

Research Article

Using Compact Coevolutionary Algorithm for Matching Biomedical Ontologies

Xingsi Xue ¹, Jie Chen,¹ Junfeng Chen ² and Dongxu Chen³

¹College of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China

²College of IOT Engineering, Hohai University, Nanjing 213022, China

³Fujian Medical University Union Hospital, Fuzhou 350001, China

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

Received 8 May 2018; Accepted 30 August 2018; Published 8 October 2018

Academic Editor: Carmen De Maio

Copyright © 2018 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.

Over the recent years, ontologies are widely used in various domains such as medical records annotation, medical knowledge representation and sharing, clinical guideline management, and medical decision-making. To implement the cooperation between intelligent applications based on biomedical ontologies, it is crucial to establish correspondences between the heterogeneous biomedical concepts in different ontologies, which is so-called biomedical ontology matching. Although Evolutionary algorithms (EAs) are one of the state-of-the-art methodologies to match the heterogeneous ontologies, huge memory consumption, long runtime, and the bias improvement of the solutions hamper them from efficiently matching biomedical ontologies. To overcome these shortcomings, we propose a compact CoEvolutionary Algorithm to efficiently match the biomedical ontologies. Particularly, a compact EA with local search strategy is able to save the memory consumption and runtime, and three subswarms with different optimal objectives can help one another to avoid the solution's bias improvement. In the experiment, two famous testing cases provided by Ontology Alignment Evaluation Initiative (OAEI 2017), i.e. anatomy track and large biomed track, are utilized to test our approach's performance. The experimental results show the effectiveness of our proposal.

1. Introduction

Ontologies provide a shared and common vocabulary for representing a domain of knowledge [1]. Over the recent years, ontologies are widely used in various domains such as medical records annotation [2], medical knowledge representation and sharing, clinical guidelines management [3], and medical decision-making [4]. However, most biomedical ontologies are developed independently by different experts who might define one entity with different names or in different ways, causing the problem of ontology heterogeneity. For example, to describe the muscles that surround and power the human heart, the National Cancer Institute's thesaurus and ontology (NCI) [5] use the name "Myocardium," whereas the Foundation Model of Anatomy (FMA) [6] uses "Cardiac Muscle Tissue." To implement the cooperation between intelligent applications based on biomedical ontologies, it is crucial to establish correspondences

between the heterogeneous biomedical concepts in different ontologies, which is so-called biomedical ontology matching.

Recently, Evolutionary Algorithms (EAs) are one of the state-of-the-art methodologies to match the heterogeneous ontologies [7]. However, huge memory consumption, long runtime, and the bias improvement of the solutions hamper EA-based ontology matching techniques from efficiently matching biomedical ontologies. Thus, besides the quality of alignments, main memory consumption and runtime needed by the ontology matcher are of prime importance when matching the biomedical ontologies. In this paper, we propose to use the compact EA [8], which utilizes a probabilistic representation of the population, to save the memory consumption of classic EA. Then, we introduce the local search strategy into its evolving process to balance the exploration and exploitation and reduce the runtime needed. On this basis, we further propose a compact Coevolutionary Algorithm, which utilizes three subswarms

with different objectives to help one another to avoid the solution's bias improvement caused by traditional metric f-measure [9].

The rest of the paper is organized as follows: Section 2 describes the related works; Section 3 gives some basic concepts of ontology, ontology alignment, and the similarity measures; Section 4 presents the optimal model problem and the details of the compact Coevolutionary Algorithm for matching biomedical ontologies; Section 5 gives the experimental results and relevant analysis; finally, Section 6 draws the conclusions.

2. Related Work

2.1. Evolutionary Algorithm-Based Ontology Matching Technique. Due to the complex and time-consuming nature of the ontology matching process, EA-based methods could present a good methodology for obtaining ontology alignments and indeed have already been applied to solve the ontology alignment problem by reaching acceptable results [10]. Different from other EA based approaches [11–13] which models the ontology alignment process as a meta-matching problem, i.e. how to determine the best appropriate weight configuration in ontology matching process in order to obtain a satisfactory alignment, in this work, ontology matching problem is considered as a global entity matching problem. Genetic Algorithm-Based Ontology Matching (GAOM) [14] is the representative system, which utilized Genetic Algorithm (GA) to determine the optimal ontology alignment. Particularly, GAOM utilizes the chromosomes to describe the potential alignments between two ontologies and utilizes GAs to determine the optimal solution. Besides, MapPSO and MapEVO [15] which exploited the Particle Swarm Optimization Algorithm (PSO) [16] and Evolutionary Programming (EP) [17], respectively, also adopted this idea. Acampora et al. [18] designed a Memetic Algorithm (MA) which introduced a local search process to improve the performance of EA. More recently, Xue et al. [19, 20], respectively, used the compact EA and compact Population-Based Incremental Learning Algorithm (PBIL) to save the memory consumption without sacrificing the solution's quality. Compact EA and compact PBIL represented the population as a probability vector (PV) over the set of solutions and are operationally equivalent to the order-one behaviour of the simple EA with uniform crossover. In this way, a much smaller number of solutions must be stored in the memory, thus significantly reducing the memory consumption.

