



## Research Article

# Optimizing Ontology Alignment through Linkage Learning on Entity Correspondences

Xingsi Xue <sup>1,2,3</sup> Chaofan Yang <sup>1,2,3</sup> Chao Jiang <sup>1,2,3</sup> Pei-Wei Tsai <sup>4</sup>  
Guojun Mao <sup>2</sup> and Hai Zhu <sup>5</sup>

<sup>1</sup>Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian 350118, China

<sup>2</sup>School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou, Fujian 350118, China

<sup>3</sup>Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou, Fujian 350118, China

<sup>4</sup>Department of Computer Science and Software Engineering, Swinburne University of Technology, John Street, Hawthorn, Victoria 3122, Australia

<sup>5</sup>School of Network Engineering, Zhoukou Normal University, Zhoukou, Henan 466001, China

Correspondence should be addressed to Xingsi Xue; jack8375@gmail.com

Received 10 January 2021; Revised 25 January 2021; Accepted 27 January 2021; Published 5 February 2021

Academic Editor: Abd E.I.-Baset Hassanien

Copyright © 2021 Xingsi Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data heterogeneity is the obstacle for the resource sharing on Semantic Web (SW), and ontology is regarded as a solution to this problem. However, since different ontologies are constructed and maintained independently, there also exists the heterogeneity problem between ontologies. Ontology matching is able to identify the semantic correspondences of entities in different ontologies, which is an effective method to address the ontology heterogeneity problem. Due to huge memory consumption and long runtime, the performance of the existing ontology matching techniques requires further improvement. In this work, an extended compact genetic algorithm-based ontology entity matching technique (ECGA-OEM) is proposed, which uses both the compact encoding mechanism and linkage learning approach to match the ontologies efficiently. Compact encoding mechanism does not need to store and maintain the whole population in the memory during the evolving process, and the utilization of linkage learning protects the chromosome's building blocks, which is able to reduce the algorithm's running time and ensure the alignment's quality. In the experiment, ECGA-OEM is compared with the participants of ontology alignment evaluation initiative (OAEI) and the state-of-the-art ontology matching techniques, and the experimental results show that ECGA-OEM is both effective and efficient.

## 1. Introduction

Semantic Web (SW) is proposed by Tims Berners-Lee in 1998, which makes the intelligent applications be able to understand a word's meaning in semantic level. Ontologies are the solution to the issue of data heterogeneity on SW since it is able to make consensus of a certain conception meaning of a field and provide abundant domain knowledge and semantic vocabulary for the interaction between application systems. However, due to SW's scattering essence, there might be different definitions on a concept in separate ontologies, which leads to the issue of ontology heterogeneity [1]. Ontology matching is regarded as

an effective method to address it, and swarm intelligent algorithm- (SIA-) based ontology matching techniques have achieved good performance in past studies [2], such as genetic algorithm (GA) [3], particle swarm optimization algorithm (PSO) [4], firefly algorithm (FA) [2], and artificial bee colony algorithm (ABC) [5]. However, there are two drawbacks in the existing SIA-based approaches: (1) massive time and memory consumption is required, which heavily blocks the efficiency of the ontology matching process; (2) an expert of related field or a reference alignment is required in the process of ontology matching which is usually not available in real application conditions. To overcome these drawbacks, an extended

compact genetic algorithm-based ontology entity matching technique (ECGA-OEM) is proposed in this work, which uses both compact encoding mechanism [6, 7] and linkage learning approach to efficiently match the ontologies. In particular, our contributions are as follows:

- (i) A new evaluating metrics on the ontology alignment is proposed, which is able to work without the reference alignment and the domain experts.
- (ii) An optimal model on ontology entity matching problem is constructed.
- (iii) An ECGA-OEM is proposed, which uses the linkage learning and compact encoding mechanism to efficiently address the ontology entity matching problem.

The rest of this paper is organized as follows: the related works are narrated in Section 2; the statement of ontology, ontology matching, and similarity measures are presented in Section 3; the ontology entity matching through ECGA proposed by this paper is revealed in Section 4; Section 5 presents the experiment results; and finally, Section 6 draws the conclusion and presents the future work.

## 2. Related Works

Ontology matching is a complex, time-consuming, and error-prone work, especially when the scale of ontologies is large. Recently, a number of the machine learning (ML) techniques [8–14] have been proposed to automatically determine the ontology alignment. To improve the matching efficiency, Araújo et al. [15] presented the matching system through parallel computing (PC) technique and Amin et al. [16] matching ontology based on cloud computing (CC). At the same time, SIA-based technique has achieved great performance in the ontology matching [1, 2, 17–20] domain [21–26].

Generally, ontology matching techniques are classified into two categories: ontology metamatching techniques and ontology entity matching techniques [27]. The former dedicates to address the problem that how to aggregate different similarity measures with appropriate weights, and the latter tries to directly determine the entity correspondence set between two ontologies. The first SIA-based ontology metamatching system is genetics for ontology alignment (GOAL), which aims at optimizing the aggregating weight set for different matchers [3, 28–30]. Memetic algorithms (MAs), which introduce local search (LS) strategy into evolutionary algorithms (EAs) to improve its local optimization capability, are proposed to solve ontology metamatching problem [31]. To overcome the drawback of overreliance reference alignment, Xue et al. presented a partial reference alignment (PRA), in which only a part of standard reference is used to assess the quality of alignment [32]. Furthermore, Xue and Wang proposed an innovative metric named unanimous improvement ratio (UIR) to assess the alignment’s quality, in which the reference alignment is not required [33]. Besides, artificial bee colony (ABC) algorithm is also adopted to address ontology metamatching problem, which further improves the solution’s quality [5].

During the matching process, ontology metamatching techniques need to maintain several similarity matrices, which leads to huge memory consumption. For this reason, ontology entity matching techniques, which aims at directly determining the optimal pair set, attracts the expert’s interests. Genetic algorithm-based ontology matching (GAOM) firstly regards certain matching pairs set as the optimizing objective [34]. MA is also utilized to solve the ontology entity matching problem, whose performance outperforms GA [35]. Bock et al. [4] use PSO to solve the ontology entity matching. In detail, it evaluates the fitness of chromosomes through a certain aggregation strategy for multiple objective functions. Alves et al. [36] argue that instances consisting in the ontology can be used to improve alignment in the condition that knowledge is embedded in them. For this reason, Xue et al. [37] take also instance-level matching into consideration to further improve the quality of alignment.

## 3. Preliminaries

### 3.1. Ontology and Ontology Matching

*Definition 1* (ontology). An ontology  $O$  is a 5-tuple [33].  $O = (C, P, I, \Lambda, \Gamma)$  where  $C$  is a set of classes that cannot be empty,  $P$  is a set of properties that cannot be empty,  $I$  is a set (it could be empty) of individuals that represent the instances of classes in the real world,  $\Lambda$  is a nonempty set of axioms that are used to check the consistency of ontologies or deduce new information, and  $\Gamma$  is a set of annotations that provide information metadata so that the researcher can understand. Particularly,  $C$ ,  $P$ , and  $I$  make up the entities in ontologies.

*Definition 2* (ontology matching). The ontology matching can be considered as a function  $f(O_1, O_2, A', p, r)$ , where  $O_1$  and  $O_2$  are two ontologies to be matched;  $A'$  is an existing initial alignment of  $O_1$  and  $O_2$ ;  $p$  is a set of parameters, e.g., threshold, in the process of ontology matching; and  $r$  is a set of external resources, e.g., background knowledge based and dictionaries, which assisted in ontology matching. The process of ontology matching is depicted in Figure 1, where  $A$  is the obtained ontology alignment.

