Research Article

Optimizing Ontology Alignment through Improved NSGA-II

Yikun Huang,1 Xingsi Xue,2,3,4,5 and Chao Jiang2

1Concord University College, Fujian Normal University, Fuzhou 350118, China
2College of Information Science and Engineering, Fujian University of Technology, Fuzhou, Fujian 350118, China
3Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou 350118, China
4Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin 541004, China
5Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou 350118, China

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

Received 18 May 2020; Accepted 1 June 2020; Published 19 June 2020

Guest Editor: Chi-Hua Chen

Copyright © 2020 Yikun Huang 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 past decades, a large number of complex optimization problems have been widely addressed through multiobjective evolutionary algorithms (MOEAs) [1–3]. Generally, the objectives are clashing, preventing concurrent optimization for each objective, which is a challenging and realistic issue [2]. The general approach is to obtain a set of solutions, the so-called Pareto front (PF), that do not dominate each other, and the knee solution of PF is most likely to be fitting for the decision maker (DM) without any user predilections [4, 5]. In this scenario, the problem of ontology alignment in the domain of semantic web (SW) is studied. Ontology, as the kernel technique in SW, can represent a formal definition on the domain knowledge. And, matching ontologies can help find their heterogeneous entities, which can speed up the translation discovery and integrate the knowledge [6]. However, due to the complex heterogeneity in different ontologies, it is arduous to get an excellent alignment to reach all DMs’ demand and the ontology matching process usually needs a trade-off between two objectives, i.e., the precision and recall of the obtained alignment.

To meet the diverse requirements of various DMs, a famous multiobjective EA (MOEA), i.e., nondominated sorting genetic algorithm (NSGA-II) [7–10], has been proposed to deal with the ontology matching problem which determines the nondominated solutions and outputs the knee solution of PF as the representation one. Meanwhile, a variety of hybrid optimization approaches have been introduced in the recent years to improve the accuracy and speed of convergence to true optimum solutions which
integrate EA with the local search algorithm (LS) [3, 11–13]. This combination allows gaining high diversity of population due to the high optimization ability which can augment the speed of convergence and reduce the probability of premature constringency. In this study, to efficiently determine the knee solution of PF, we propose an improved NSGA-II (iNSGA-II) which incorporates into NSGA-II’s evolutionary process the monkey king evolution algorithm (MKE) [14] as the LS algorithm. The proposed iNSGA-II is able to better converge to the real Pareto optimum solution and ameliorate the quality of the solution. The remainder of the paper is organized as follows: Section 2 demonstrates the related work. Section 3 provides definition of the ontology matching problem and detailed depiction of similarity measures and the aggregation strategy; Section 4 shows the iNSGA-II-based ontology matching technique in detail; Section 5 demonstrates the experimental results; and eventually, Section 6 delineates the conclusion.

2. Related Work

A variety of EA-based matchers with the trait of effectively tackling the ontology alignment problem have been introduced in the recent years. The first category of the EA-based matcher is for the purpose of addressing the problem of ontology metamatching aiming at optimizing the parameters in aggregating diverse ontology matchers’ alignment. Naya et al. [15] first proposed the approach by using EA where several similarity measures were combined into one to improve the quality of alignments, and its encoding mechanism is still widely utilized. After that, a memetic algorithm is proposed by Acampora et al. that incorporates the local perturbation into EA to enhance the performance of the algorithm [11]. Acampora et al. [7] and Xue et al. [10] applied the NSGA-II to optimize the alignment whose results were better than the genetic algorithm. Biniz et al. proposed a hybrid approach that combines NSGA-II and a neural network, and its results were effective. Xue et al. [3] proposed that the MatchFmeasure can be approximated to an $f$-measure for better instructing the algorithm’s optimization track without using the reference alignment.