2.2. Coevolutionary Algorithm. The Coevolutionary Algorithm [21] makes multiple swarms simultaneously evolve and communicate with one another to improve the search performance. Currently, distributed coevolution is the most popular coevolving process, which shares the search information among multiple swarms through the population migration strategy. During the searching process, different swarms have evolving strategies and configurations. Tan et al. [22] proposed to decompose the problem's solution

vector into multiple swarms to evolve simultaneously. Mu and Liu [23] presented an *M*-elite Coevolutionary Algorithm that applied different elite strategies in the coevolving process. The elite centered swarm has the highest priority, and other swarms implemented the cooperative coevolving process. In [24], a parallel evolving mechanism was designed by dividing the population into three swarms that evolved independently. However, all the swarms use the same evolving strategy, and the swarm's evolving process swarm was relatively independent, which decreased the algorithm's exploration and exploitation ability. More recently, Wang et al. [25] proposed a two-elite strategy which makes use of the differences between two elites to guide the whole evolving process.

Different from all the techniques mentioned above, in this work, we propose a compact coevolutionary Algorithm to match the biomedical ontologies, which combines the advantages of the compact EA and coEvolutionary Algorithm to save the memory consumption and runtime and overcome the bias improvement of solutions.

2.3. Preliminaries

2.3.1. Ontology, Ontology Alignment, and Ontology Matching Process. In this work, an ontology is defined as a quadruple $O = (C, P, I, A)$, where

- (i) C is the class set, i.e., the set of concepts that populate the domain of interest,
- (ii) P is the property set, i.e., the set of relations between the concepts of domain,
- (iii) I is the instance set, i.e., the set of objects in the real world representing the instances of a concept, and
- (iv) A is the axiom set, i.e., the statements that say what is true about the modeled domain.

An alignment A between two ontologies O_1 and O_2 is defined as a set of correspondences, and each correspondence is a triple (e_1, e_2, n) , where e_1 and e_2 are the entities in O_1 and O_2 , respectively, and $n \in [0, 1]$ is a confidence value holding for the correspondence between them. In this work, the relation existing between two ontology entities is the equivalence ($=$). The ontology matching process can be defined as a function $\theta(O_1, O_2, p, r)$ [26], where p is the parameter set and r is the resource set. Ontology matching process returns a new alignment A_N between ontologies O_1 and O_2 .

2.3.2. Concept Similarity Measure. Concept similarity measure is the foundation of biomedical ontology matching [27]. In this work, we utilize an asymmetrical concept similarity measure to calculate the biomedical concepts' similarity values. First, for each biomedical concept, we construct a profile for it by collecting the label, comment, and property information such as label, domain, and range, from itself and all its direct descendants. Then, the similarity of two biomedical concepts c_1 and c_2 is measured based on the similarity of their profiles p_1 and p_2 , which can be calculated by the following two asymmetrical measures:

$$\begin{aligned} sim_1(p_1, p_2) &= \frac{|p_1 \cap p_2|}{p_1}, \\ sim_2(p_1, p_2) &= \frac{|p_1 \cap p_2|}{p_2}, \end{aligned} \quad (1)$$

where $|p_1|$ and $|p_2|$ are the cardinalities of the profile p_1 and p_2 , respectively, $|p_1 \cap p_2|$ is the number of identical elements in p_1 and p_2 . The similarity value of e_1 and e_2 is equal to $(sim_1(p_1, p_2) + sim_2(p_1, p_2))/2$ when $|sim_1(p_1, p_2) - sim_2(p_1, p_2)| \leq \delta$, and otherwise, 0.