An example of matching two ontologies is presented in Figure 2, where  $O_1$  and  $O_2$  are two ontologies to be matched in this figure. The strings in the rounded rectangle are the classes, e.g., “Reference,” “Entry,” and “Book.” The black lines between two classes of the same ontology represent their relation “has a” or “is a” in turn; e.g., “Reference” has a “Book” and “Book” is a “Reference,” which means “Reference” is the supclass of “Book,” and “Book” is the subclass of “Reference.” There are datatype properties that describe the features of a class; e.g., “Data,” “Title,” and “Human Creator” are the properties of “Reference.” The instances of a class are in a rectangle; e.g., “Of Natures Obvious Laws & Processes in Vegetation” is an instance of “Article.” The relation of the entities between the two ontologies are linked by the lines with double arrowhead, and there are symbols: “ $\equiv$ ,” “ $\subseteq$ ” (or “ $\supseteq$ ”), and “ $\perp$ ,” which, respectively, means equivalence, more

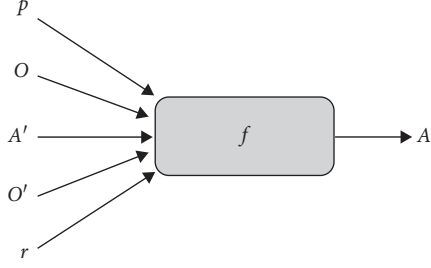


FIGURE 1: The process of ontology matching.

specific (or less specific), and disjointness relation; e.g., classes “Entry” and “Book” of  $O_2$  are the equivalent and hyponym of class “Reference” of  $O_1$ , property “Event” of  $O_1$ , and property “Year” of  $O_2$  are irrelevant.

**3.2. Similarity Measures.** It is necessary to measure to what extent two ontology entities are similar before finding the reliable entity correspondences. In the ontology domain, we usually use the similarity measure to calculate two entities’ similarity values. Generally, there are three categories of similarity measures, i.e., syntactic, linguistic, and taxonomy-based measures [33].

**3.2.1. Syntactic Measures.** Two syntactic measures, i.e., SMOA (string metric for ontology alignment) [37] and Levenshtein [38], are employed in this paper. Given two strings  $s_1$  and  $s_2$ , the SMOA and Levenshtein similarity are, respectively, defined as follows:

$$\text{SMOA}(s_1, s_2) = \text{Comm}(s_1, s_2) - \text{Diff}(s_1, s_2) + \text{WinklerImpr}(s_1, s_2), \quad (1)$$

where  $\text{Comm}(s_1, s_2)$  stands for the common length of  $s_1$  and  $s_2$  while  $\text{Diff}(s_1, s_2)$  for the different lengths and  $\text{WinklerImpr}(s_1, s_2)$  is the improvement to results yielded by the method that proposed by Winkler [39].

$$\text{Levenshtein}(s_1, s_2) = \max\left(0, \frac{\min(|s_1|, |s_2|) - d(s_1, s_2)}{\min(|s_1|, |s_2|)}\right), \quad (2)$$

where  $|s_1|$  and  $|s_2|$  are the cardinality of the letters contained in  $s_1$  and  $s_2$ , respectively, and  $d(s_1, s_2)$  is the number of letters that need to be modified from  $s_1$  to  $s_2$ . The final syntactic similarity is equal to the average of SMOA and Levenshtein.

**3.2.2. Linguistic Measures.** The linguistic similarity between two strings is worked out by considering the semantic relations (such as synonyms and hypernym) which usually requires using the thesaurus and dictionaries. In this work, WordNet [23, 40], an electronic vocabulary database that has collected every meaning of various words, is used. Given two words  $w_1$  and  $w_2$ , Linguistic Similarity ( $w_1, w_2$ ) equals:

- (i) 1, if words  $w_1$  and  $w_2$  are synonyms in Wordnet.

- (ii) 0.5, if word  $w_1$  is the hypernym of word  $w_2$  or vice versa in Wordnet.  
 (iii) 0, otherwise.

**3.2.3. Taxonomy-Based Measures.** The core ideal of taxonomy-based measures is to make full use of the hierarchy relationship of ontology to determine two entities’ similarity by considering their neighbor’s similarity. In this work, a mutual reasoning between class and property (MRCP) is proposed as the taxonomy-based measure, which is shown in Figure 3.

In Figure 3, the circle is the class of the ontology, the triangle is the properties of the ontology, and the one-way arrow represents the hierarchical relationship; i.e., class  $c_{a1}$  is the supclass of class  $c_{a3}$ , the dividing line arrow between the class and the property indicates that the property belongs to this class, the bidirectional arrow indicates that there is a high similarity between the two entities, and the dashed two-way arrow indicates that the similarity between them is improved after the operation. Subgraph (a) depicts the classes’ similarity gained from their neighbor, i.e., supclass and subclass. There is high similarity between classes  $c_{a1}$  and  $c_{b1}$ , so the similarity between their subclasses  $c_{a3}$  and  $c_{b2}$  is supposed to increase. Likewise, similarity of classes  $c_{a3}$  and  $c_{b2}$  would be increased because their subclasses  $c_{a6}$  and  $c_{b4}$  are highly similar. Subgraph (b) is the properties’ similarity gained from their supproperty and subproperty. The similarity of properties  $p_{a3}$  and  $p_{b2}$  will be improved since their supproperties  $p_{a1}$  and  $p_{b1}$  and subproperties  $p_{a6}$  and  $p_{b4}$  are highly similar, respectively. Subgraph (c) is the properties’ similarity gained from the classes which they belong to.  $c_{a1}$  and  $c_{b1}$  are the classes of two ontologies and  $p_{a1}, p_{a2}, p_{a3}, p_{b1}, p_{b2},$  and  $p_{b3}$  are their properties, respectively. The similarities between properties of class  $c_{a1}$  and properties of class  $c_{b1}$  would be improved because of the high similarity of  $c_{a1}$  and  $c_{b1}$ ; i.e., the similarity of  $p_{a3}$  and  $p_{b1}$  would be promoted and so are the remaining eight combinations. On the contrary, classes’ similarity will be increased due to the same or highly similar properties they shared, as is depicted in subgraph (d). Since pairs  $p_{a1}$  and  $p_{b3}, p_{a2}$  and  $p_{b2},$  and  $p_{a3}$  and  $p_{b1}$ , the similarity of  $c_{a1}$  and  $c_{b1}$  is increased too.

**3.2.4. Aggregation Strategy.** Three similarity matrixes are generated when the three measures have been applied. In this work, three matrixes need to be aggregated into one matrix. The final similarity value  $S_a(s_1, s_2)$  between two entities  $s_1$  and  $s_2$  is defined as follows:

$$S_a(s_1, s_2) = \begin{cases} 1, & S_s = 1 \text{ or } S_l = 1, \\ 0.8 * S_{sl} + 0.2 * S_t, & S_{sl} > \text{Threshold}, \\ 0.2 * S_{sl} + 0.8 * S_t, & S_{sl} \leq \text{Threshold}. \end{cases} \quad (3)$$

where  $S_s, S_l,$  and  $S_t$  is, respectively, the syntactic, linguistic, and taxonomy-based similarity of  $s_1$  and  $s_2$ ;  $S_{sl}$  is the average of  $S_s$  and  $S_l$ ; and Threshold is a given parameter to filter the matching pairs with low similarity.

