MATHEMATICAL MODELING AND COMPUTING, Vol. 8, No. 4, pp. 573–583 (2021)

PSOBER: PSO based entity resolution

Aassem Y., Hafidi I., Khalfi H., Aboutabit N.

National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco

(Received 23 May 2021; Accepted 7 June 2021)

Entity Resolution is the task of mapping the records within a database to their corresponding entities. The entity resolution problem presents a lot of challenges because of the absence of complete information in records, variant distribution of records for different entities and sometimes overlaps between records of different entities. In this paper, we have proposed an unsupervised method to solve this problem. The previously mentioned problem is set as a partitioning problem. Thereafter, an optimization algorithm-based technique is proposed to solve the entity resolution problem. The presented approach enables the partitioning of records across entities. A comparative analysis with the genetic algorithm over datasets proves the efficiency of the considered approach.

Keywords: *entity resolution, cluster validity index, particle swarm optimization, distance measure, genetic algorithm, unsupervised algorithm.*

2010 MSC: 68P05, 68P15 **DOI:** 10.23939/mmc2021.04.573

1. Introduction

Nowadays, the amount of data is growing exponentially. These data represent multiple entities. Sometimes we may want to obtain the data related to a specific entity. Nonetheless, this is not an easy task, because multiple entities can share similar information as the name per example.

In similar cases, it becomes confusing to find the desired result because of the same name that many entities share. Given a database and a set of records, the goal is to partition the records into clusters such that each cluster refers to a single real-world entity. This task of assigning records to real-world objects is dubbed entity matching. By definition, entity matching is the task of linking records to their corresponding entities. This process has several names: Object distinction [1], Duplicate detection, Entity resolution [2], Merge/Purge problem [3], Entity Name Disambiguation [4].

This issue has drawn the attention of researchers, and different approaches are proposed to solve it. Among those methods, we distinguish Sorted Neighborhood Methods [5], Blocking [6] algorithms and Machine Learning based-approaches [1]. The problem of Entity Resolution can be considered as a partitioning problem where the objective is to partition the records so that all the records within the same partition should belong to one entity. Some of the suggested techniques for Entity Resolution [5–7] partition records automatically without providing the number of clusters beforehand, while other techniques require the number of clusters to be given as input parameter [8].

Previous researchers have utilized many approaches using genetic programming GP [9–11]. The first approaches [12, 13] were supervised in nature. Next, an unsupervised approach [14] is proposed using the search capabilities of the genetic algorithm to solve the entity resolution problem, where the problem is considered as a partitioning problem. However the limitation of this work is that it targets only bibliographic databases which are quite different from other databases.

Our work is the first one to adapt Particle Swarm Optimization technique (PSO) [15] which is unsupervised in nature to efficiently solve the problem of entity matching problem for generic datasets. We keep the same modeling as in [14] of the problem as an optimization problem and applied PSO to solve it.

To measure the quality of the resulting partitioning there are used internal validity measures [16]. The aim of that optimization problem is to optimize the internal cluster validity index. In the current paper, we have proposed a PSO based framework for solving the entity resolution problem. In order to encode the clusters we have utilized medoid based representation. There are used different similarity functions to measure the similarity between two records. In order to check the goodness of the encoded partitioning, some cluster validity indices are used. Also it's established a comparative performance study of the utilized cluster validity indices.

The main contribution of the current paper is the utilization of the PSO algorithm as an unsupervised based approach for solving the entity resolution problem, where no labeled dataset is used. The proposed approach is applied on any dataset that doesn't contain unlabeled data. This eliminates the need for manual annotation in the generation of labelled data. This is, without a doubt, a time-consuming and costly procedure. Another major contribution of the current work is the use of a wide number of features to address the problem of entity matching. The use of a single feature may not permit disambiguating records. In general, there exist many clustering algorithms based on the search capability of the PSO algorithm. The proposed PSO algorithm based technique is automatic in nature. It can divide the given set of records automatically into a number of clusters.