In this work, δ is the threshold to measure the extent of the semantic equivalence between $sim_1(p_1, p_2)$ and $sim_2(p_1, p_2)$. When the similarity value between two profile elements is above the threshold, they are identified as semantically similar. Generally, δ should be set relatively small to reflect $sim_1(e_1, e_2)$ and $sim_2(e_1, e_2)$ have little difference when the entity e_1 and e_2 are semantically equivalent. However, if δ is too small, we would miss many semantically equivalent terms. Therefore, the suggested domain of δ is $[0.01, 0.10]$. In this work, to obtain a suitable, we conducted a pre-experiment on the benchmark by varying the value of δ in its suggested domain, and found the semantic equivalence performs well when δ is assigned to 0.06.

Moreover, the similarity value of two profile elements is calculated by N-gram distance [28], which is the most performing string-based similarity measure for the biological ontology matching problem, and a linguistic measure, which calculate a synonymy-based distance through the Unified Medical Language System (UMLS) [29]. Given two words w_1 and w_2 , their similarity $sim_2(w_1, w_2)$ is equal to 1 when two words are synonymous, and otherwise, $N - \text{gram}(w_1, w_2)$.

2.4. Compact Coevolutionary Algorithm

2.4.1. Rough Alignment Evaluations. In this work, we suppose that, in the golden alignment, one concept in the ontology is matched with only one concept in the other ontologies and vice versa. Two rough alignment evaluations, i.e., *MatchCoverage* and *MatchRatio*, are utilized to measure the alignment's quality. In particular, *MatchCoverage* is utilized to approximate recall [9], which calculates the fraction of concepts which exist in at least one correspondence in the resulting alignment in comparison to the total number of concepts in the ontology. The formula of it is presented as follows:

$$MatchCoverage = \frac{|C_{O_1-Match}| + |C_{O_2-Match}|}{|C_{O_1}| + |C_{O_2}|} \epsilon [0, 1], \quad (2)$$

where

- (i) $C_{O_1-Match}$ and $C_{O_2-Match}$ are the matched concept sets of ontology O_1 and O_2 , respectively; and
- (ii) C_{O_1} and C_{O_2} are the concept sets of ontology O_1 and O_2 , respectively.

And, *MatchRatio* is used to approximate precision [9], which calculates the ratio between the number of found

correspondences and the number of matched concepts. The formula of it is presented as follows:

$$MatchRatio = \frac{|C_{O_1-Match}| + |C_{O_2-Match}|}{2 \cdot |Corr_{O_1-O_2}|} \epsilon [0, 1], \quad (3)$$

where

- (i) $Corr_{O_1-O_2}$ is the correspondence set in the alignment; and
- (ii) $C_{O_1-Match}$ and $C_{O_2-Match}$ are the matched concept sets of ontology O_1 and O_2 , respectively;

In most instances, it requires considering both *MatchCoverage* and *MatchRatio* to measure the alignment's quality. By referring to the most common combining function f-measure [9], we define *MatchFmeasure* as follows:

$$MatchFmeasure = 2 \times \frac{MatchCoverage \cdot MatchRatio}{MatchCoverage + MatchRatio}. \quad (4)$$

2.4.2. The Optimal Model for Ontology Entity Matching Problem. Given two biomedical ontologies O_1 and O_2 , we take maximizing *MatchFmeasure* as the goal, and the optimal model for ontology entity matching problem can be defined as follows:

$$\begin{aligned} \max \quad & MatchFmeasure(X), \\ \text{s.t.} \quad & X = \left(x_1, x_2, \dots, x_{|O_1|}, x_{|O_1|+1} \right)^T, \\ & x_i = 1, 2, \dots, x_{|O_2|}, \\ & x_{|O_1|+1} \in [0, 1], \end{aligned} \quad (6)$$

where the decision variable X represents an alignment between O_1 and O_2 , x_i represents the i th correspondence between i th concept in O_1 and x_i th concept in O_2 , $|O_1|$ and $|O_2|$ are the cardinalities of the concept set in O_1 and O_2 , respectively, and $x_{|O_1|+1} \in [0, 1]$ is the threshold to filter the final alignment.

One of the shortcomings of *MatchFmeasure* is that the improvement of it does not say anything about whether both *MatchCoverage* and *MatchRatio* are simultaneously improved or not. In other words, no matter how large a measured improvement in *MatchFmeasure* is, it can still be extremely dependent on the improvement on one of the individual metrics [30]. To overcome this bias improvement, we propose a compact coevolutionary Algorithm, which has three PVs that characterize subswarms that aim at maximizing *MatchCoverage*, *MatchRatio*, and *MatchFmeasure*, respectively. Through the cooperation of three PVs, we dedicate to ensure the simultaneous improvement on *MatchCoverage* and *MatchRatio* during the evolving process.

