) have proposed modified artificial bee colony for numerical function optimization. In the method gbest, which is adopted from the particle swarm optimization algorithm (PSO), is used to guide the artificial bee colony to perform the optimization by considering the exploration and exploitation problem. Inspired from the above researched, the proposed method improves the optimization techniques by incorporating exploration and exploitation for duplicate detection techniques.

4. Exploration and Exploitation

The main difficulties regarding the modelling of optimization algorithm is in consideration with the balancing of exploration and exploitation. The main objective of the optimization algorithms is to produce optimal solution. The optimal solution defined by an optimization algorithm after ending criteria may be the best that it can produce but it may always not the best for the problem. Thus exploration and exploitation scenarios are introduced to the optimization algorithm. The exploration problem deals with convergence of the unlikely solutions to a single group and finding the best from it. On the other hand, the exploitation of the problem deals with exploring the top solution from the likely solutions.

The Figure 1 represents the exploration and exploitation process with unlikely solutions and likely solutions converge to the goal solution. Here "ul" represents the unlikely solutions, "l" represents the likely solutions and the "goal" represents the goal solutions. When considering the exploration process, the problem is limited by taking a cluster containing the solutions, which are away from the target. In the case of including exploitation problem the top solution from the iterations are taken to cluster and then processed. The proposed method deals with two optimization algorithms, the particle swarm optimization (PSO) and the artificial bee colony algorithm (ABC) algorithms. The algorithms are selected to improve their performance by including the exploitation and exploration problem.

5. Modified PSO algorithm for duplicate detection

The PSO algorithm is characterized by optimizing a number of solutions from a swarm of solutions. The typical mathematical methods used in the PSO algorithm give extra hand for the PSO to differ from other optimization algorithms. The main features of PSO algorithm are the position and velocity of the particle. The initialization and the updation of the velocity and position to particular iteration are done in order to get the problem to be optimized. The number of iteration is set by the user itself because; the user is the one aware about the input data.

Different phases of execution of PSO are discussed in the following section.

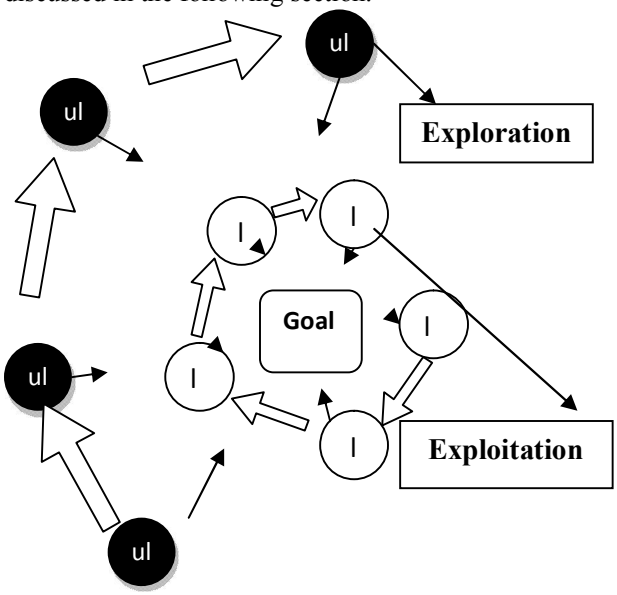

Figure 1. Exploration and Exploitation

5.1. Population

The population in PSO is the expressions used for the validation of duplicates. The population is a swarm of expression specified by the user. The PSO algorithm begins with initializing the population. In the current scenario, the population is considered as the set of expressions, which is used for the validation for duplicates. The input can be represented as in the following set of expressions.

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

The above plot represents the model of population as per the proposed approach. The variables a,b,c,d in the expressions represents the feature vectors obtained after the similarity calculations. This approach generates four similarity values based on Levenshtein distance function and the cosine similarity functions. Every similarity measures generate two similarity values after dividing the record into two parts. With the help of these four similarity values 'a','b','c','d', a set of expression for finding the duplicate or non-duplicate pairs has been developed. The expressions generated from the feature vector are subjected for fitness evaluation.

5.2. Fitness

Every optimization programs are bounded with some fitness functions. The value generated from the fitness function is called fitness value. The proposed approach find the fitness values for the expressions generated for determining the duplicates. The fitness function that is used in the proposed approach is composed of three factors.