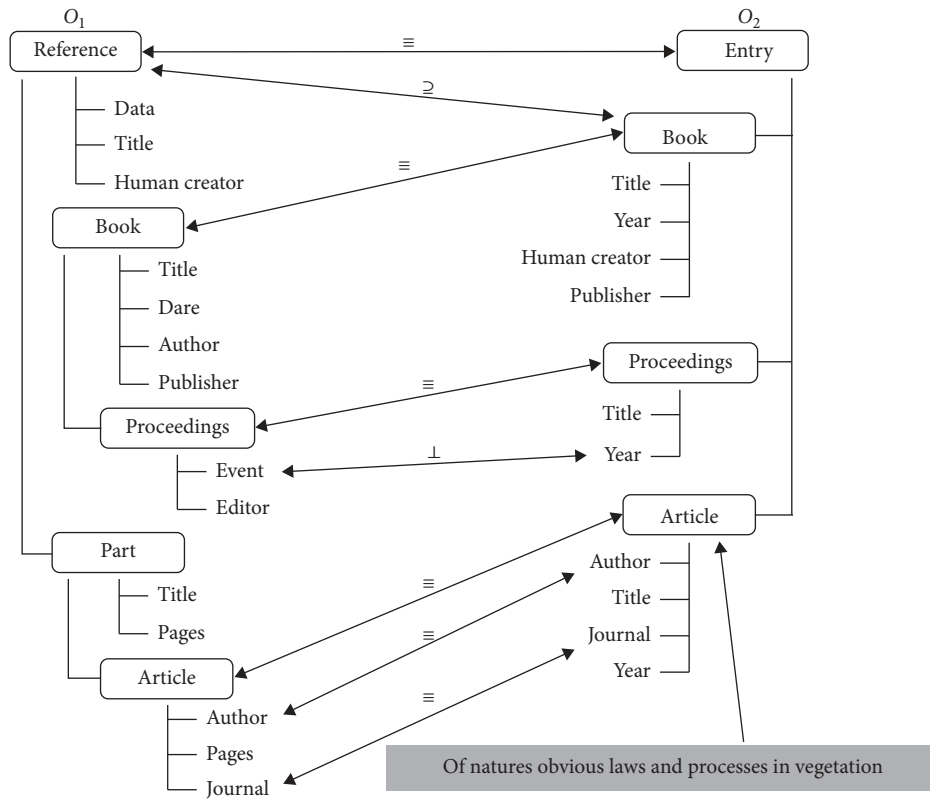


FIGURE 2: An example of matching two ontologies.

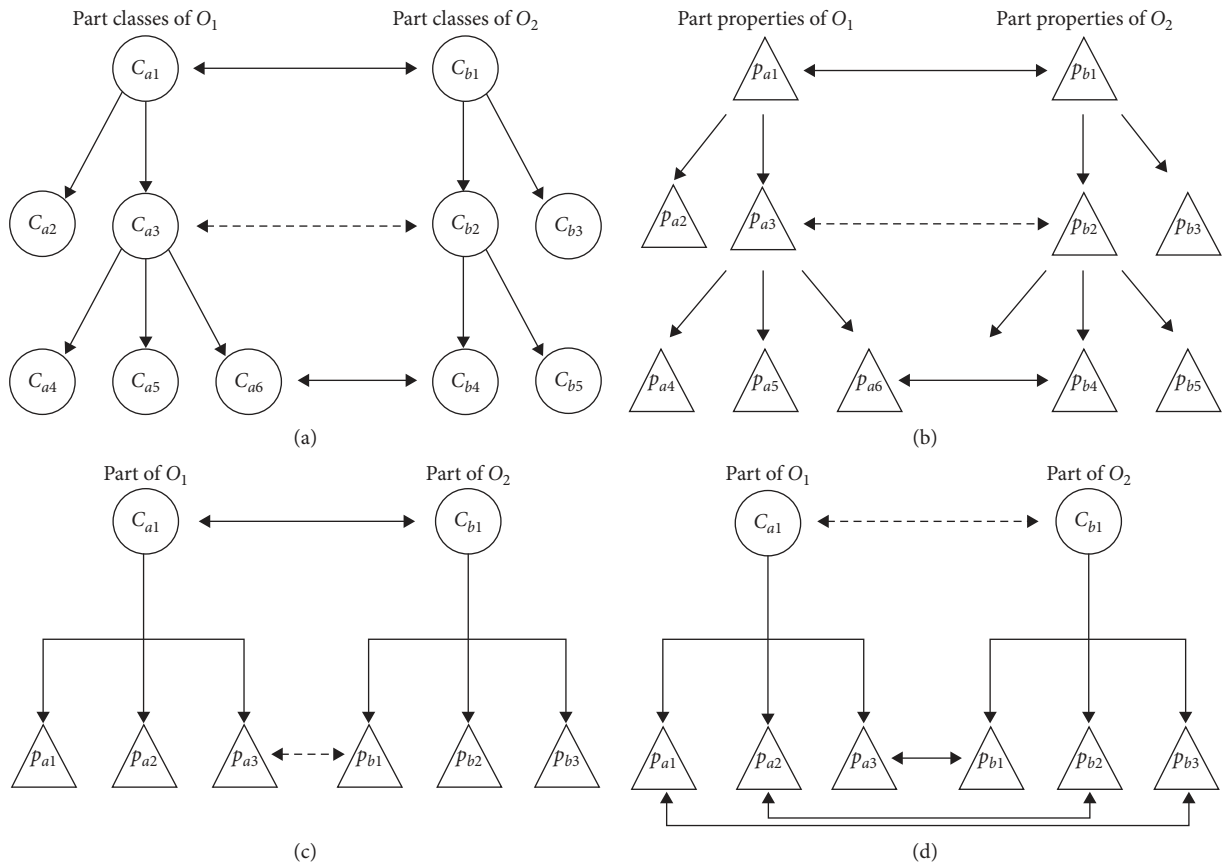


FIGURE 3: MRCP taxonomy-based measure.

## 4. Extended Compact Genetic Algorithm-Based Ontology Entity Matching

GA is an excellent methodology to solve the ontology matching problem due to its potential parallel search characteristic and good searchability. In our work, an ECGA to efficiently address the ontology entity matching problem is proposed.

*4.1. Optimal Model.* The optimal model of ontology entity matching problem is given as follows:

$$\begin{cases} \max & \xi(\sigma), \\ \text{s.t.} & \sigma = (x_1, x_2, \dots, x_{|O_1|})^T, \\ & x_i \in \{1, 2, \dots, |O_2|, -1\}, \quad i = 1, 2, \dots, |O_1|, \end{cases} \quad (4)$$

where  $|O_1|$  and  $|O_2|$ , respectively, are the cardinalities of two ontologies  $O_1$  and  $O_2$ ;  $x_i$ ,  $i = 1, 2, \dots, |O_1|$  is the  $i_{\text{th}}$  matching pairs. Particularly, it means there is no matching of  $i_{\text{th}}$  entity in  $O_1$  when  $x_i = -1$ . The objective of this work is to maximize  $\xi(\sigma)$ , and for the details of it, refer to Section 4.3.3.

*4.2. The Framework of ECGA-OEM.* Two ontologies are to be matched as input and a reference alignment as output, and the framework of ECGA-OEM is shown in Figure 4, whose critical components are narrated in the rest of this section. Two ontologies, generally in XML or RDF format, are extracted into two hierarchy schema in the preprocessing model. The operation of the ECGA optimization model relies on the similarity matrix obtained in the similarity measure model, which has been stated in detail in Section 3.2. Finally, the alignment is generated by the solution generation model. In detail, the ECGA optimization model is described in Section 4.3.

*4.3. ECGA Optimization Model.* Given the virtual population's maximum generation,  $\text{MaxGeneration} = 2000$  (normally, the number of iterations in the convergence of ECGA is much less than  $\text{MaxGeneration}$ ),  $\text{Threshold} = 0.7$ , and the hierarchy schema of ontology1 and ontology2 as input and final alignment as output. The pseudocode of ECGA is proposed in Algorithm 1, where PV and BB are probability vector (see also Section 4.3.1) and building blocks (refer to also Section 4.3.7), respectively.