2.4.3. Compact Evolutionary Algorithm. Model-based optimization using probabilistic modeling of the search space is

one of the areas where research on Compact Evolutionary Algorithm (CEA) has considerably advanced in recent years. In each generation, CEA updates the probability vector (PV), which is a probabilistic model describing the univariate statistics of the best solutions and then uses it to generate new candidate solutions. By employing the PV, instead of a population of solutions, to simulate the behavior of classic EA, a much smaller number of individuals is needed to be stored in the memory. Thus, CEA can significantly reduce the memory consumption [31]. In order to further improve CEA performance, we introduce the local search strategy into CEA's evolving process. This marriage between global search and local search is helpful in reducing the possibility of the premature convergence and increasing the convergence speed.

In the next, three main components of CEA, i.e., chromosome-encoding mechanism, probability vector, and local search strategy are, respectively, presented.

- (1) *Chromosome-Encoding Mechanism*: in this work, the genes are encoded through the binary coding mechanism and can be divided into two parts. The first part stands for the correspondences in the alignment, and the other one stands for a threshold. Given the total number n_1 and n_2 of two biomedical concepts in ontologies, the first part of a chromosome (or PV) consists of n_1 gene segments, and the binary code length (BCL) of each gene segment is equal to $\log_2(n_2) + 0.5$, which ensures each gene segment could present any target ontology class's index, while the second part of a chromosome (or PV) has only one gene segment, whose BCL is equal to $\log_2(1/numAccuracy) + 0.5$, which can ensure this gene segment could present any threshold value under the numerical accuracy $numAccuracy$. Thus, the total length of the chromosome (or PV) is equal to $n_1 \times \log_2(n_2) + 0.5 + \log_2(1/numAccuracy) + 0.5$.

Given a gene segment $geneSeg = \{geneBit_1, geneBit_2, \dots, geneBit_n\}$, where $geneBit_i$ is the i th gene bit value of the gene segment, we decode to obtain a decimal number whose value is equal to $\sum_{i=1}^n 2^{geneBit_i}$. In particular, with respect to the first part decoding results, the decimal numbers obtained represent the indexes of the target classes, where 0 means the source instance is not mapped to any target ontology's class. With regard to the second part of decoding result, the decimal number obtained should multiply the threshold's numerical accuracy. Last but not least, if a decimal number d obtained is larger than u , we will replace it with u/d .

- (2) *Probability Vector*: in general, CEA aims at generating a PV which represents a population of high evaluation solutions, and its operations take place directly on the PV. In this work, the number of elements in PV is equal to the number of individual's gene bits and each element's value is in $[0,1]$, and here is an example on how to use PV $(0.5, 0.9, 0.3, 0.8)^T$ to generate a new solution. First, generate four random numbers, such as 0.6, 0.5, 0.8, and 0.9. Then, compare the numbers with the elements in PV accordingly to determine the

```

(1)  $ind_{neighbor} = ind_{elite}.copy()$ ;
(2) generate  $i = round(rand(0; len))$ ;
(3)  $ind_{neighbor}[i] = ind_{new}[i]$ ;
(4) while( $random(0; 1) < p_c$ )
(5)    $i = i + 1$ ;
(6)   if ( $i == len$ )
(7)      $i = 0$ ;
(8)   end if
(9)    $ind_{neighbor}[i] = ind_{new}[i]$ ;
(10) end while

```

ALGORITHM 1

new generated individual's gene values. For example, since $0.6 > 0.5$, the first gene bit's value of the new solution is 0, and similarly, the remaining gene bits' values are 1, 0, and 0, respectively. In this way, the new solution we obtain is 0100. By repeating this procedure, we can obtain various individuals. In addition, if 0100 is the elite solution in the current generation, PV should be updated according to its information. Given PV's update rate, say 0.1, if the gene value of the elite is 0, the corresponding element of PV will minus 0.1, otherwise add 0.1. In this way, the updated PV is $(0.4, 1.0, 0.2, 0.7)^T$.

- (3) *Local Search Strategy*: local search process tries to improve the elite solution by searching in the neighborhood of it. In this work, we utilize a crossover operator to implement the local search process, which randomly copies a sequential fragment of ind_{new} 's genes into the corresponding positions of $ind_{neighbor}$, to generate a new solution. For the sake of clarity, given the length of the chromosome len and the crossover probability p_c , the pseudocode of the binary crossover operator is shown in Algorithm 1.