Section 2 briefly outlines some interesting work in the area of the entity matching. In Section 3, we present our proposed approach. In Section 4, cluster validity measures are described, which serve as objective functions. Experimental evaluation is discussed in Section 5. Finally, Section 6 concludes the paper and gives the future direction to the work.

2. Related works

Many researchers have studied entity matching, which is a well-known problem. The researchers have suggested a number of solutions to this problem. Two broad categories can be identified among the available methods. The first deals with non-temporal attributes. These methods are used for records attributes which have no meaningful significance for the time. The second approach deals with temporal attributes. In this category, records evolve over the time. Consequently, it is crucial to include the time as a dimension in the calculation of distance between compared records.

Most of the previously proposed approaches were dealing with bibliographic dataset. Yin et al. suggested DISTINCT [1], a methodology that considered a set of distinct objects as the training set and applied SVM algorithm to identify several types of linkage. However, the dependency of that methodology on the training dataset was one of its major limitations. Also, the method cannot consider clusters with single records, which is a very likely case in almost all datasets.

Most of the previous work was related to bibliographic databases. A probabilistic approach was proposed by Tang et al. [17] which requires the number of entities k to be supplied beforehand which is unpractical. To overcome this limitation authors have tried to estimate k [18–20]. Another work done by Wang et al. [21] to identify best similarity function and threshold to maximize an objective function for entity matching. Other approaches like Bibnet Miner, DBLife and SAND [22–24] were suggested for the case of bibliographic datasets, which makes them not appropriate for generic datasets.

Recently, researches were carried out to measure the quality of algorithms developed for entity matching. Mishra et al. utilized some of internal cluster validity indices [16, 25–27] for evaluating correctness of the obtained partitioning. Bic index was used by Tang et al. to predict the number of clusters. A later work done by Mishra [28] consists of proposing two other cluster validity indices. Nevertheless, these indices were used to assess the quality of clustering only for bibliographic databases.

Researchers incorporated genetic programming (GP) [29], which is inspired from the genetic algorithm [30] for entity matching problems. A GP-based approach [31] was presented for de-duplication. In [32] GenLink, which is a genetic programming based supervised learning algorithm was proposed. It learns linkage rules from a set of existing reference links. Mishra et al. [14] proposed an unsupervised method for entity matching problems using genetic algorithm. Being dedicated specifically to bibliographic databases is one of the major limitations of this approach. In this paper, we have proposed an unsupervised method for the entity matching problem using Particle Swarm Optimization (PSO).

3. The proposed approach

This section describes the particle swarm optimization (PSO) for entity matching problems. The basic steps of PSO are shown in Algorithm 1.

Algorithm 1 PSO Algorithm

```
1. Initialization
        1.1 For each particle i in a swarm population size $P$:
          1.1.1 Initialize $X_i$ randomly
          1.1.2 Initialize $V_i$ randomly
          1.1.3 Evaluate the fitness $f(X_i)$
          1.1.4 Initialize $pbest_i$ with a copy of $X_i$
2. Repeat until a stopping criterion is satisfied.
        2.1 For each particle i:
          2.1.1 Update $V_i^t$ and $X_i^t$
          2.1.2 Evaluate the fitness $f(X_i^t)$
          2.1.3 $pbest_i \leftarrow X_i^t$ if $f(pbest_i) < f(X_i^t)$
          2.1.3 $gbest_i \leftarrow X_i^t$ if $f(gbest_i) < f(X_i^t)$
```

Instead of a chromosome in Genetic Algorithm, a population in a PSO algorithm is a set of particles. Each particle has a position vector X_i that represents its corresponding solution. In each iteration a particle updates its position vector based on the combination of its previously found best position and the best point achieved regardless of which particle had found it. Consequently, all particles take advantage of their search capabilities plus the search capabilities of each other, which helps to converge to the best solution over iterations.

The complete flowchart of the algorithm is shown in Fig. 1.