The second category of the ontology matching technique based on EA devotes to determine the optimum alignment set of entities. Wang et al. [16] proposed the GAOM (genetic algorithm-based ontology matching) that first models a discrete optimization model for optimizing the mapping set. Chu et al. [17] presented a new metric that took into consideration the entity information in the vector space, and then an EA-based matcher was applied to enhance the alignment quality. A memetic algorithm was proposed by Xue et al., which was effective in instance coreference resolution [12]. Several approaches have been proposed which can efficiently determine the alignments using EA involving user coordination [13, 18, 19]. Xue et al. [20] proposed the CETS (compact evolutionary Tabu search algorithm) to match sensor ontologies, and Acampora et al. [21] compared the performance of five local search algorithms in the ontology matching problem and found that Tabu search results presented the best performance. The proposed iNSGA-II belongs to the first category. Because of the characteristics of exploitation and exploration of MKE, our proposal incorporates NSGA-II, with the best trade-off between precision and recall, and MKE, as a local search algorithm, to further ameliorate the results’ quality and reach DMs’ demand.

3. Preliminaries

3.1. Ontology Matching Problem. An ontology consists of classes, data type properties, and object properties, which are generally called ontology entities [22]. The purpose of ontology matching is determining the entity correspondence set, which is the so-called ontology alignment. Traditionally, an $f$-measure was often utilized to assess the quality of alignments, which is defined as follows:

$$\text{recall} = \frac{|R \cap A|}{|R|},$$
$$\text{precision} = \frac{|R \cap A|}{|A|},$$
$$f - \text{measure} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}},$$

where $R$ is the reference alignment originated from domain experts and $A$ is the alignment derived from the ontology matcher. Metrics of recall and precision present the alignments’ completeness and correctness, which usually balance each other through the reconciliation mean of them, i.e., the alleged $f$-measure. But in the real world, obtaining the reference alignments is particularly expensive, especially when handling the large-scale ontologies. In this paper, we utilize three approximate measures, i.e., MatchCover [3], MeanSimilarity [17], and $f$-measure, to roughly approximate recall, precision, and $f$-measure, respectively. The motivation of MatchCover and MeanSimilarity is to map more identical entities and to ensure the higher similarity of aligned entities, respectively. Assuming the golden alignment’s cardinality is one to one, three approximate measures are, respectively, defined as follows:

$$\text{MatchCover} = 2 \times \frac{|O_1 - \text{match}| + |O_2 - \text{match}|}{|O_1| + |O_2|},$$
$$\text{MeanSim} = \frac{\sum_{i=1}^{|A|} \text{simValue}_i}{|A|},$$
$$f - \text{measure}' = 2 \times \frac{\text{MatchCover} \times \text{MeanSim}}{\text{MatchCover} + \text{MeanSim}},$$

where $|O_1|$ and $|O_2|$ are, respectively, the cardinalities of $O_1$ and $O_2$; $|O_1 - \text{match}|$ and $|O_2 - \text{match}|$ are the number of matched entities in two ontologies, respectively; $|A|$ is the number of correspondence; $\text{simValue}_i$ is the $i$th entity correspondence’s similarity value. Eventually, we define the multiobjective optimization model for the ontology matching problem as follows:
where \( f_1(\tilde{X}) \) and \( f_2(\tilde{X}) \) are the MatchCover and Mean-Similarity of \( \tilde{X} \), respectively; \( n \) is the number of similarity measures utilized; \( x_i \) represents the corresponding weight, \( i = 1, 2, \ldots, n \); \( x_{n+1} \) is the threshold for the purpose of filtering the final alignments.

### 3.2. Similarity Measures

Typically, similarity measures can be classified into terminology-based, semantic-based, and structure-based measures.

#### 3.2.1. Terminology-Based Measures

Terminology-based measures calculate the string distance between entity identifiers, labels, and comments of two ontologies. There are various terminology-based measures, for instance, Levenshtein distance [23] and Jaro–Winkler distance [24]. In this work, the widely used terminology-based measure is employed, i.e., Levenshtein distance. The Levenshtein distance can be defined as the following equation by the given two strings \( s_1 \) and \( s_2 \):

\[
\text{Levenstein}(s_1, s_2) = \max \left( 0, \frac{\min(|s_1|, |s_2|) - d(s_1, s_2)}{\min(|s_1|, |s_2|)} \right),
\]

where \( |s_1| \) and \( |s_2| \) are, respectively, the string lengths of \( s_1 \) and \( s_2 \) and \( d(s_1, s_2) \) is the quantity of operation necessary to transform \( s_1 \) to \( s_2 \).