This procedure is similar with the two-point crossover where the first cut point is randomly selected from $\{1; 2; \dots; len\}$, and the second point is determined such that L consecutive genes (counted in a circular manner) are taken from ind_{new} . Since ind_{new} and ind_{elite} are both generated through the PV, most of their gene bit values are the same. Therefore, even when p_c is large, $ind_{neighbor}$ only mutates a few gene bit values of ind_{elite} . In this sense, this variation operator can be considered fairly exploitative.

2.4.4. Pseudocode of Compact Coevolutionary Algorithm. In this work, we use three PVs to represent the subswarms for maximizing *MatchRatio*, *MatchCoverage*, and *MatchFmeasure*, respectively. In particular, the PV here represents the population that consists of the solutions of its corresponding representative subproblem and this problem's neighbor subproblems. Finally, these PVs help each other in the process of determining three representative solutions, which are given in the following. Here, we mark three representative subproblems of maximizing *MatchRatio*, maximizing *MatchCoverage*, and maximizing *MatchFmeasure* with the symbols P_{mr} , P_{mc} , and P_{mf} ,

Input :

- (i) O_1 and O_2 : two biomedical ontologies;
- (ii) len : the length of PV;
- (iii) $maxGen$: maximum number of generations;
- (iv) UR : PV's update rate;
- (v) p_c : crossover probability;
- (vi) p_m : mutation probability;
- (vii) MR : mutation rate.

Output: the solution with best *MatchFmeasure*

Step 1. Initialization:

Step 1.1. Set the generation $gen = 0$;

Step 1.2. Set the neighbor subproblem of P_{mr} and P_{mc} as P_{mf} and the neighbor subproblems of P_{mf} as P_{mr} and P_{mc} .

Step 1.3. Initialize PV_{mr} , PV_{mc} , and PV_{mf} by setting all the probabilities inside as 0.5.

Step 1.4. Using PV_{mr} , PV_{mc} , and PV_{mf} to generate the elites, which are marked with symbols $elite_{mr}$, $elite_{mc}$, and $elite_{mf}$ for P_{mr} , P_{mc} , and P_{mf} , respectively.

Step 2. Evolving process:

Step 2.1. Update PV_{mr} , PV_{mc} , and PV_{mf} , respectively.

Take updating PV_{mr} for instance, the procedures of updating PV_{mc} and PV_{mf} is similar to it:

Step 2.1.1. Crossover

- (1) Generate a new individual ind_{new} through PV_{mr} ;
- (2) $[winner, loser] = compete(elite_{mr}, ind_{new})$;
- (3) if($winner == ind_{new}$)
- (4) $elite_{mr} = ind_{new}$;
- (5) for $i = 0$; $i < PV_{mr}.length$; $i++$
- (6) if($winner[i] == 1$)
- (7) $PV_{mr}[i] = PV_{mr}[i] + UR$;
- (8) if($PV_{mr}[i] > 1$)
- (9) $PV_{mr}[i] = 1$;
- (10) else
- (11) $PV_{mr}[i] = PV_{mr}[i] - UR$;
- (12) if ($PV_{mr}[i] < 0$)
- (13) $PV_{mr}[i] = 0$;

Step 2.1.2. Mutation

- (14) for($i = 0$; $i < len$; $i++$)
- (15) if($(random(0, 1) < P_m)$)
- (16) $PV_{mr}[i] = PV_{mr}[i] \times (1 - MR) + random(0, 1) \times MR$;

Step 2.1.3. Local search

- (17) Generate an individual ind_{new} through PV_{mr} ;
- (18) $ind_{neighbor} = elite_{mr}.copy()$;
- (19) Generate $i = round(random(0, len))$;
- (20) $ind_{neighbor}[i] = ind_{new}[i]$;
- (21) while($(random(0, 1) < P_c)$)
- (22) $i = i + 1$;
- (23) if($(i == ind_{neighbor}.length)$)
- (24) $i = 0$;
- (25) $ind_{neighbor}[i] = ind_{new}[i]$;
- (26) end While
- (27) $[winner, loser] = compete(elite_{mr}, ind_{neighbor})$;
- (28) if($(winner == ind_{new})$)
- (29) $elite_{mr} = ind_{new}$;

Step 2.2. Update PV_{mr} , PV_{mc} and PV_{mf} mutually.

For P_{mr} (or P_{mc}), PV_{temp} is generated by applying the p_c -based uniform crossover operator [32] on the PV_{mr} (or P_{mc}) and its neighbor subproblem's probability vector PV_{mf} . Then, generate an individual a through PV_{temp} and try to update the PV_{mr} and PV_{mf} through the competition with $elite_{mr}$ (or $elite_{mc}$) and $elite_{mf}$.