In the preprocessing step, the data is mapped to its numerical representation as explained in the next subsection. Next, the algorithm initializes a population of n particles. The associate vectors of position X_i and velocity V_i of each particle are initialized randomly. The update of these vectors for each particle is performed at each iteration. The self cognitive p_{best_i} at a current generation holds reference to the best current position that optimize the particle's fitness value $f(X_i^t)$. Similarly, in terms of fitness value, the social cognitive $g_{best_i^t}$ holds the best position in the current population. While the stop condition is not met, the update process is repeated. After all iterations have been completed, the best position vector is chosen. Following that, reverse mapping is performed to obtain the corresponding data clusters.

3.1. Records mapping

Let $R = \{r_1, \ldots, r_n\}$ be the set of n records of a dataset. Each record in the dataset is encoded by a unique integer ranging from 1 to n.

Example 1. Let $R = \{r_1, \ldots, r_{10}\}$ be a set of 10 records. Their encoding is $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Record r_2 is presented by the integer 2.

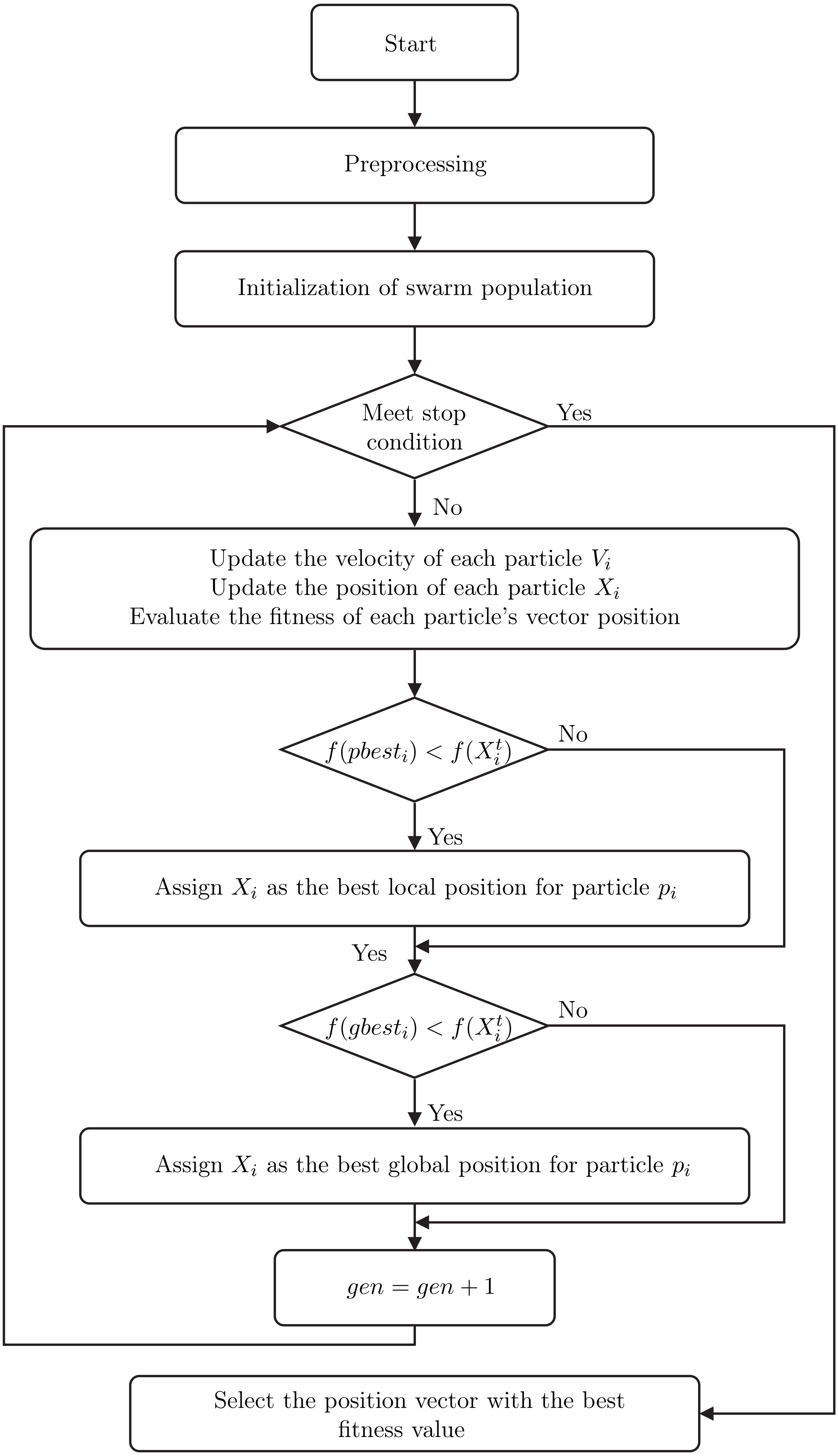
Start
Preprocessing
Initialization of swarm population
Meet stop condition
Yes
No
Update the velocity of each particle V_i
Update the position of each particle X_i
Evaluate the fitness of each particle's vector position
$f(pbest_i) < f(X_i^t)$
No
Yes
Assign X_i as the best local position for particle p_i
Yes
$f(gbest_i) < f(X_i^t)$
No
Yes
Assign X_i as the best global position for particle p_i
$gen = gen + 1$
Select the position vector with the best fitness value