The factors are as follows:

True Positives (TP): it is the number of duplicates present in the dataset detected as duplicates.

True Negatives (TN): it is considered as the number of non-duplicates in the datasets detected as non-duplicates.

False Positives (FP): It is considered as the number of duplicates assumed as non-duplicates.

False Negatives (FN): it is considered as the number of non-duplicates assumed as duplicates.

Accuracy: Accuracy is used for the calculation of fitness value. The accuracy is calculated according to values generated from the recall and precision.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Select the population which has been already defined, for calculation of the fitness. Every expression in the population is selected for calculating their fitness. The feature vector is processed with expressions in the population; which produces the count of the duplicates. The result is then processed by means of above furnished values. In accordance with those values, the accuracy value is calculated and thus the fitness. Likewise every expression is treated in same manner and the fitness values of all expression are selected. This process is continued till the iteration specified by the user. The expression with highest fitness value is selected as the global best.

5.3. Optimal solution generation

In the original PSO algorithm the new populations are calculated as per setting the two specifications such as position and velocity of the elements in the population. In different iterations the position and velocity of the particles in the swarm are changed and at the stopping criteria the optimal solution is obtained which has the global best position. Global best is the swarm's best fit value. Deflecting from the original approach, a few improvements to the PSO can enhance the optimal solution. The most important feature of the swarm under consideration is the *P_best,* the best position defined for a particle in the swarm and the *G_best,* the best position defined for particle among the *P_best*. The values of the P_best and the g_best are controlled by their fitness. When considering the PSO algorithm about duplicate detection process, the exploitation problem is solved up to a level, because g_best is obtained from the best of the p_best, which simulates the exploitation process.

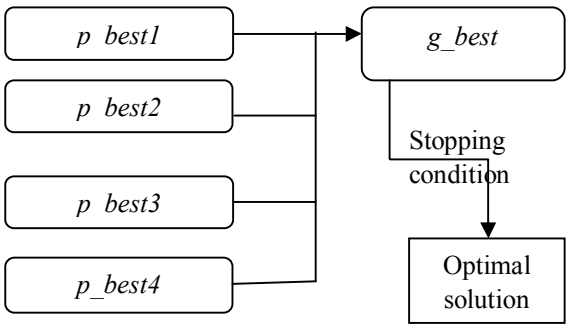

Figure 2. Original PSO

The Figure 2 represents the processing of the original PSO algorithm. The proposed approach includes additional operations to improvise the performance of the duplicate detection process. The original PSO extracts the best position of the particle to find the g_best. Consider,

[p_best: $(a+b)*(c-d)$,p_best: $(a-b)*(c+d)$,p_best: $(a-b)+(c+d)$]➔ [g_best: $(a-b)*(c+d)$]

Here only few operators are included in the particles p_best positions, thus the optimal solution obtained may or may not be the best. So in order to obtain optimal solution maximum possibilities of the operators should be checked. i.e. exploring the maximum possibilities to find the optimal solution for the duplicate detection process.

The modified algorithm considered multiple p_best values for a particle in a single iteration. The initials p_best values for a particle are random generated values. Once all particles are ready with their p_best values, a set of corresponding p_best values are calculated for particle in association with p_best values of other particles. These corresponding p_best values will explore the maximum probability of the obtaining the optimal solution. Let P be the set of p_best value of all the particles.

$P=[p_1,p_2,...,p_n]$, and each p_best,defined as p, values are derived to corresponding p_best value according to other p_best values.

$$p_i \rightarrow [+,*,...] :: [p_{i-1}, p_{i+1},..., p_n] \rightarrow P_i^{corresponds}$$

Where p_i is the p_best value under consideration andp_{i-1}, $p_{i+1,...,}$ $p_n$is the p_best values of the other particles. The corresponding p_best values are generated by incorporating the operators and the other p_best values. Once the corresponding p_best values are generated, a best among those for a particular particle is selected and stored in a cluster. The best value of the stack cluster is selected as the g_best. Once the g_best is calculated the a crossover and mutation function is applied to the current g_best. The crossover and mutation is applied to anyone of the operator in the g_best, to ensure that exploitation is also complete in the modified PSO algorithm for duplicate detection. The crossover and mutation of g_best is subjected check, whether the optimal solution will get improved or not.With this step a single iteration is completed for the modified PSO. Now new population are to be generated for ensuring the credibility of the method.