#### 3.2.2. Semantic-Based Measures

The similarity value determined by the semantic-based measures takes into account the semantic information. With the consideration of the entities’ identifiers, we employ the WordNet [25], i.e., an electronic vocabulary database that combines different words into a group of synonyms, to calculate the distance based on linguistic relationships, such as synonymy and hypernym. The linguistic distance, \( \text{Sim}_\text{lin}(w_1, w_2) \), is represented as follows by the given two words \( w_1 \) and \( w_2 \):

\[
\text{Sim}_\text{lin}(w_1, w_2) = \begin{cases} 
1, & \text{if two words are synonymous,} \\
0.5, & \text{if one word is the hypernym of the other,} \\
0, & \text{otherwise.}
\end{cases}
\]

Another semantic-based similarity method used is the cosine distance which is expressed by means of the magnitude and dot product from two vectors [26]. Machine learning has a wide range of application scenarios [27–30]; in this study, we employ the trained Word2Vec model (https://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/) that trains through a Google News dataset with roughly 100 billion words. Given two entity vectors \( V_1 \) and \( V_2 \), the cosine distance is defined as follows:

\[
\text{cosine distance}(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \times |V_2|}.
\]

#### 3.2.3. Structure-Based Measures

In our study, we compute the structure-based measure that says “elements are similar in two different ontologies if they are related to similar elements.” Particularly, the structure-based distance is computed by using the famous algorithm, similarity flooding (SF) [31], where an iterative fixpoint computation (see equation (7)) is used to generate correspondence between the elements of two ontologies:

\[
\delta^{i+1} = \text{norm}(\delta^i + f(\delta^i)),
\]

where \( \delta^i \) is the value of the last iteration changed in each iteration and \( f \) is a function that enables a similarity value of an element pair to increment in accordance with the similarity of its adjacent elements. For more information for the SF algorithm, please see [31].

#### 3.2.4. Aggregation Strategy

A straightforward but practical aggregation strategy is utilized in this study, as an averaged weight method to integrate all mentioned similarity measures above. The method can be described as follows:

\[
\text{simValue}(e_1, e_2) = \sum_{i=1}^{n} w_i \times \text{sim}(e_1, e_2),
\]

where \( e_1 \) and \( e_2 \) are the entities from different ontologies; \( n \) is the number of similarity approaches considered; \( w_i \) is the weight of the corresponding similarity measure; \( \text{sim}(e_1, e_2) \) is the instance of the represented similarity function above.

### 4. iNSGA-II for Optimizing Ontology Alignment

In this study, we apply the iNSGA-II to tackle the ontology matching problem. As a flexible and robust approach, NSGA-II can quickly find various nondominated solutions. To further improve the probability of true convergence and enhance the quality of the solutions, as the character of exploitation and exploration of MKE, we introduce the MKE as the local search algorithm. We depict in detail the encoding mechanism, genetic operators, and local search algorithm in the next sections.

#### 4.1. Encoding Mechanism

At the beginning of iNSGA-II, a random parent population is engendered through decimal encoding. The chromosome of an individual can be split into two parts, one representing the several weights and the other representing the threshold. Supposing \( p \) is the number of weights we need, then the several cuts could be expressed as \( c = \{c_1, c_2, \ldots, c_{p-1}\} \). The chromosome decoding is carried out by ordering \( c \) from lower to higher, and then we get \( c = \{c_1, c_2, \ldots, c_{p-1}\} \) and computing the weights as follows:
4.2. Genetic Operators. The INSGA-II utilizes the following genetic operators to generate an offspring population.

4.2.1. Selection. The selection operator’s target is to select two individuals that can be utilized in the crossover operator. In this study, two individuals are selected randomly from parent population as parent\(_1\) and parent\(_2\), respectively. Figure 1 demonstrates the encoding and decoding mechanisms of several weights.

4.2.2. Crossover. According to the selection operator, we select two chromosomes in the population as parents, i.e., parent\(_1\) and parent\(_2\). We check whether the crossover operator could be applied on the basis of the crossover probability, that is, the parameter in the algorithm. Then, two children can be engendered according to the following formula:

\[
\text{child}_i = \text{rand} \times \text{parent}_1 + (1 - \text{rand}) \times \text{parent}_2, \quad i = 1, 2,
\]

where \(\text{child}_i\) is the \(i^{th}\) generated individual and \(\text{rand}\) is a random number in the interval [0, 1].