Fig. 1. Flowchart of PSO based ER algorithm.

3.2. Solution representation

The solution of a problem resolved by PSO is a vector X of a length K. The vector X is a sequence of some positive integers representing the clusters. We utilize variable length encoding to encode the clusters. This is because the number of clusters is not known in advance. The k length encoding represents the k clusters. The elements in cluster representatives are not cluster centers, rather they are medoid points of different clusters.

The number of clusters varies between 1 and n. Thus, the size of cluster representatives should vary between 1 and n. We have varied it between 2 and $n-1$. This is because for the boundaries it is difficult to measure the quality of partitioning. We discarded these cases in the current work since they are very rare.

3.3. Population initialization

As the vector X_i for a particle is a sequence of cluster representatives, that component needs to be initialized to form a complete vector X. The K records encoded in cluster representatives are initialized to k randomly chosen records from the dataset. For each particle the value of k is selected randomly. This is a random number between 2 and $n-1$. These k records should be unique.

3.4. Assignment of records to different clusters

After generating the vector for each particle of the population, the next step is to assign the rest of the records to representative cluster medoids in order to form the complete partitioning. The process of record assignment is summarized in Algorithm 2.

Algorithm 2 Assignment of records to different clusters

```
input : positionVector: a vector of length K
        $\mathbb{R}$: a vector of length n
$Cluster[\,] \leftarrow \varnothing$
for $i = 1$ to $K$ do
    $Cluster[i] \leftarrow Cluster[i] \cup positionVector_i$
    $unassignedRecord \leftarrow \mathbb{R} \backslash positionVector_i$
end
for $i = 1$ to $n - K$ do
    Assign the i-th records from $unassignedRecord$ to cluster $j$ such that
    $d(unassignedRecord, Cluster[j]) \leqslant d(unassignedRecord, Cluster[k]), 1 \leqslant j, k \leqslant K, j \neq k$
end
return Cluster
```

In the proposed approach, the distance is calculated using the function described in section 4. The distance between a record r and a cluster U_k is the distance between r and its nearest record within U_k. The distance between record r and cluster U is denoted by the equation below.

$$d = \min_{r_i \in U} \operatorname{dist}(r, r_i).$$

4. Objective functions and distance measures used

4.1. Objective functions

Entity matching is essentially a task of records clustering so that similar records are grouped in the same cluster, where each cluster refers to a distinct entity. Given a set of records $R = \{r_1, \ldots, r_n\}$.

Let the number of clusters be K. Let the set of clusters $U = \{u_1, \ldots, u_K\}$. The size of a cluster U_k is n_k. Let c_k be the center of cluster U_k $\forall k \in \{1, \ldots, K\}$. C is the center of all records.