The new populations are generated for the finding the best fit expressions among the other expressions in the population. The new populations are calculated as per setting two specification described by the PSO algorithm. The specifications are position and velocity of the elements in the population, in the current scenario, the expressions are considered as the elements of the population. Initially, set the velocity and position of each expression with a value range in between 0 and 1. The velocity of the particle is defined as the following,

$$v_{ex} = v^0 + \varphi.(pbest - pos^0) + \phi(gbest - pos^0)$$

Where, v^0 is the current velocity, p_best is current best position of the particle, pos^0 is the current position of the particle, $gbest$ is the best position of a particle in the swarm. The p_best and g_best are calculated based on the position of the particle and the velocity is used to update the position of the particle.

$$pos = pos^0 + v_{ex}$$

Every particles has a best position in its swarm called p_best , if the updated position is greater than the p_best , then it is considered as the p_best . The best value among the p_best is considered as the g_best value of the whole swarm (population). If the updated position is greater than the g_best , then it is considered as the g_best . This process is continued up to a termination criteria are met, mostly the number of iterations is considered as the termination criteria.

Algorithm mod_PSO(pos, p_best, g_best)
Input group of expressions
Output non-duplicate expressions

Step1. **Select** input
Step2. **Create** random population
Step3. **Initialize** population
Step4. **Generate** random p_best values

$$P=[p_1,p_2,...,p_n]$$

Step5. **Calculate** fitness
Step6. **Calculate** corresponding p_best

$$p_i \rightarrow [+,*,...] :: [p_{i-1}, p_{i+1},..., p_n] \rightarrow P_i^{corresponds}$$

Step7. **Find**g_best

$$P_i^{correspond\,s} \Rightarrow p_i \leftarrow g\_best$$

Step8.**Apply** crossover/mutation

$$g\_best \Leftarrow crossover / mutation$$

Step9. **Generate** new populations

$$v_{ex} = v^0 + \varphi.(p\_best - pos^0) + \phi(g\_best - pos^0)$$

$$pos = pos^0 + v_{ex}$$

Step10. **Calculate** fitness

Step11. If new_fitness>old_fitness

Replace g_best

Step12. **Repeat** step4 to step11 until stopping criteria

Step13. **Stop**

6. Modified ABC algorithm for duplicate detection

The ABC algorithm is one of the newly introduced optimization algorithm, the algorithm is introduced in 2005 by Karaboga. The ABC algorithm is characterized by optimizing a number of solutions according to the foraging feature of the bees.

The typical mathematical method used in the ABC algorithm gives 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 incorporates some additions to the original ABC algorithm, in which each cycles are characterised by multilayer layer processing. The aim behind the improvement in the ABC algorithm is that the exploration and exploitation can improve the determination of optimal solutions.

6.1. Employee-bee Phase

The employed bee phase is the process starting phase of the ABC algorithm. The population considered for optimization should be initialized first for the processing of ABC algorithm. Then a random set of expressions are considered as the employed bees. Once the employed bees are defined, the next step is to find the fitness of the employed bees. The fitness function of the employed bees are predefined and which is a common calculation for all the bees namely, onlooker and scout bees. Since the initial population is randomly generated, the relevance of obtaining more proper optimal solution is less. Thus the approach explores the maximum chances of the employed bees according to other employed bees present in the colony. The process is similar to the PSO's p_best calculation. Each employed bee is derived to a set of corresponding employed bees. Consider E is the set of employed bees and each employed bee is represented as e_i and i ranging from 0 to n, where n is the total number of employed bees.

$$E = [e_1, e_2, ..., e_n]$$

An employed bee e_i is converted to a set of corresponding e_i based on the other e_i values. The corresponding e_i values are created based on the operators in the expressions and other expressions since an expression is considered as the e_i according to the duplicate detection problem.

$$e_i \rightarrow [+,-,*,....] :: [e_{i-1}, e_{i+1}, ..., e_n] \Rightarrow E_i^{correspond}$$ W

here, $E_i^{corresponds}$ is the set of corresponding e_i values of the employed bee e_i. Now the best value from the corresponding e_i values are selected as the best fit employed bee from the initial population. The so created employed bees are called e_{best}.