4.2.3. Mutation. For the purpose of preventing premature convergence and assuring the diversity of population, the mutation operator is utilized. By means of the mutation probability, the newly generated individual can be produced as the following formula:

\[
\text{Ind}_{\text{new}} = \begin{cases} 
\text{rand} \times \text{Ind}_{\text{old}}, & \text{if } r < 0.5, \\
\text{Ind}_{\text{old}} + \text{rand} \times (1 - \text{Ind}_{\text{old}}), & \text{otherwise}, 
\end{cases}
\]

where \(\text{rand}\) is a random number in the interval [0, 1] and \(r\) is a random number to determine whether \(\text{Ind}_{\text{old}}\) should become bigger or not. Furthermore, the formula ensures the newly produced individual in the interval [0, 1].

4.3. Generation of New Population. For the sake of speeding up the algorithm’s convergence and ensuring the diversification of population, the generation of a new parent population chooses half of the chromosomes from the front end of the population which combines parent population and offspring population by using nondomination ranks and crowding distance. More details can be seen in [1].

4.4. Local Search Algorithm. After the new parent population is engendered, to efficiently determine the knee solutions of PF, we apply the MKE for local perturbation on each knee solution in the Pareto front when the knee solution does not change in five generations. In this study, the knee solution is the one in the PF with the best recall or precision. In particular, the knee position which is the latent location of the Pareto front presents the greatest trade-off between objectives. For each knee solution, i.e., the best recall and precision, the adopted algorithm for local perturbation in each knee solution is the second version of MKE [14]. The design of MKE is inspired by the Chinese fairy tales, Journey to the West, and the Monkey King which plays a key role in protecting the master and solving various problems in the journey. Equation (12) denotes the equation of \(\bar{X}\) and \(\bar{X}_{\text{knee,G}}\), where \(\bar{X}\) is the matrix representing the population, the \(i^{th}\) individual \(X_i\) is the \(i^{th}\) row vector of \(\bar{X}\), \(s\) is the scale of population, \(X_{\text{knee,G}}\) is the knee solution in the population, and \(\bar{X}_{\text{knee,G}}\) denotes the knee solution matrix with the \(C \times D\) row vector:

\[
\bar{X} = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_s \end{bmatrix},
\]

\[
\bar{X}_{\text{knee,G}} = \begin{bmatrix} X_{\text{knee,G}} \\ X_{\text{knee,G}} \\ \cdots \\ X_{\text{knee,G}} \end{bmatrix}_{C \times D}.
\]

The formula of updating the knee solution is defined as follows:

\[
\bar{X}_{\text{diff}} = \bar{X}_{r_1} - \bar{X}_{r_2},
\]

\[
\bar{X}_{m,k} = \bar{X}_{\text{knee,G}} + FC \times \bar{X}_{\text{diff}},
\]

\[
X_{\text{knee,G}+1} = \text{opt}\{X_{m,k}(i), X_{\text{knee,G}}\},
\]

where \(\bar{X}_{r_1} = \begin{bmatrix} X_{r_1}^1 \\ X_{r_1}^2 \\ \cdots \\ X_{r_1}^C \end{bmatrix}_{C \times D}\) and \(\bar{X}_{r_2} = \begin{bmatrix} X_{r_2}^1 \\ X_{r_2}^2 \\ \cdots \\ X_{r_2}^C \end{bmatrix}_{C \times D}\), in which \(C\) and \(D\) are, respectively, the scales of differential individuals.
and the cardinality of dimensions; $\bar{X}_{r_1}$ and $\bar{X}_{r_2}$ are randomly produced by choosing the $C \times D$ row vector from the population $\bar{X}$; $\bar{X}_{\text{diff}}$ is the difference matrix engendered by the disparity of two random matrices $\bar{X}_{r_1}$ and $\bar{X}_{r_2}$; $\text{FC}$ is the fluctuation coefficient of the difference matrices; $X_{mk}(i)$ is the $i$th row vector in $\bar{X}_{mk}$, $i = 1, 2, \ldots, C$; $X_{\knee,G+1}$ is the optimum row vector in $\bar{X}_{mk}(i)$ and $X_{\knee,G+1}$, i.e., the individual with the highest fitness. The flow of determining the alignments is shown in Figure 2.