The clustering of these records in k clusters satisfies the following rules:

$$\cup_{k=1}^{K} U_k = \mathbb{R},$$

$$U_k \cap U_k' = \varnothing, \qquad 1 \leqslant k \neq k' \leqslant K.$$

In order to assess the quality of a resulting partitioning, cluster validity indices [16, 25–28] are suggested by researchers community. These cluster validity indices permit to quantify and measure the quality of the obtained partitioning. Thus, a set of partitioning may be ranked based on the goodness of each solution. In the literature, several cluster validity indices have been developed each with its own definition. In our work, we have utilized two of the cluster validity indices as the objective functions. We have described the used cluster validity indices in more details in this section. At a time, PSO is utilized to optimize any one of these cluster validity indices.

4.1.1. CH index

The Calinski–Harabasz index [16] is computed by the following equation:

$$\mathrm{CH} = \frac{\operatorname{trace}(S_B)}{\operatorname{trace}(S_W)} \times \frac{n-k}{k-1},$$

$$\operatorname{trace}(S_B) = \sum_{k=1}^{K} n_k \times \operatorname{dist}(c_k, c),$$

$$\operatorname{trace}(S_w) = \sum_{k=1}^{K} \sum_{r_q \in C_k}^{K} n_k \times \operatorname{dist}(r_q, C_k).$$

$\operatorname{trace}(S_B)$ is related to the distance between clusters and should be maximized. $\operatorname{trace}(S_W)$ is the internal distance which should be minimized. Thus better partitioning is achieved by maximizing the value of the index.

4.1.2. CS index

The index's numerator computes the distance between records in the same cluster of the resulted partitioning. The numerator which refers to the compactness should be minimized. The denominator computes the separation and it is formulated by the distance between two different clusters. Consequently, its value should be maximized. Therefore, for a better partitioning the index value should be minimized,

$$\mathrm{CS} = \frac{\sum\limits_{k=1}^{K} \left[\frac{1}{n_k} \sum\limits_{r_q \in C_k} \max\limits_{r_r \in C_k} \operatorname{dist}(r_k, r_r) \right]}{\sum\limits_{k=1}^{K} \left[\min\limits_{k' \in K,\, k' \neq k} \operatorname{dist}(c_k, c_k') \right]}.$$

4.2. Distance measures

Accurate clustering necessitates a precise description of the closeness of two objects, either in terms of pairwise similarity or distance. Meanwhile, dissimilarity or distance are often used to describe similarity [33–35]. To determine pairwise distances, measurements like Euclidean distance and relative entropy were used in clustering.

The distance measures can be broadly classified into two categories which are non temporal distance measures and temporal distance measures. From the first category we cite Levenshtein [33], NGram [34], Metric Longest common subsequence (MLCS) [35], Jaccard distance etc.

Regarding temporal distance measures, they are characterized by the consideration of the time in the distance calculation. A useful case for using this kind of distance measure is when dealing with databases related to data that may evolve over time such as bibliographic datasets.

In our paper we utilize only non temporal distance measures. The used metrics are Levenshtein distance, MLCS and NGram distance.

5. Experimental evaluation

This section describes the experimental setting and evaluation of the proposed approach. We have used both PSO and genetic algorithms as the underlying optimization techniques. The following are the parameter values: no. of generation = 100, population size = 100, mutation probability = 0.1, crossover probability = 0.9. PSO parameters values are: $w = 0.9$, $c_1 = 2$, $c_2 = 2$.

For evaluation purposes, we used a synthetic dataset generated using Febrl [36] and a dataset that we have generated manually. The information regarding used datasets is given in the table below.

Table 1. Description of datasets used for evaluation purpose.

Datasets	Number of records	Number of entities
Febrl dataset	1000	500
Rating	50	12

5.1. Evaluation

Two objective functions are utilized for experimental purposes, which are discussed in section 4. The results of the proposed approach are compared with genetic algorithm based ER.