6.2. Onlooker bee and scout bees

The employed bee phase is followed by the onlooker bee phase. This phases is the replacement of new population generation phase of the PSO algorithm. The e_{best} values of the employed bee phase are considered as the input values of the onlooker bee phase. The each e_{best} are updated using the position update feature of the original ABC algorithm. The position update is defined as following,

$$[e_{best} \leftarrow v_i = v^0 + \varphi(v^0 - v_k)] \rightarrow updatedto \rightarrow o_i$$

Here the o_i represents the onlooker bees generated from the e_{best} values and the values v_i represents the new position of the bee, considered as onlookers. The values φ and k are random values and φ ranging from 0 to 1. The generated o_i values are stored in cluster O, which consitsts all the o_I values of the corresponding bee colony.

$$O = [o_1, o_1, ..., o_n]$$

Once the o_i values are generated, they are subjected to operator update process, i.e. the any operator of the o_i changed to another operator, which enhances the exploration process. This process generates another set of onlooker bees and the newly generated onlookers are subjected to fitness calculation as defined by the ABC algorithm. The onlooker with best fitness among others is selected as the best onlooker or solution of the current cycle.

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 cycle is considered as abandon cycle.

The role of the scout bee occurs when an abandon phase is happened, when there is no improved solution after the end of the cycle, a solution is randomly added to the bee colony and

processed. According to the proposed algorithm, the randomly introduced bee is processed through steps of onlooker bee processing and then the best value is introduced in to the bee colony. This process continues till the occurrence of an improved solution.

Algorithm Modified ABC
Input group of expressions
Output non-duplicate expressions

Step1. **Select** input
Step2. **Create** random population
Step3. **Initialize** employed bees
Step4. **Generate** random e_i values

$$E = [e_1, e_2, ..., e_n]$$

Step5. **Calculate** fitness

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

Step6. **Calculate** corresponding e_i values

$$e_i \rightarrow [+, -, *, ...] : [e_{i-1}, e_{i+1}, ...e_n] \Rightarrow E_i^{corresponsa}$$

Step7. **Find** e_{best}

Step8. **Select** e_{best} as onlookers

$$[e_{best} \leftarrow v_i = v^0 + \varphi(v^0 - v_k)] \rightarrow updatetb \rightarrow o_i$$

Step9. **Apply** operator update process on o_i
Step10. **Calculate** fitness of o_i
Step11. Select $o_{best.}$
Step11. If new_fitness>old_fitness
Replace o_{best}
Step12. **Repeat** step4 to step11 until stopping criteria
Step13. **If** abandon cycle
Introduce scout bee s_i
Step14. Apply operator update process on s_i
Step15. $S_i \rightarrow$ bee colony
Step16. **Repeat** step4 to step11 until stopping criteria
Step17. stop

7. Result and Discussion

The comparative analysis provided the analysis of the modified ABC and PSO algorithm with the original ABC and PSO algorithms. The same CORA and RESTAURENT datasets are considered for the comparative analysis. The analysis conducted based on the thresholds 1, 1.5 and 2.

7.1. Accuracy based analysis on RESTUARANT dataset

Experiment 1: Table 1 and Figure 3 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T1. The accuracy of modified PSO on different iterations is 8% greater than the PSO and modified ABC maintained the same accuracy as ABC.

Table 1. Accuracy based on T1

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	76.8	80.6	79.8	78
10	79.6	82	85.6	82
100	80.6	94	95	91

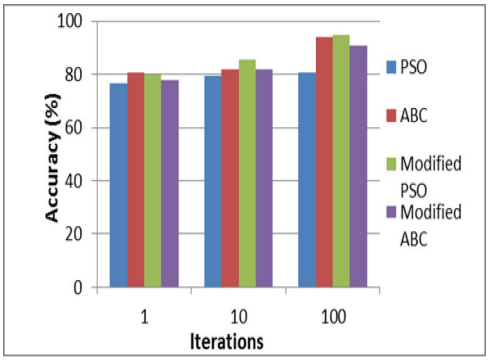

Figure 3. Accuracy based on T1

Experiment 2: Table 2 and Figure 4 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T2. The accuracy of modified PSO on different iterations is 8% greater than the PSO and modified ABC maintained the same accuracy as ABC.