*4.3.1. Probability Vector Initialization.* Different from binary coding, the probability vector (PV) in this work is two-dimensional. The initialized PV and convergent PV are shown in Tables 1 and 2, respectively. The value in the  $i_{\text{th}}$  row and  $j_{\text{th}}$  column represents the possibility of matching between the  $i_{\text{th}}$  entity in  $O_1$  and the  $j_{\text{th}}$  entity in  $O_2$ ; i.e., it means the probability of 0<sub>th</sub> entity of  $O_1$  (reference) and 0<sub>th</sub> entity of  $O_2$  (entry) is  $1/1 + 1 + 1 + 1 + 1$ , which is shown in

Table 1 (peculiarly, the header “-1 (null)” denotes the probability of no matching). The convergence condition is that the probability of taking a unique number on each locus in the PV is 1; i.e., in Table 2, the probability of the 0<sub>th</sub> entity of  $O_1$  (reference) and the 0<sub>th</sub> entity of  $O_2$  (entry) is  $23.72/0 + 23.72 + 0 + 0 + 0$  and so do the rest.

*4.3.2. Chromosomes Generation.* Certain size chromosomes are produced in each generation through PV. An example of chromosome is given in Figure 5. In particular, subgraph (a) shows the locus of the chromosome and corresponding code, and subgraph (b) illustrates decoding chromosome in subgraph (b); i.e., it denotes that the fourth entity of  $O_1$  “Article” correspondent to the third entity of  $O_2$  “Article” as the code of the fourth locus is “3” (“-1 (Null)” indicates no matching).

*4.3.3. Fitness Function.* The fitness function is used to determine which chromosomes in the population can better adapt the environment. In the context of ontology matching, the objective of fitness function is to find the best chromosome, whose corresponding alignment's quality is the highest, with algorithm convergence. The objective function of the optimal model is used as the fitness function of this work, and given a chromosome  $\sigma$ , its fitness function is defined as follows:

$$\xi(\sigma) = 2 \cdot (\beta \cdot \phi(|A|) + (1 - \beta) \cdot f(A)), \quad (5)$$

where  $\xi(\sigma)$  is the fitness function that is used in this paper;  $A$  is the alignment determined by  $\sigma$ ;  $|A|$  is the cardinality of  $A$ ;  $\beta$ , a fraction in the range  $[0, 1]$ , is the relative weight of  $\phi(|A|)$  and  $f(A)$ , which is set to 0.25 in this paper;  $\phi$  is a normalization function; and  $f$  is a function that calculates the mean of the matched entity pairs' similarity values in  $A$ . In addition,  $\phi(|A|)$  and  $f(A)$  are defined as follows:

$$\phi(|A|) = \frac{|A|}{\min(|O_1|, |O_2|)}, \quad (6)$$

$$f(A) = \frac{\sum_{i=1}^{|A|} \eta_i}{|A|}, \quad (7)$$

where  $|O_1|$  and  $|O_2|$  are, respectively, the cardinality of  $O_1$  and  $O_2$  and  $\eta_i$  is the similarity of the  $i_{\text{th}}$  matching pair in alignment  $A$ . In particular,  $\phi(|A|)$  is the ratio of the number of matching pair found to the value of the smaller entity number of the two ontologies and  $f(A)$  is the average similarity of the matching pairs found, which, respectively, approximates the recall value and precision value.

*4.3.4. Selection Operator.* The selection operator selects the best chromosomes in the current population to participate in the next step [41] and updates the PV. Firstly, the chromosomes are sorted in descending order according to their fitness scores. Secondly, the first half of the chromosomes will be retained as a temporary population. Finally, with fitness as the weight through roulette, we can select the chromosome from the temporary population for subsequent operation.

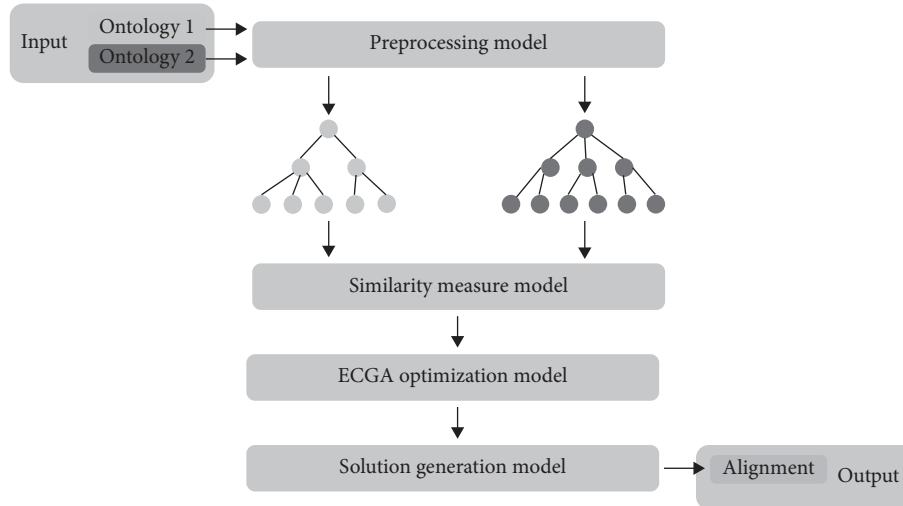
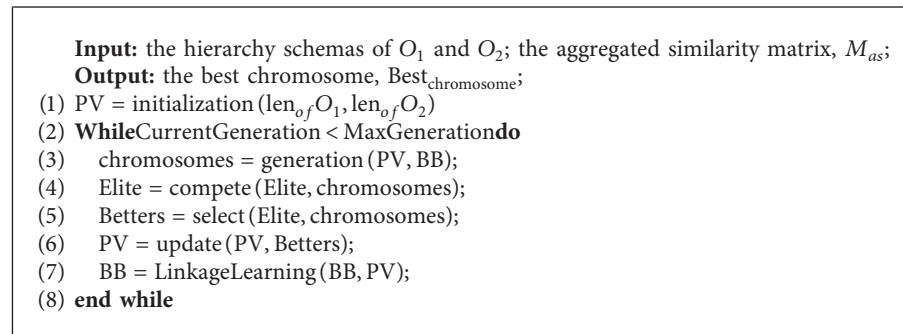


FIGURE 4: The framework of ECGA-OEM.



ALGORITHM 1: ECGA optimization model.

TABLE 1: An example of initialized PV.

PV	0 (Entry)	1 (Book)	2 (Proceedings)	3 (Article)	-1 (Null)
0 (Reference)	1	1	1	1	1
1 (Book)	1	1	1	1	1
2 (Proceedings)	1	1	1	1	1
3 (Part)	1	1	1	1	1
4 (Article)	1	1	1	1	1

TABLE 2: An example of convergent PV.

PV	0 (Entry)	1 (Book)	2 (Proceedings)	3 (Article)	-1 (Null)
0 (Reference)	23.72	0	0	0	0
1 (Book)	0	25.33	0	0	0
2 (Proceedings)	0	0	26.01	0	0
3 (Part)	0	0	0	0	22.46
4 (Article)	0	0	0	24.11	0

**4.3.5. Elite Strategy.** The goal of elite strategy is to keep the historical optimal solution and prevent the fitness of the optimal chromosome from “degenerating” in the process of evolution. In this work, the elite strategy has two steps: the historical optimal solution Elite first competes with the current optimal solution Best, and the winner will become the new Elite; the historical optimal solution then participates in the PV update in each generation.