We conducted two experiments. The first one consists of measuring the convergence of both algorithmes to 100% in terms of aforementioned metrics (Precision, Recall, FScore). The goal of this experiment is to highlight the fast convergence of PSO relatively to GA. The results are reported in Tables 2 and 3.

The second experiment aims at measuring the performance of both algorithms for a specific number of generations. We show the results in Tables 4, 5 and 6.

Table 2. Required iterations for our proposed approach.

Number of iterations		5	10	15	20	21
PSOBER	Precision	73.33	57.14	77.77	71.42	100
	Recall	91.66	66.66	63.63	83.33	100
	Fscore	81.48	61.53	70	76.92	100

Table 3. Required iterations for genetic algorithm based approach.

Number of iterations		5	10	15	20	21	25
GAEM	Precision	46.15	50	66.66	45	66	100
	Recall	54.54	50	66.66	45	66	100
	Fscore	50	50	66.66	45	66	100

For the synthetic dataset, Tables 2 and 3 show the minimal number of required iterations to gain respectively the precision, recall and FScore values for our approach compared with genetic algorithm based approach. The results show non-monotonic variation of the values for both algorithms. However, our approach requires few iterations to cover all true duplicates. This out-performance is for reason that PSO has the capability of sharing information across particles and thereby helping them improve their own solution. This results in faster convergence for the solutions in PSO. In contrast, genetic algorithms have no such mechanism. Better solutions pass the information only by participating in the crossover with some chromosomes.

Table 4. Precision values for the compared algorithms using CH and CS indices.

Distance Measures	CVI	Levenshtein	MLCS	NGRAM
PSOBER	CH	**96.59**	95.34	95.34
	CS	**95.55**	92.30	92.30
GAEM	CH	94.04	**94.11**	94.04
	CS	87.5	**92.5**	86.11

Table 5. Recall values for the compared algorithms using CH and CS indices.

Distance Measures	CVI	Levenshtein	MLCS	NGRAM
PSOBER	CH	**96.59**	95.34	95.34
	CS	**97.72**	92.30	92.30
GAEM	CH	94.04	**94.11**	94.04
	CS	89.74	**92.5**	86.11

Table 6. FScore values for the compared algorithms using CH and CS indices.

Distance Measures	CVI	Levenshtein	MLCS	NGRAM
PSO	CH	**96.59**	95.34	95.34
	CS	**96.62**	92.30	92.30
GA	CH	94.04	**94.11**	94.04
	CS	88.6	**92.5**	86.11

Table 4 shows the result of our proposed approach in terms of precision for three different distance measures using respectively CH index and CS index as the objective function. This table also contains the precision values for Genetic Algorithm based approach.

From this table, it is clear that the performance using both indices are approximately similar for each method. The values of precision for both indices are near to 100% for PSO and GA based approach. However, PSO still outperforms GA for the three distance measures. This is due to the search capabilities of PSO which utilize the previous best collective and individual solution in the construction of the new solution. In contrast, genetic algorithms are based on randomness which has a significant impact on its precision values. To show the impact of distance measure, we compared the algorithms using different distance measures. For instance, PSO achieves its highest precise value using Levenshtein distance. On the other hand, Metric Longest Common Subset (MLCS) is the convenient distance measure that permits attaining the highest precision value for the genetic algorithm being applied to this dataset.

For recall values, it is clear that the aim is to maximize its value. This can be achieved by merging all records in a single cluster. Nevertheless, for this case, the precision will be too low. Thus, to ensure better performance of the algorithm, we should find a compromise between precision and recall so that they are both maximized. This constraint is what makes clustering a crucial problem. When precision and recall are equal to 100%, the obtained results match the gold standard.

Table 5 shows the values of recall for PSO and GA using respectively CH index and CS index. PSO takes advantage of its search strategy to cover more true positives than genetic algorithm. In other words, our approach covers more correct duplicates than the genetic algorithm based approach.