Continued.

For PV_{mf} , PV_{temp} is generated through applying the uniform crossover operator between PV_{mr} and P_{mc} , which are its neighbor subproblems' PVs. Then, generate an individual a through PV_{temp} and try to update the P_{mf} through the competition with $elite_{mf}$.

Step 3. Stopping Criteria:

- (30) if ($maxGen$ is reached)
- (31) stop and the elite with best $MatchFmeasure$;
- (32) else
- (33) $gen = gen + 1$;
- (34) go to Step 2;
- (35) end if

In the evolving process, we first update PV_{mr} , PV_{mc} , and PV_{mf} , respectively (Step 2.1), which is equivalent to the process of updating the solutions of P_{mr} , P_{mc} , and P_{mf} . Then, we update PV_{mr} , PV_{mc} , and PV_{mf} mutually (Step 2.2), which is equal to updating the solutions of P_{mr} , P_{mc} , and P_{mf} through their shared neighbor subproblems' solutions, i.e., using the information of a PV to help its neighbor PVs.

ALGORITHM 2

respectively, and three PVs for solving P_{mr} , P_{mc} , and P_{mf} with the symbols PV_{mr} , PV_{mc} , and PV_{mf} , respectively. We present the pseudocode of compact Coevolutionary Algorithm in Algorithm 2.

2.5. Experimental Results and Analysis. In this work, we exploit the Anatomy (<http://oaei.ontologymatching.org/2017/anatomy/index.html>) and Large Biomed (<http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2017/>) track to study the effectiveness of our approach, which are provided by the Ontology Alignment Evaluation Initiative (OAEI 2017) (<http://oaei.ontologymatching.org/2017>). The Anatomy track includes two ontologies (1 task), i.e., the Adult Mouse Anatomy (AMA) ontology (2,744 classes) and a part of NCI describing the human anatomy (3,304 classes). Large Biomed track (3 tasks) aims at finding alignments between FMA, SNOMED CT, and NCI, which, respectively, contains 78,989, 122,464, and 66,724 classes. Particularly, The large Biomed track is split into three matching problems: FMA-NCI, FMA-SNOMED, and SNOMED-NCI and each matching problem in these tasks involving different fragments of the input ontologies.

The Compact Coevolutionary Algorithm uses the following parameters which represent a trade-off setting obtained in an empirical way to achieve the highest average alignment quality on all exploited testing datasets:

- (i) Numerical accuracy = 0.01;
- (ii) Update rate = 0.1;
- (iii) Crossover probability = 0.6;
- (iv) Mutation probability = 0.03;
- (v) Mutation rate = 0.05;
- (vi) Maximum generation = 3000.

3. Results and Analysis

In order to compare the quality of our proposal with the participants of OAEI 2017 (<http://oaei.ontologymatching.org/>

2017/results/index.html) and Population-Based Incremental Learning Algorithm (PBIL) [20], which is a state-of-the-art compact EA-based ontology matching technique, we evaluate the obtained alignments with traditional recall, precision, and f-measure. PBIL and our approach's results in Table 1 and Table 2 are the mean values in thirty time independent executions. The symbols P , R , and F in tables stand for precision, recall, and f-measure, respectively.

As can be seen from Table 1, our approach's f-measure outperforms all the competitors, and our approach's runtime is ranked the 4th place. In Table 2, our approach's f-measure is the highest in task1, task2, and task3. For the running time, in task1 and task 2, our approach is in the 3rd place and 4th place in task3. In both tracks, our approach outperforms AML, which is the top ontology matcher and developed primarily for the biomedical ontology matching, in all tasks in terms of f-measure, and the runtime in our approach is also very close to or less than AML. The experimental results show that the cooperation among three swarms with different objectives can effectively overcome the bias improvements and improve the quality of biomedical ontology alignments.

In particular, PBIL works with one PV, but our approach utilizes three PVs to cooperate with each other during the evolving process to improve the solution's quality. As can be seen from the experimental results, although our approach takes only a little more runtime than PBIL, the qualities of our results are much better than PBIL in terms of both recall and precision, which shows that our approach can effectively overcome the bias improvement of solutions in PBIL.

4. Conclusion

In this work, in order to overcome the drawbacks in traditional E-based ontology matching techniques, we for the first time propose a compact Coevolutionary Algorithm to efficiently match the biomedical ontologies. In our approach, three PVs are utilized to characterize three subswarms that

TABLE 1: Comparison of our approach with the participants in OAEI 2017 on anatomy track.