Table 2 Accuracy based on T2

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	76.9	78.6	79.8	79
10	80.6	84	86.6	82.6
100	81.6	92	96.4	92

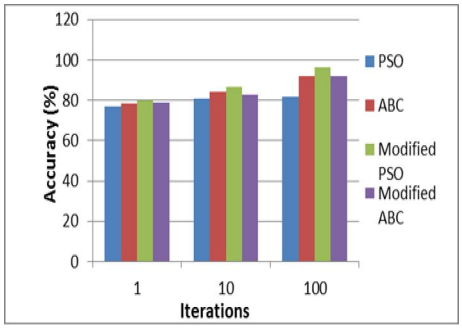

Figure 4. Accuracy based on T2

Experiment 3: Table 3 and Figure 5 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T3. The accuracy of modified PSO on different iterations is

14% greater than the PSO and modified ABC is 4% greater than the ABC.

Table 3. Accuracy based on T3

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	69.6	79	79.8	78
10	72.8	79.6	85.6	82
100	76.6	81	95	91

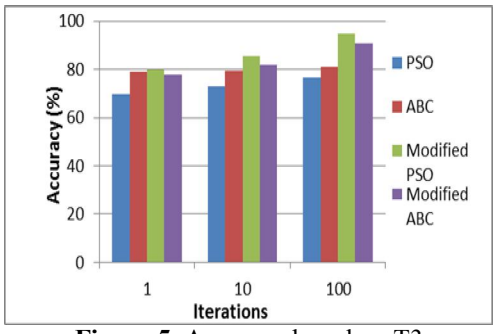

Figure 5. Accuracy based on T3

7.2. Accuracy based analysis on CORA dataset

Experiment 4: Table 4 and Figure 6 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T1. The accuracy of modified PSO on different iterations is 2% greater than the PSO and modified ABC is 3% greater than the ABC.

Table 4. Accuracy based on T1

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	79	82	81	86.3
10	82	86.8	85	91.8
100	92	96	92	97

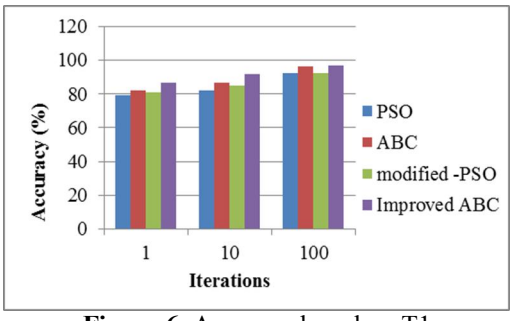

Figure 6. Accuracy based on T1

Experiment 5: Table 5 and Figure 7 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T2. The accuracy of modified PSO on different iterations is 4% greater than the PSO and modified ABC is 5% greater than the ABC.

Table 5. Accuracy based on T2

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	80	81	82	86.3
10	81	85	86	92.8
100	90	95	95	97

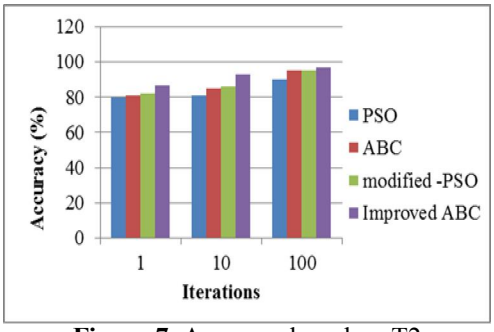

Figure 7. Accuracy based on T2

Experiment 6: Table 6 and Figure 8 shows the modified PSO and ABC algorithms outperformed original PSO and ABC on Threshold T3. The accuracy of modified PSO on different iterations is 2% greater than the PSO and modified ABC is 4% greater than the ABC.

Table 6. Accuracy based on T3

Iterations	PSO	ABC	Mod-PSO	Mod-ABC
1	78	81	81	86.3
10	80	88	85.5	91.8
100	94	92	92	94

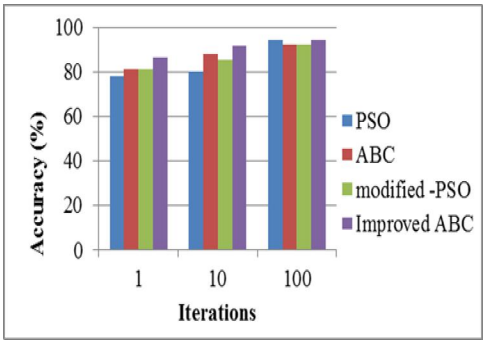

Figure 8. Accuracy based on T3

8. Conclusion

The duplicate detection is one of the most discussing problems in the field of information retrieval. Even though the original ABC algorithm and the PSO algorithm provides better performance and accuracy than the genetic algorithm based techniques, some improvements to the original methods gives even better response for duplicate

detection. Improvements could be done by incorporating exploration and exploitation process.