**4.3.6. Probability Vector Updating.** An example of updating the PV is presented in Figure 6, where subgraph (a) is a chromosome generated by the initialized PV. The similarity is derived from the similarity matrix according to the chromosome’s code. It should be noted that the code of the third locus is “0,” which means that the entity “Part” with sequence number “3” in  $O_1$  does not match any entity in  $O_2$ , so its similarity is equal to 1 minus the value of the highest

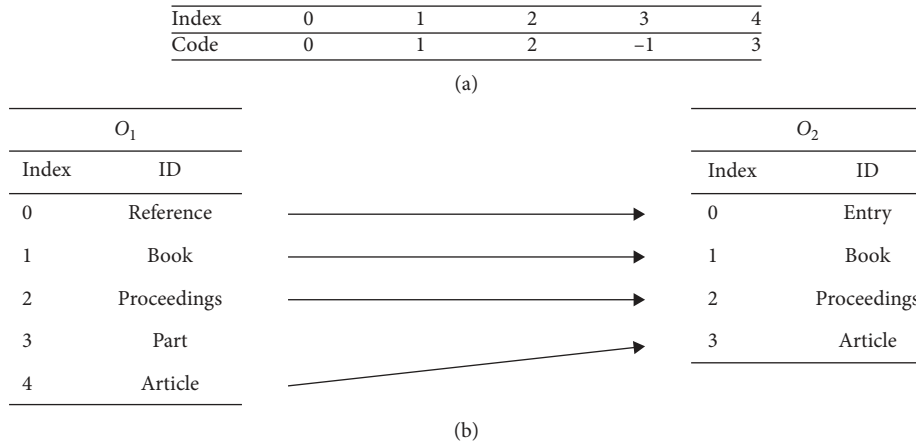


FIGURE 5: An example of chromosome.

Index	0	1	2	3	4
Code	0	1	2	-1	3
Similarity	0.92	1.00	1.00	0.62	1.00
Normalized similarity	0.20	0.22	0.22	0.14	0.22

(a)

PV	0 (Entry)	1 (Book)	2 (Proceedings)	3 (Article)	-1 (Null)
0 (Reference)	<b>1.2</b>	1	1	1	1
1 (Book)	1	<b>1.22</b>	1	1	1
2 (Proceedings)	1	1	<b>1.22</b>	1	1
3 (Part)	1	1	1	<b>1</b>	1.14
4 (Article)	1	1	1	1.22	1

(b)

FIGURE 6: An example of update PV.

similarity between “Part” and all entities in  $O_2$ , i.e.,  $(1 - 0.38) = 0.62$ . PV will be updated by the normalized similarity of each locus, which is shown in subgraph (b). The probability of being updated in the PV is bold; i.e., the probability of matching pair “Reference” and “Entry” changed from “1” to “1.2” since the normalized similarity of the corresponding locus of chromosome is “0.20.”

**4.3.7. Linkage Learning.** Building blocks will be saved through linkage learning, thus to improve the efficiency of algorithm and quality of solution [42]. In simple GA, linkage learning is to identify great locus and protect them so that they will not be destroyed in the subsequent crossover and mutation operations; in ECGA, linkage learning keeps good probability distribution so that they are not disturbed in the subsequent update process. A linkage learning approach is proposed in this work, and its detail is shown in Figure 7.

For clarity, only column 1 of the original probability vector is selected for narration. In each generation, a low probability clearing operation is performed. The value of each column is divided by the maximum value of the row, and the value of the column will be cleared if the decimal is less than a specific value (0.2 in this work); i.e., 0, 2, 3, and -1

bits in PV are cleared and marked. The row of PV is a “good” probability distribution when all but one bit of this row are zero. Link learning generates building blocks based on the “good” probability distribution, i.e., a building block, the pair of index “0,” and code “0,” which was included in the rounded rectangle produced by linkage learning. After that, all the building blocks are directly put into each chromosome (the bold numbers), which reduces the consumption of runtime and memory consumption.

## 5. Experiments and Discussion

**5.1. Experiment Setup.** In the experiment, the Biblio benchmark provided by the Ontology Alignment Evaluation Initiative (OAEI) is used to verify the effectiveness of our approach. Normally, two ontologies to be matched and a reference alignment are included in each testing case as a standard to evaluate the quality of matching results. The testing cases can be classified into five categories, which are briefly described in Table 3.

In this work, the method is compared with the participants of OAEI-, GA-, and CGA-based ontology matching techniques. The experimental results are the H-mean values of 30 independent runs.

PV	0 (Entry)	1 (Book)	2 (Proceedings)	3 (Article)	-1 (Null)
1 (Book)	1.25	25.33	1.36	3.87	2.14

Low probability clearing ↓

PV	0 (Entry)	1 (Book)	2 (Proceedings)	3 (Article)	-1 (Null)
1 (Book)	0	25.33	0	0	0

Linkage learning ↓

Index 1 Code 1 Building block

Chromosome generation ↓

Index	0	1	2	3	4
Code	0	1	2	-1	3

FIGURE 7: An example of linkage learning.

TABLE 3: A brief description of OAEI Biblio benchmark.

Cases	Brief description
101–104	The ontologies to be matched are the same or there is only a slight difference in OWL restriction between them
201–210	The ontologies to be matched have the same structural, but different lexical and linguistic features
221–247	The ontologies to be matched have the same lexical and linguistic, but different structural features
248–266	The ontologies to be matched have different lexical, linguistic, and structure features
301–304	The ontologies in the real world often have strong heterogeneity

5.2. *Alignment Evaluation Metrics according to Reference Alignment.* A criterion is needed when evaluating the quality of matching systems. Given an alignment result  $A$ , two measures, recall and precision, are employed in this work and their formula is as follows [22, 23]:

$$\text{recall}(A) = \frac{|R \cap A|}{|R|}, \quad (8)$$

$$\text{precision}(A) = \frac{|R \cap A|}{|A|}, \quad (9)$$

where  $|R|$  and  $|A|$  are the cardinality of matching pairs in the reference alignment provided in case set and the matching pairs in alignment are produced by the matching system and  $|R \cap A|$  is the cardinality of matching pairs, which exist in both reference alignment and alignment found. It means that all the matching pairs in reference alignment have been found when recall is 1, while all the matching pairs found is correct when precision is 1.

Both recall and precision are important parameters of the evaluation results and they should be considered at the same time. A weighted harmonic mean of recall and precision, F-measure, is used in this work, which is presented in the following equation[43]:

$$f\text{-measure}(A) = \frac{\text{recall}(A) \cdot \text{precision}(A)}{\alpha \cdot \text{recall}(A) + (1 - \alpha) \cdot \text{precision}(A)}, \quad (10)$$

where  $\alpha[0, 1]$  is the relative weight of recall and precision and it is set as 0.5 in this work, which is named  $f_1$ -measure.

5.3. *Comparison with OAEI's Participants.* The participants from OAEI 2016, 2015, and 2014 are selected to compare with our approach. In particular, if a matching system has participated for more than one year, only the latest results are used. The harmonic mean comparison of participants and ECGA-OEM is shown in Figure 8. The vertical axis represents different matching systems, and the horizontal axis represents the score of their corresponding parameters. In terms of  $f$ -measure, ECGA-OEM ranks third, which it is slightly lower than Lily and CroMatch. Wiki has been used as the linguistic measure in Lily and CroMatch, which is able to improve the performance of the algorithm at the expense of efficiency. In addition, ECGA-OEM has an unparalleled performance in maintaining the balance between precision and recall, while one of them is much higher than the other in the participants, which is very important in evaluating results quality.

Further  $f$ -measure comparison of OAEI participants and ECGA-OEM in a total of 32 test cases is given. Figure 9 shows the numbers of participants with ECGA-OEM superior, equal, and inferior, respectively. The horizontal axis is the set of different test cases, and the vertical axis is the number of matching systems. In the vast majority of test cases, the number of matching systems with ECGA-OEM superior to is much higher than those ones with ECGA-OEM inferior to. Only in No. 246, No. 247, and No. 254 cases, the ranking of ECGA-OEM is relatively low (refer to also Table 4 for the specific  $f$ -measure values in each testing case).