5. Experiment

In this study, the experiment is done by applying the famous benchmark track given by the ontology alignment evaluation initiative (OAEI) (http://oaei.ontologymatching.org/2016/). In the benchmark track, each testing case comprises two to-be-mapped ontologies and the reference alignment, to assess the effectiveness of the ontology matcher. Table 1 provides a concise statement of OAEI’s benchmark track.

5.1. Experimental Configuration. In this study, we conduct the experiment by means of the four mentioned similarity measures above, i.e., Levenshtein distance, linguistic distance, cosine distance, and structure distance. The similarity measures should be applied to both input ontologies, and the results are saved in the CSV file before the execution of iNSGA-II. This is done to avoid recalculating the similarities and to boost efficiency of the matching process. The proposed iNSGA-II utilizes the following parameters which provide the trade-off settings acquired empirically to fulfil the highest average quality of alignments across all testing cases in the exploited dataset. Using this parameter configuration, the experiment of this study proves that the selected parameters are robust to all the heterogeneous issues between two ontologies and are expected to be robust for the universal situations of heterogeneity in the real world.

Scales of population = 100; iteration for local search = 30; mutation probability = 0.05; crossover probability = 0.7; scales of differential individuals $C = 3$; fluctuation coefficient FC = 0.2; budget condition = 3000 fitness evaluations.

5.2. Experimental Results. For the purpose of comparing the quality of our approach with the NSGA-II-based ontology matcher and OAEI’s participants (http://oaei.ontologymatching.org/2016/results/benchmarks/index.html), the obtained alignments have been assessed with conventional recall, precision, and $f$-measure. The results of NSGA-II and iNSGA-II are mean values of thirty independent runs, and we choose the individual of the best $f$-measure from knee solutions as a representative. Table 2 displays the mean values and standard deviation of the produced three metrics from our proposal and NSGA-II. Table 3 shows the $t$-test statistical analysis [32] of values in Table 2. It can be seen from the tables that in all testing cases, as the characteristics of exploitation and exploration of MKE, the quality of alignments obtained by iNSGA-II is better than or equal to NSGA-II.

In Figure 3, $X$-axis represents different ontology matchers and $Y$-axis shows the value of $f$-measure, precision, and recall. Because of iNSGA-II which takes both recall and precision into consideration and the exploitation of the local search, it can be seen from Figure 3 that the $f$-measure of our proposal is 0.92, which outperforms all the OAEI’s participants and NSGA-II in terms of the $f$-measure. Figure 4 displays that our proposal is more effective than NSGA-II by compared evaluation times of the $f$-measure value reaching 0.6.

From the above, the quality of alignments of iNSGA-II is better than that of NSGA-II and OAEI’s participants, which demonstrates its better trade-off on the algorithm’s exploration and exploitation. To conclude, iNSGA-II can efficiently determine high-quality ontology alignments when matching various heterogeneous ontologies.
Table 1: The concise statement for the benchmark track.

| ID   | Concise statement                                           |
|------|------------------------------------------------------------|
| 101-104 | Two identical ontologies                                   |
| 201-210 | Two ontologies with different terminology and semantics characteristics |
| 221-247 | Two ontologies with different structure characteristics    |
| 248-266 | Two ontologies with different terminology, semantics, and structure characteristics |
| 301-304 | The real-world ontologies                                  |

Table 2: Comparison of the $f$-measure, recall, and precision obtained by iNSGA-II and NSGA-II through the mean value and standard deviation.