References

[1]. Moises G. de Carvalho, Alberto H. F. Laender, Marcos Andre Goncalves & Altigran S. da Silva (2011). A Genetic Programming Approach to Record Deduplication, IEEE Transaction on Knowledge and Data Engineering, 399-412.

[2]. N. Koudas, S. Sarawagi & D. Srivastava, (2006). Record linkage: similarity measures and algorithms, Proceedings of the *ACM SIGMOD* International Conference on Management of Data, 802–803.

[3]. S. Dorward & S. Quinlan (2002). Venti: A new approach to archival data storage, Proceedings of the 1st USENIX Conference on File and Storage Technologies FAST '02, CA.

[4]. Danny Harnik, Benny Pinkas & Alexandra Shulman Peleg (2010). Side channels in cloud services, the case of deduplication in cloud storage, IEEE Security and Privacy Magazine, special issue of Cloud Security, 8,2,40-47.

[5]. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon & M. Theimer (2002). Reclaiming space from duplicate files in a serverless distributed file system, Proceedings. of the International Conference on Distributed Computing Systems (ICDCS 2002), Vienna, Austria.

[6]. H. S. Gunawi, N. Agrawal, A. C.Arpaci-Dusseau, R. H. Arpaci- Dusseau & J. Schindler (2005). Deconstructing commodity storage clusters, Proceedings of the 32nd annual International symposium on Computer Architecture ISCA '05. Washington, DC, USA:IEEE Computer Society, 60–71.

[7]. A. Muthitacharoen, B. Chen & D. Mazieres (2001). A low-bandwidth network file system, Proceedings of the eighteenth ACM symposium on Operating systems principles SOSP '01, 174–187.

[8]. M. Vrable, S. Savage & G. M. Voelker (2009). Cumulus: File system Backup to the Cloud, Proceedings of the 7th conference on File and storage Technologies FAST '09,CA.

[9]. Yujuan Tan, Hong Jiang, Dan Feng, Lei Tian, Zhichao Yan & Guohui Zhou (2010). SAM: A Semantic-Aware Multi-Tiered Source De-duplication Framework for Cloud Backup, International Conference on Parallel Processing (ICPP), San Diego, CA, 614-623.

[10]. D. Bhagwat, K. Eshghi, D. D. Long & M. Lillibridge (2009). Extreme Binning: Scalable, Parallel Deduplication for Chunkbased File Backup," HP Laboratories, Tech. Rep. HPL-2009- 10R2.

[11]. M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise & P. Campbell (2009). Sparse Indexing: Large scale, inline deduplication using sampling and locality, Proceedings of the 7th conference on File and storage Technologies, FAST '09, CA,111-123.

[12]. Ye Qingwei, Wu Dongxing, Zhou Yu & Wang Xiaodong (2010). The duplicated of partial content detection based on PSO, IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, Changsha, 350 – 353.

[13]. Guopu Zhu & Sam Kwong (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation,217, 3166-3173.

[14]. Mariem Gzara & Abdelbasset Essabri (2011). Balanced Explore- Exploit Clustering based Distributed Evolutionary Algorithm for Multi-objective Optimisation, International transaction on Studies in informatics and control,20, 2, 97-106.

[15]. Luís Leitão & Pável Calado (2011). Duplicate detection through structure optimization, ACM International Conference on Information and knowledge management, 443-452.

[16]. Ektefa, M, Sidi. F, Ibrahim. H, Jabar. M.A., Memar. S & Ramli. A (2011). A threshold-based similarity measure for duplicate detection, IEEE Conference on Open systems, 37-41.

[17]. Elhadi. M, Al-Tobi. A (2009). Duplicate Detection in Documents and WebPages Using Improved Longest Common Subsequence and Documents Syntactical Structures, International Conference on Computer Sciences and Convergence Information Technology, 679 - 684.

[18]. Karaboga, D. & C. Ozturk (2010). Fuzzy clustering with artificial bee colony algorithm, Sci. Res. Essays, 5: 1899-1902.

[19]. J Prasanna Kumar & P Govindarajulu (2009). Duplicate and Near Duplicate Documents Detection: A Review, European Journal of Scientific Research, 32,514-527.

5/12/2013