In Table 4, the numbers from 1 to 18 in the first row are edna, AML, CroMatch, Lily, LogMap, LogMapLt, Xmap,



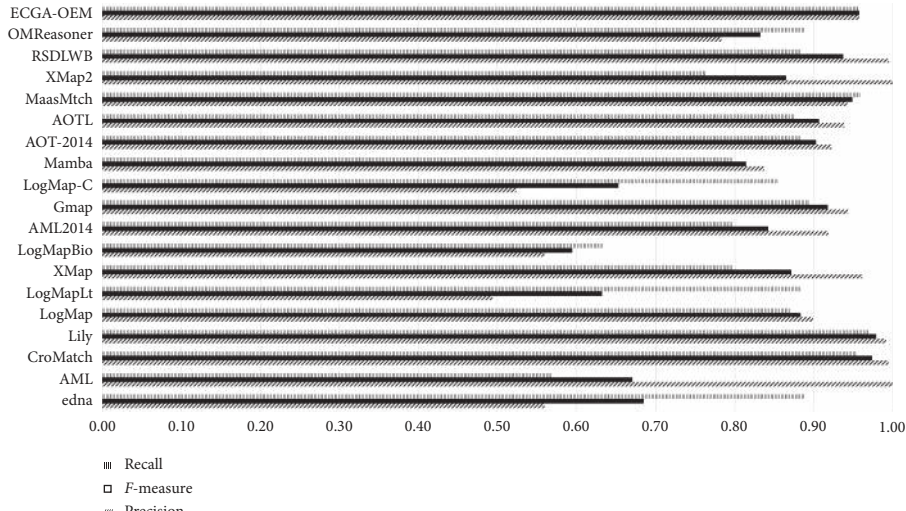


FIGURE 8: The harmonic mean of OAEI's participants and ECGA-OEM.

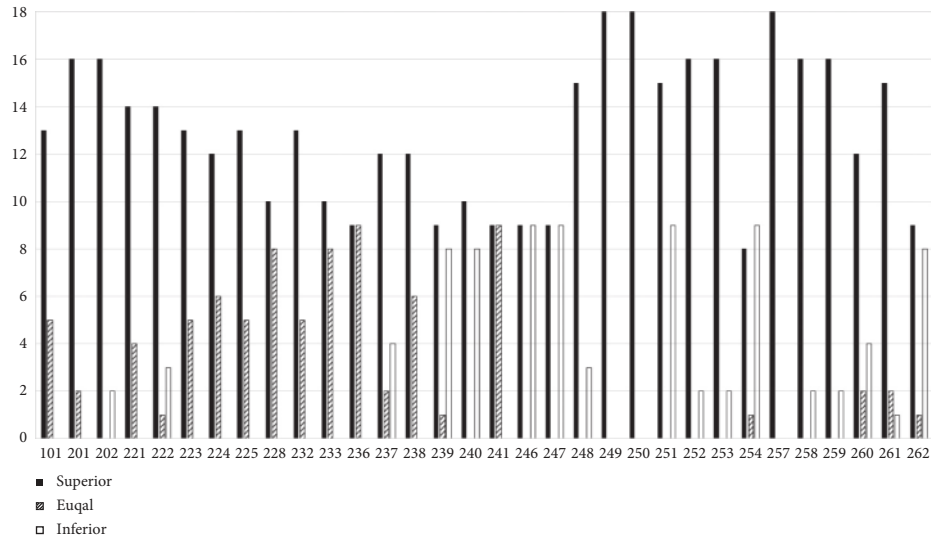


FIGURE 9: The number of participants with ECGA-OEM superior, equal, and inferior.

LogMapBio, AML-2014, Gmap, LogMap-C, Mamba, AOT-2014, AOTL, MassMatch, OMReasoner, RSDLWB, and Xmap2, respectively. ECGA-OEM is the matching system proposed by us. The value of each column represents the  $f$ -measure score of the matching system in the corresponding case. The  $f$ -measure of participants higher than that of ECGA-OEM is bold, and the equal ones are underlined. The “+,” “=,” and “-” in the last column, respectively, indicate the number of matching systems, with ECGA-OEM superior, equal, and inferior.

**5.4. Comparison among GA, CGA, and ECGA.** To verify the performance of linkage learning, we compare ECGA with GA and CGA. The detailed  $f$ -measure and runtime of the three competitors are, respectively, shown in Tables 5 and 6. All the GA, CGA, and ECGA's results are the mean value of 30 independent runs. It can be seen that the replacement of

crossover and mutation operators (GA) with probability vector (CGA) improves the  $f$ -measure and significantly reduces the runtime. The average  $f$ -measure is slightly improved, while the average runtime is reduced from 31.975 seconds to 3.540 seconds; i.e., it largely improves the algorithm's efficiency with only takes about 1/10 of runtime. Except testing cases No. 301 and No. 304, CGA is more stable than GA in terms of both  $f$ -measure and runtime since their standard deviation is smaller. Testing cases No. 301 and No. 304 are the representatives of real-world cases with unique heterogeneity, which make the  $f$ -measure produced by CGA decreased slightly. Linkage learning, the technique applied in ECGA, further increased the score of  $f$ -measure and made decrement in runtime with average 1.749 seconds based CGA. A smaller standard deviation than CGA was obtained by ECGA, which certified the strong stability of ECGA. It is worth to be noticed that the  $f$ -measure score of ECGA in testing case No. 301 and No. 304 is almost the same as that of GA (only 0.003 score lower in testing case No.

TABLE 4: Comparison on  $F$ -measure harmonic mean on each testing case among OAEI participants and ECGA-OEM.