Moreover, we see that both distance measures and the used objective function impact the approaches' performances in terms of recall. For CS index there is a clear difference between both algorithms performances using Levenshtein distance measure. The former distance produces better results for PSO, while at the same time it decreases the performances of GA based approach.

The obtained results concerning FScore are shown in Table 6. Since the mathematical definition is a harmonic mean of precision and recall, it is clear that it is linear with the algorithms regarding precision and recall. Concerning the distance measure, PSO is more efficient using Levenshtein while genetic algorithm achieves its highest value when utilizing MLCS as a distance measure.

6. Conclusion and future work

In the current paper, we have proposed a PSO algorithm based approach for solving the entity matching problem. The proposed approach can automatically determine the number of clusters present in a regular dataset as well as determine the optimal partitioning from the dataset. Several attributes are considered for finding the distance between different records. A representation was adapted to cohere with PSO specification. All these concepts have made the proposed approach a very powerful one. The results of the proposed technique in comparison with the genetic algorithm over two datasets using two objective functions prove the efficiency of the proposed technique.

[1] Yin X., Han J., Yu P. S. Object Distinction: Distinguishing Objects with Identical Names. IEEE 23rd International Conference on Data Engineering. 1242–1246 (2007).

[2] Christen P., Goiser K. Quality and Complexity Measures for Data Linkage and Deduplication. Quality Measures in Data Mining. 127–151 (2007).

[3] Hernández M. A., Stolfo S. J. The merge/purge problem for large databases. ACM SIGMOD Record. **24** (2), 127–138 (2007).

[4] Mishra S., Mondal S., Saha S. Entity matching technique for bibliographic database. Database and expert systems applications. DEXA 2013. 34–41 (2013).

[5] Draisbach U., Naumann F., Szott S., Wonneberg O. Adaptive Windows for Duplicate Detection. 2012 IEEE 28th International Conference on Data Engineering. 1073–1083 (2012).

[6] Christen P. Data Matching: Concepts and Techniques for Record Linkage. Entity Resolution and Duplicate Detection. Springer (2012).

[7] Aassem Y., Hafidi I., Aboutabit N. Enhanced Duplicate Count Strategy: Towards New Algorithms to Improve Duplicate Detection. NISS2020: Proceedings of the 3rd International Conference on Networking, Information Systems & Security. Article No. 58, 1–7 (2020).

[8] Benkhaled H., Berrabah D., Boufares F. A novel approach to improve the Record Linkage process. 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT). 1504–1509 (2019).

[9] De Carvalho D. M., Laender A. H. F., Goncalves M. A., Da Silva A. S. A genetic programming approach to record deduplication. IEEE Transactions on Knowledge and Data Engineerin. **24** (3), 399–412 (2012).

[10] Isele R., Bizer C. Learning expressive linkage rules using genetic programming. Proceedings of the VLDB Endowmen. **5** (11), 1638–1649 (2012).

[11] Lyaqini S., Nachaoui M., Quafafou M. Non-smooth classification model based on new smoothing technique. Journal of Physics: Conference Series. **1743** (1), 012025 (2021).

[12] Golberg D. E. Genetic algorithms in search, optimization, and machine learning. Addion Wesley Professional (1989).

[13] Ribeiro Filho J. L., Treleaven P. C., Alippi C. Genetic algorithm programming environments. Computer. **27** (6), 28–43 (1994).

[14] Mishra S., Saha S., Mondal S. GAEMTBD: Genetic algorithm based entity matching techniques for bibliographic databases. Applied Intelligence. **47**, 197–230 (2017).

[15] Eberhart R. C., Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. 39–43 (1995).

[16] Caliński T., Harabasz J. A dendrite method for cluster analysis. Communications in Statistics. **3** (1), 1–27 (1972).

[17] Tang J., Zhang J., Yao L., Li J., Zhang L., Su Z. Arnetminer: extraction and mining of academic social networks. KDD '08: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 990–998 (2008).

[18] Tang J., Fong A. C. M., Wang B., Zhang J. A unified probabilistic framework for name disambiguation in digital library. IEEE Transactions on Knowledge and Data Engineering. **24** (6), 975–987 (2012).