System	R	P	F	Runtime (second)
AML	0.93	0.95	0.94	37
YAM-BIO	0.92	0.94	0.93	70
POMap	0.90	0.94	0.93	808
LogMapBio	0.89	0.88	0.89	820
XMap	0.86	0.92	0.89	37
LogMap	0.84	0.91	0.88	22
KEPLER	0.74	0.95	0.83	234
LogMapLite	0.72	0.96	0.82	19
SANOM	0.77	0.89	0.82	295
Wiki2	0.73	0.88	0.80	2204
ALIN	0.33	0.99	0.50	836
EA	0.76	0.88	0.78	22
Our approach	0.94	0.97	0.95	34

TABLE 2: Comparison of our approach with the participants in OAEI 2017 on the Large Biomed track.

System	R	P	F	Runtime (second)
<i>Task1: whole FMA and NCI ontologies</i>				
XMap	0.85	0.88	0.87	130
AML	0.87	0.84	0.86	77
YAM-BIO	0.89	0.82	0.85	279
LogMap	0.81	0.86	0.83	92
LogMapBio	0.83	0.82	0.83	1552
LogMapLite	0.82	0.67	0.74	10
Tooll	0.74	0.69	0.71	1650
PBIL	0.76	0.88	0.78	22
Our approach	0.87	0.89	0.88	72
<i>Task2: whole FMA and SNOMED ontologies</i>				
XMap	0.84	0.77	0.81	625
YAM-BIO	0.73	0.89	0.80	468
AML	0.69	0.88	0.77	177
LogMap	0.65	0.84	0.73	477
LogMapBio	0.65	0.81	0.72	2951
LogMapLite	0.21	0.85	0.34	18
Tooll	0.13	0.87	0.23	2140
PBIL	0.72	0.74	0.72	147
Our approach	0.81	0.84	0.82	183
<i>Task3: whole SNOMED and NIC ontologies</i>				
AML	0.67	0.90	0.77	312
YAM-BIO	0.70	0.83	0.76	490
LogMapBio	0.64	0.84	0.73	4728
LogMap	0.60	0.87	0.71	652
LogMapLite	0.57	0.80	0.66	22
XMap	0.55	0.82	0.66	563
Tooll	0.22	0.81	0.34	1105
PBIL	0.64	0.81	0.71	304
Our approach	0.73	0.88	0.79	326

take as objectives maximizing *MatchCoverage*, *MatchRatio*, and *MatchFmeasure*, respectively, and in each generation, PVs are first updated with CEA paradigm and then help each other to search for better solutions in the search space. In the experiment, OAEI 2017's Anatomy track and Large Biomed track are utilized to test our approach's performance, and the results show that our approach can efficiently determine better ontology alignments than state-of-the-art biomedical ontology matching techniques.

Data Availability

The data used to support the findings of this study have not been made available because of the protection of technical privacy and confidentiality.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61503082 and 61403121), Natural Science Foundation of Fujian Province (No. 2016J05145), Scientific Research Startup Foundation of Fujian University of Technology (No. GY-Z15007), Scientific Research Development Foundation of Fujian University of Technology (No. GY-Z17162), and Fujian Province Outstanding Young Scientific Researcher Training Project (No. GY-Z160149).

References

- [1] S. M. Falconer, *Cognitive support for semi-automatic ontology mapping*, Ph.D. dissertation, University of Victoria, Victoria, Canada, 2009.
- [2] H. L'opez-Fern'andez, M. Reboiro-Jato, D. Glez-Peña et al., "Bioannotate: a software platform for annotating biomedical documents with application in medical learning environments," *Computer Methods and Programs in Biomedicine*, vol. 111, no. 1, pp. 139–147, 2013.
- [3] D. Isern, D. S'anchez, and A. Moreno, "Ontology-driven execution of clinical guidelines," *Computer Methods and Programs in Biomedicine*, vol. 107, no. 2, pp. 122–139, 2012.
- [4] P. De Potter, H. Cools, K. Depraetere et al., "Semantic patient information aggregation and medicinal decision support," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 2, pp. 724–735, 2012.
- [5] J. Golbeck, G. Fragoso, F. Hartel et al., "The National Cancer Institute's thesaurus and ontology," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no. 1, pp. 75–80, 2011.
- [6] C. Rosse and J. L. Mejino Jr, "A reference ontology for biomedical informatics: the foundational model of anatomy," *Journal of Biomedical Informatics*, vol. 36, no. 6, pp. 478–500, 2003.
- [7] X. Xue and J. Liu, "Collaborative ontology matching based on compact interactive evolutionary algorithm," *Knowledge-Based Systems*, vol. 137, pp. 94–103, 2017.
- [8] S. Baluja, "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning," Department Of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep., 1994.
- [9] C. J. V. Rijsberge, *Information Retrieval*, University of Glasgow, London, UK, 1975.
- [10] 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, 2016.
- [11] J. Martinez-Gil, E. Alba, and J. Montes, "Optimizing ontology alignments by using genetic algorithms," in *Proceedings of First International Workshop on Nature based reasoning for*