Case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	ECGA-OEM	+/-
101	0.97	0.96	1	1	1	0.93	0	0.98	0.73	0.87	0.78	0	1	1	0.95	0.71	0.97	0.52	1	13/5/0
201	0.95	0.94	0.93	0.61	0.89	0.82	0.68	0.95	0.62	0.77	0.62	0.44	1	1	0.84	0.62	0.77	0.47	1	16/2/0
202	0.81	0.84	0.92	0.81	0.86	0.82	0.64	0.87	0.62	0.76	0.62	0.42	0.98	0.99	0.85	0.62	0.76	0.48	0.96	16/0/2
221	0.97	0.95	1	1	0.98	0.93	0.99	0.98	0.73	0.87	0.78	0.51	1	1	0.94	0.72	0.97	0.53	1	14/4/0
222	0.97	0.95	1	0.8	0.98	0.94	0.77	0.99	0	0.85	0.77	0.5	1	1	0	0.72	0.78	0	0.99	14/1/3
223	0.97	0.96	1	1	1	0.93	0.99	0.98	0.73	0.86	0.78	0.51	1	1	0.94	0.72	0.97	0.53	1	13/5/0
224	0.96	0.95	1	1	1	0.93	0.97	0.98	0.92	0.87	1	0.51	1	1	0.94	0.9	0.97	0.53	1	12/6/0
225	0.97	0.96	1	1	1	0.93	0.99	0.98	0.73	0.87	0.78	0.51	1	1	0.95	0.72	0.97	0.52	1	13/5/0
228	1	1	0.93	1	1	0.84	0.97	0.96	0.5	0.88	0.55	1	1	1	0.92	0.48	1	0.8	1	10/8/0
232	0.97	0.96	1	1	0.98	0.93	0.97	0.98	0.93	0.87	1	0.51	1	1	0.94	0.9	0.97	0.53	1	13/5/0
233	1	1	0.93	1	1	0.84	0.97	0.96	0.5	0.88	0.55	1	1	1	0.92	0.48	1	0.8	1	10/8/0
236	1	1	0.93	1	1	0.84	0.97	0.96	0.86	0.88	1	1	1	1	0.92	0.8	1	0.8	1	9/9/0
237	0.96	0.95	1	0.8	1	0.94	0.77	0.99	0	0.85	0.99	0.5	1	1	0	0.91	0.78	0	0.99	12/2/4
238	0.97	0.95	1	1	1	0.93	0.99	0.98	0.93	0.87	1	0.51	1	1	0.95	0.9	0.97	0.52	1	12/6/0
239	1	1	0.93	1	1	0.84	0.97	0.96	0.5	0.88	0.55	1	1	1	0.92	0.48	1	0.8	0.97	9/1/8
240	1	1	0.93	1	1	0.84	0.97	0.96	0.5	0.88	0.55	1	1	1	0.92	0.48	1	0.8	0.99	10/0/8
241	1	1	0.93	1	1	0.84	0.97	0.96	0.86	0.88	1	1	1	1	0.92	0.8	1	0.8	1	9/9/0
246	1	1	0.93	1	1	0.84	0.97	0.96	0.86	0.88	1	1	1	1	0.92	0.8	1	0.8	0.98	9/0/9
247	1	1	0.93	1	1	0.84	0.97	0.96	0.86	0.88	1	1	1	1	0.92	0.8	1	0.8	0.99	9/0/9
248	0.83	0.84	0.93	0.81	0.86	0.82	0.84	0.87	0.63	0.76	0.62	0.45	0.98	0.97	0.8	0.62	0.77	0.43	0.9	15/0/3
249	0.81	0.8	0.87	0.65	0.89	0.82	0.81	0.87	0.81	0.76	0.79	0.42	0.95	0.95	0.82	0.8	0.82	0.48	0.96	18/0/0
250	0.81	0.83	0.85	0.88	0.88	0.72	0.87	0.84	0.42	0.76	0.43	0.9	0.94	0.98	0.81	0.4	0.88	0.69	1	18/0/0
251	0.83	0.82	0.94	0.6	0.87	0.83	0.59	0.87	0	0.75	0.61	0.43	0.93	0.97	0	0.61	0.59	0	0.91	15/0/3
252	0.82	0.83	0	0.86	0.89	0.82	0.78	0.87	0.61	0.75	0.62	0.45	0.97	0.95	0.83	0.62	0.76	0.49	0.92	16/0/2
253	0.83	0.83	0	0.79	0.87	0.82	0.79	0.87	0.81	0.76	0.79	0.42	0.97	0.94	0.86	0.79	0.81	0.47	0.9	16/0/2
254	0.83	0.85	0	0.88	0.88	0.72	0.89	0.84	0.42	0.76	0.43	0.92	0.94	0.98	0.81	0.4	0.88	0	0.83	8/1/9
257	0.84	0.85	0	0.88	0.88	0.72	0.87	0.84	0.74	0.76	0.79	0.92	0.9	0.95	0.88	0.72	0.88	0.74	1	18/0/0
258	0.81	0.83	0	0.6	0.88	0.83	0.58	0.87	0.68	0.73	0.79	0.45	0.93	0.95	0	0.8	0.66	0	0.91	16/0/2
259	0.83	0.85	0	0.8	0.89	0.82	0.83	0.87	0.8	0.75	0.79	0.46	0.96	0.94	0.83	0.8	0.81	0	0.92	16/0/2
260	0.81	0.82	0	0.88	0.88	0.74	0.87	0.85	0	0.79	0.43	0.9	0.93	0.95	0	0.41	0.9	0	0.88	12/2/4
261	0.8	0.85	0	0.88	0.88	0.72	0.87	0.84	0.42	0.76	0.43	0.9	0.89	0.89	0.83	0.4	0.88	0	0.89	15/2/1
262	0.81	0.83	0	0.88	0.88	0.72	0.85	0.84	0.74	0.76	0.79	0.92	0.94	0.94	0.81	0.7	0.88	0.69	0.83	9/1/8

TABLE 5: Comparison of  $F$ -measure and its standard deviation.

Testing case	GA		CGA		ECGA	
	$f$ -Measure	St. dev	$f$ -Measure	St. dev	$f$ -Measure	St. dev
101	0.996	0.004	1.000	0.000	1.000	0.000
201	0.999	0.003	1.000	0.000	1.000	0.000
221	0.996	0.006	1.000	0.000	1.000	0.000
223	0.994	0.007	0.998	0.004	1.000	0.002
224	0.997	0.005	1.000	0.000	1.000	0.000
233	0.996	0.007	1.000	0.000	1.000	0.000
236	0.997	0.007	1.000	0.000	1.000	0.000
240	0.978	0.020	0.994	0.012	1.000	0.000
241	0.994	0.010	1.000	0.000	1.000	0.000
250	0.995	0.009	1.000	0.000	1.000	0.000
257	0.993	0.010	1.000	0.000	1.000	0.000
301	0.606	0.003	0.586	0.025	0.603	0.007
304	0.665	0.004	0.650	0.033	0.672	0.007
Average	0.939	0.007	0.941	0.006	0.944	0.001

TABLE 6: Comparison of runtime and its standard deviation.

Testing case	GA		CGA		ECGA	
	Runtime (s)	St. dev	Runtime (s)	St. dev	Runtime (s)	St. dev
101	44.296	4.447	4.146	0.255	2.016	0.047
201	43.463	6.195	4.144	0.279	1.976	0.042
221	43.204	5.365	4.115	0.223	2.038	0.041
223	51.146	6.640	6.947	0.208	3.226	0.255
224	44.624	6.106	4.143	0.277	2.053	0.058
233	12.686	3.080	1.276	0.078	0.734	0.022
236	13.463	2.797	1.244	0.054	0.720	0.030
240	22.740	4.950	3.294	0.028	1.771	0.145
241	13.474	3.804	1.236	0.076	0.728	0.024
250	14.256	2.576	1.235	0.093	0.624	0.026
257	14.507	2.765	1.264	0.082	0.611	0.028
301	45.056	4.253	6.306	0.880	3.353	0.086
304	52.759	6.400	6.673	0.705	2.885	0.067
Average	31.975	4.568	3.540	0.249	1.749	0.067

301 and this minor difference could be ignored), which embodied the ability of linkage learning to overcome the CGA's shortcoming.

## 6. Conclusions

Ontology matching can effectively solve the problem of data heterogeneity by discovering correspondence between two ontologies' entities. Compact encoding mechanism shows high efficiency in ontology matching, especially in ontology entity matching. Linkage learning, which is employed in ECGA-OEM proposed in this paper, can produce qualified alignment of ontology matching. The experiment results have shown that our approach outperforms the participants of OAEI in terms of f-measure, recall, and precision. The comparison on terms of f-measure and runtime with GA, CGA, and ECGA shows that ECGA-OEM is able to greatly reduce the runtime consumption while maintaining the alignment's quality.

In the future, we would like to apply the linkage learning in other compact SIAs for better results. In addition, matching large-scale ontologies, such as anatomy and large biomedical track in OAEI, are an open challenge in the domain of ontology matching. We are interested in using the improved ECGA to match these large-scale ontologies.

## Data Availability

The data used to support the findings of this study are available upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province (no. 2020J01875) and the National Natural Science Foundation of China (nos. 61773415, 61801527, and 61103143).