[19] Wang X., Tang J., Cheng H., Yu P. S. ADANA: Active name disambiguation. 2011 IEEE 11th International Conference on Data Mining. 794–803 (2011).

[20] Nachaoui M. Parameter learning for combined first and second order total variation for image reconstruction. Advanced Mathematical Models & Applications. **5** (1), 53–69 (2020).

[21] Wang J., Li G., Yu J. X., Feng J. Entity matching: how similar is similar. Proceedings of the VLDB Endowment. **4** (10), 622–633 (2011).

[22] Sun Y., Wu T., Yin Z., Cheng H., Han J., Yin X., Zhao P. BibNetMiner: mining bibliographic information networks. SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 1341–1344 (2008).

[23] DeRose P., Shen W., Chen F., Lee Y., Burdick D., Doan A., Ramakrishnan R. DBLife: A community information management platform for the database research community. CIDR. 169–172 (2007).

[24] Jin H., Huang L., Yuan P. Name disambiguation using semantic association clustering. 2009 IEEE International Conference on e-Business Engineering. 42–48 (2009).

[25] Mishra S., Saha S., Mondal S. Cluster validation techniques for bibliographic databases. Proceedings of the 2014 IEEE Students' Technology Symposium. 93–98 (2014).

[26] Rousseeuw P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. **20**, 53–65 (1987).

[27] Xie X. L., Beni G. A validity measure for fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. **13** (8), 841–847 (1991).

[28] Mishra S., Saha S., Mondal S. On validation of clustering techniques for bibliographic databases. 2014 22nd International Conference on Pattern Recognition. 3150–3155 (2014).

[29] Cramer N. L. A representation for the adaptive generation of simple sequential programs. Proceedings of the First International Conference on Genetic Algorithms. 183–187 (1985).

[30] Holland J. H. Adaptation in natural and artificial systems. MIT (1975).

[31] De Carvalho M. G., Laender A. H., Goncalves M. A., Da Silva A. A genetic programming approach to record deduplication. IEEE Transactions on Knowledge and Data Engineering. **24** (3), 399–412 (2012).

[32] Isele R., Bizer C. Learning expressive linkage rules using genetic programming. Proceedings of the VLDB Endowment. **5** (11), 1638–1649 (2012).

[33] Wagner R. A., Fischer M. J. The String-to-String Correction Problem. Journal of the ACM. **21** (1), 168–173 (1974).

[34] Kondrak G. N-gram similarity and distance. Proceedings of the 12th international conference on String Processing and Information Retrieval. 115–126 (2005).

[35] Hsu W. J., Du M. W. Computing a longest common subsequence for a set of strings. BIT Numerical Mathematics. **24**, 45–59 (1984).

[36] Christen P., Churches T. Febrl–Freely extensible biomedical record linkage. ANU Computer Science Technical Reports (2002).

PSOBER: пов'язування об'єктів на основі PSO

Аассем Й., Гафіді І., Халфі Г., Абутабіт Н.

Національна школа прикладних наук, Університет імені Султана Мулея Слімана, Хурібга, Марокко

Пов'язування об'єктів — це задача зіставлення записів у базі даних з відповідними об'єктами. Задача пов'язування об'єктів є множиною задач через відсутність повної інформації в записах, варіантний розподіл записів для різних об'єктів, а іноді і перекривання записів різних об'єктів. У цій роботі запропоновано метод вирішення цієї проблеми без необхідності зовнішнього контролю. Вищезгадана задача подається як задача про розбиття. Після цього, запропоновано методику на основі алгоритму оптимізації для вирішення задачі пов'язування об'єктів. Запропонований підхід дозволяє визначити розподіл записів за категоріями. Порівняльний аналіз із генетичним алгоритмом за наборами даних доводить ефективність запропонованого підходу.

Ключові слова: *пов'язування об'єктів, індекс валідності кластера, метод рою частинок, міра відстані, генетичний алгоритм, некерований алгоритм.*