- the semantic Web*, vol. 419, pp. 31–45, Karlsruhe, Germany, November 2008.
- [12] J. M. V. Naya, M. M. Romero, and J. P. Loureiro, *Improving Ontology Alignment through Genetic Algorithms*, Information Science Reference, Hershey, New York, 2010, pp. 240–259.
- [13] A.-L. Ginsca and A. Iftene, “Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment,” in *Proceedings of 9th Roedunet International Conference*, pp. 118–122, Sibiu, Romania, June 2010.
- [14] J. Wang, Z. Ding, and C. Jiang, “GAOM: genetic algorithm based ontology matching,” in *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, pp. 617–620, Guangzhou, China, December 2006.
- [15] J. Bock and J. Hettenhausen, “Discrete particle swarm optimisation for ontology alignment,” *Information Sciences*, vol. 192, pp. 152–173, 2012.
- [16] J. Kennedy, *Particle Swarm Optimization*, Springer, Boston, MA, USA, 2011.
- [17] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley, Chichester, UK, 1966.
- [18] G. Acampora, V. Loia, S. Salerno et al., “A hybrid evolutionary approach for solving the ontology alignment problem,” *International Journal of Intelligent Systems*, vol. 27, pp. 189–216, 2012.
- [19] X. Xue, J. Liu, P. Tsai et al., “Optimizing ontology alignment by using compact genetic algorithm,” in *Proceedings of 11th International Conference on Computational Intelligence and Security*, pp. 231–234, Guangzhou, China, December 2016.
- [20] X. Xue and J. Chen, “Optimizing ontology alignment through hybrid population-based incremental learning algorithm,” *Memetic Computing*, pp. 1–9, 2018, In press.
- [21] P. R. Ehrlich and P. H. Raven, “Butterflies and plants: a study in coevolution,” *Evolution*, vol. 18, no. 4, pp. 586–608, 1964.
- [22] Y. J. Y. K. C. Tan and C. K. Goh, “A distributed cooperative coevolutionary algorithm for multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527–549, 2006.
- [23] L. C. J. C. H. Mu and Y. Liu, “M-elite coevolutionary algorithm for numerical optimization,” *Journal of Software*, vol. 20, no. 11, pp. 2925–2938, 2009.
- [24] Q. Zhou and W. J. Luo, “A novel multi-population genetic algorithm for multiple-choice multidimensional knapsack problem,” in *Proceedings of 5th International Symposium on Advances in Computation and Intelligence*, Berlin, Germany, Springer-Verlag, October 2010, pp. 148–157.
- [25] X. Wang, Q. Liu, Q. Fu, and L. Zhang, “Double elite coevolutionary genetic algorithm,” *Journal of Software*, vol. 23, no. 4, pp. 765–775, 2012.
- [26] J. Euzenat and P. Valtchev, “Similarity-based ontology alignment in owlite,” in *Proceedings of 16th European Conference on Artificial Intelligence Proceeding*, pp. 333–337, Valencia, Spain, August 2004.
- [27] A. Maedche and S. Staab, “Measuring similarity between ontologies,” in *Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management*, pp. 251–263, Ischia Island, Italy, July 2002.
- [28] G. Kondrak, “N-gram similarity and distance,” in *International Symposium on String Processing and Information Retrieval*, pp. 115–126, Springer, Glasgow, UK, October 2005.
- [29] O. Bodenreider, “The unified medical language system (umls): integrating biomedical terminology,” *Nucleic Acids Research*, vol. 32, no. 90001, pp. D267–D270, 2004.
- [30] X. Xue and Y. Wang, “Optimizing ontology alignments through a memetic algorithm using both match f-measure and unanimous improvement ratio,” *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.
- [31] F. Neri, G. Iacca, and E. Mininno, *Compact Optimization, Handbook of Optimization*, Springer, Berlin, Germany, 2013.
- [32] G. Syswerda, “Uniform crossover in genetic algorithms,” in *Proceedings of Third International Conference on Genetic Algorithms*, pp. 2–9, San Francisco, USA, June 1989.