## References

- [1] X. Xue, J. Liu, P.-W. Tsai, X. Zhan, and A. Ren, "Optimizing ontology alignment by using compact genetic algorithm," in *Proceedings of the 2015 11th International Conference on Computational Intelligence and Security (CIS)*, pp. 231–234, IEEE, Shenzhen, China, December 2015.
- [2] X. Xue, "A compact firefly algorithm for matching biomedical ontologies," *Knowledge and Information Systems*, vol. 62, no. 11, pp. 1–17, 2020.
- [3] J. M. V. Naya, M. M. Romero, J. P. Loureiro, C. R. Munteanu, and A. Pazos Sierra, "Improving ontology alignment through genetic algorithms," in *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*, pp. 240–259, IGI Global, Hershey, PA, USA, 2010.
- [4] J. Bock and H. Jan, "Discrete particle swarm optimisation for ontology alignment," *Information Sciences*, vol. 192, pp. 152–173, 2012.
- [5] He Yao, X. Xue, and S. Zhang, "Using artificial bee colony algorithm for optimizing ontology alignment," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 4, pp. 766–773, 2017.
- [6] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, p. 333, 2020.
- [7] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.
- [8] N. Alboukaey and A. Joukhadar, "Ontology matching as regression problem," *Journal of Digital Information Management*, vol. 16, no. 1, 2018.
- [9] M. Ali Khoudja, M. Fareh, and H. Bouarfa, "Ontology matching using neural networks: survey and analysis," in *Proceedings of the 2018 International Conference on Applied Smart Systems (ICASS)*, pp. 1–6, IEEE, Medea, Algeria, November 2018.
- [10] M. T. Dhoub, C. F. Zucker, and A. G. B. Tettamanzi, "An ontology alignment approach combining word embedding and the radius measure," in *Proceedings of the 15th International Conference on Semantic Systems*, pp. 191–197, Springer, Karlsruhe, Germany, September 2019.
- [11] A. Ali, H. Mcheick, K. Ahmad, and W. Dhifli, "Context-aware instance matching through graph embedding in lexical

- semantic space,” *Knowledge-Based Systems*, vol. 186, Article ID 104925, 2019.
- [12] F. Ali, K.-S. Kwak, and Y.-G. Kim, “Opinion mining based on fuzzy domain ontology and support vector machine: a proposal to automate online review classification,” *Applied Soft Computing*, vol. 47, pp. 235–250, 2016.
  - [13] X. Xue and J.-S. Pan, “A segment-based approach for large-scale ontology matching,” *Knowledge and Information Systems*, vol. 52, no. 2, pp. 467–484, 2017.
  - [14] S. Amrouch, S. Mostefai, and M. Fahad, “Decision trees in automatic ontology matching,” *International Journal of Metadata, Semantics and Ontologies*, vol. 11, no. 3, pp. 180–190, 2016.
  - [15] T. B. Araújo, C. E. Santos Pires, T. Pereira Da Nóbrega, and D. C. Nascimento, “A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches,” *Knowledge-Based Systems*, vol. 111, pp. 17–26, 2016.
  - [16] M. B. Amin, W. A. Khan, S. Hussain et al., “Evaluating large-scale biomedical ontology matching over parallel platforms,” *Iete Technical Review*, vol. 33, no. 4, pp. 415–427, 2016.
  - [17] X. Xue and J. Liu, “Collaborative ontology matching based on compact interactive evolutionary algorithm,” *Knowledge-Based Systems*, vol. 137, pp. 94–103, 2017.
  - [18] X. Xue and J.-S. Pan, “A compact co-evolutionary algorithm for sensor ontology meta-matching,” *Knowledge and Information Systems*, vol. 56, no. 2, pp. 335–353, 2018.
  - [19] X. Xue, J. Chen, J. Chen, and D. Chen, “Using compact coevolutionary algorithm for matching biomedical ontologies,” *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 2309587, , 2018.
  - [20] X. Xue and J. Chen, “A compact co-firefly algorithm for matching ontologies,” in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2629–2632, IEEE, Xiamen, China, December 2019.
  - [21] X. Xue and J. Chen, “Using compact evolutionary tabu search algorithm for matching sensor ontologies,” *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.
  - [22] S.-C. Chu, X. Xue, J.-S. Pan, and X. Wu, “Optimizing ontology alignment in vector space,” *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.
  - [23] X. Xue and J. Chen, “Optimizing sensor ontology alignment through compact co-firefly algorithm,” *Sensors*, vol. 20, no. 7, p. 2056, 2020.
  - [24] Y. Wang, H. Yao, L. Wan et al., “Optimizing hydrography ontology alignment through compact particle swarm optimization algorithm,” in *Proceedings of the International Conference on Swarm Intelligence*, pp. 151–162, Springer, Chiang Mai, Thailand, July 2020.
  - [25] X. Xue and P.-W. Tsai, “Matching biomedical ontologies with compact evolutionary algorithm,” in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–10, Springer, Singapore, Singapore, May 2020.
  - [26] X. Xue, H. Yang, J. Zhang, J. Zhang, and D. Chen, “An automatic biomedical ontology meta-matching technique,” *Journal of Network Intelligence*, vol. 4, no. 3, pp. 109–113, 2019.
  - [27] X. Xue and J.-S. Pan, “An overview on evolutionary algorithm based ontology matching,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, pp. 75–88, 2018.
  - [28] J. Martinez-Gil, E. Alba, and J. F. Aldana-Montes, “Optimizing ontology alignments by using genetic algorithms,” in *Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web*, Karlsruhe, Germany, October 2008.
  - [29] J. Martinez-Gil and J. F. Aldana-Montes, “Evaluation of two heuristic approaches to solve the ontology meta-matching problem,” *Knowledge and Information Systems*, vol. 26, no. 2, pp. 225–247, 2011.
  - [30] G. Alexandru-Lucian and I. Adrian, “Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment,” in *Proceedings of the 9th RoEduNet IEEE International Conference*, pp. 118–122, IEEE, Sibiu, Romania, June 2010.
  - [31] G. Acampora, V. Loia, and A. Vitiello, “Enhancing ontology alignment through a memetic aggregation of similarity measures,” *Information Sciences*, vol. 250, pp. 1–20, 2013.
  - [32] X. Xue, Y. Wang, and A. Ren, “Optimizing ontology alignment through memetic algorithm based on partial reference alignment,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3213–3222, 2014.
  - [33] X. Xue and Y. Wang, “Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio,” *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.
  - [34] J. Wang, Z. Ding, and C. Jiang, “Gaom: genetic algorithm based ontology matching,” in *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC’06)*, pp. 617–620, IEEE, Guangzhou, China, December 2006.
  - [35] G. Acampora, V. Loia, S. Salerno, and A. Vitiello, “A hybrid evolutionary approach for solving the ontology alignment problem,” *International Journal of Intelligent Systems*, vol. 27, no. 3, pp. 189–216, 2012.
  - [36] A. Alves, K. Revoredo, and F. Baião, “Ontology alignment based on instances using hybrid genetic algorithm,” in *Proceedings of the 7th International Conference on Ontology Matching-Volume 946*, pp. 242–243, November 2012.
  - [37] X. Xue and Y. Wang, “Using memetic algorithm for instance coreference resolution,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2015.
  - [38] W. Jan Heeringa, *Measuring dialect pronunciation differences using levenshtein distance*, Ph.D thesis, University Library Groningen, Groningen, Netherlands, 2004.
  - [39] G. Stoilos, G. Stamou, and S. Kollias, “A string metric for ontology alignment,” in *Proceedings of the International Semantic Web Conference*, pp. 624–637, Springer, Galway, Ireland, November 2005.
  - [40] G. A. Miller, “WordNet,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
  - [41] D. Cai, Y. Wang, M. Ye, X. Xue, and H. Liu, “An orthogonal evolutionary algorithm with learning automata for multi-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3306–3319, 2015.
  - [42] G. R. Harik, F. G. Lobo, and K. Sastry, *Linkage Learning via Probabilistic Modeling in the ECGA*, Springer, Berlin, Germany, 1999.
  - [43] X. Xue and X. Yao, “Interactive ontology matching based on partial reference alignment,” *Applied Soft Computing*, vol. 72, pp. 355–370, 2018.