| ID   | $f$-measure (st. dev.) | Recall (st. dev.) | Precision (st. dev.) | $f$-measure (st. dev.) | Recall (st. dev.) | Precision (st. dev.) |
|------|------------------------|-------------------|----------------------|------------------------|-------------------|----------------------|
| 101  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 103  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 104  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 201  | 0.94 (0.02)            | 0.90 (0.01)       | 0.98 (0.02)          | 0.95 (0.00)            | 0.95 (0.01)       | 0.95 (0.00)          |
| 203  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 204  | 0.98 (0.01)            | 1.00 (0.00)       | 0.96 (0.02)          | 0.99 (0.00)            | 0.99 (0.00)       | 0.99 (0.00)          |
| 205  | 0.94 (0.02)            | 0.91 (0.02)       | 0.91 (0.01)          | 0.93 (0.00)            | 0.93 (0.01)       | 0.93 (0.01)          |
| 206  | 0.91 (0.01)            | 0.92 (0.01)       | 0.91 (0.01)          | 0.93 (0.00)            | 0.93 (0.01)       | 0.93 (0.01)          |
| 211  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 222  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 231  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 233  | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          | 1.00 (0.00)            | 1.00 (0.00)       | 1.00 (0.00)          |
| 301  | 0.85 (0.02)            | 0.81 (0.02)       | 0.90 (0.01)          | 0.86 (0.01)            | 0.82 (0.01)       | 0.91 (0.01)          |
| 302  | 0.74 (0.01)            | 0.58 (0.02)       | 0.97 (0.01)          | 0.75 (0.00)            | 0.62 (0.01)       | 0.95 (0.01)          |
| 303  | 0.87 (0.02)            | 0.81 (0.01)       | 0.93 (0.00)          | 0.87 (0.01)            | 0.81 (0.00)       | 0.93 (0.00)          |
| 304  | 0.93 (0.02)            | 0.89 (0.02)       | 0.97 (0.01)          | 0.94 (0.01)            | 0.91 (0.01)       | 0.97 (0.01)          |

Table 3: The analysis of the $t$-test statistical.

| ID   | $t$-value of $f$-measure (iNSGA-II, NSGA-II) | $t$-value of recall (iNSGA-II, NSGA-II) | $t$-value of precision (iNSGA-II, NSGA-II) |
|------|---------------------------------------------|----------------------------------------|-------------------------------------------|
| 101  | 0.00                                        | 0.00                                   | 0.00                                       |
| 103  | 0.00                                        | 0.00                                   | 0.00                                       |
| 104  | 0.00                                        | 0.00                                   | 0.00                                       |
| 201  | 2.74                                        | 19.36                                  | −8.22                                     |
| 203  | 0.00                                        | 0.00                                   | 0.00                                       |
| 204  | 5.48                                        | 0.00                                   | 8.22                                       |
| 205  | 5.48                                        | 12.25                                  | −7.75                                     |
| 206  | 10.95                                       | 5.48                                   | 10.95                                     |
| 221  | 0.00                                        | 0.00                                   | 0.00                                       |
| 222  | 0.00                                        | 0.00                                   | 0.00                                       |
| 223  | 0.00                                        | 0.00                                   | 0.00                                       |
| 224  | 0.00                                        | 0.00                                   | 0.00                                       |
| 225  | 0.00                                        | 0.00                                   | 0.00                                       |
| 228  | 0.00                                        | 0.00                                   | 0.00                                       |
| 230  | 0.00                                        | 0.00                                   | 0.00                                       |
| 231  | 0.00                                        | 0.00                                   | 0.00                                       |
| 301  | 2.45                                        | 2.45                                   | 3.87                                       |
| 302  | 5.48                                        | 9.80                                   | −7.74                                     |
| 303  | 0.00                                        | 0.00                                   | 0.00                                       |
| 304  | 2.45                                        | 4.90                                   | 0.00                                       |
6. Conclusion

Ontology alignment can find heterogeneous entities of diverse ontologies which can speed up the translation discovery and integrate the knowledge. In this study, the iNSGA-II, which incorporates into NSGA-II’s evolutionary process the monkey king evolution algorithm (MKE) as the LS algorithm, is introduced to match the heterogeneous entities which can effectively determine the knee solution of PF enhancing the quality of alignments. By means of the benchmark track of OAEI, the experiment results show that our proposal outperforms the NSGA-II-based ontology matching technique and the OAEI’s participators. To further enhance the quality of alignments, in the future, we will focus on semiautomatic ontology matching techniques which introduce interacting process into the MOEA-based matcher, and we will be interested in the word embedding-based similarity measure which would take background knowledge and entities’ information of ontologies into consideration to improve the performance of ontology matching [33] techniques.

Data Availability

The data used to support this study can be found in http://oaei.ontologymatching.org.

Conflicts of Interest

The authors declare that they have no conflicts of interest related to this work.

Acknowledgments

This work was supported by the 2019 Fujian Province Undergraduate Universities Teaching Reform Research Project (no. FBJG20190156), the 2018 Program for Outstanding Young Scientific Researcher in Fujian Province University, the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (no. YQ20206), the Program for New Century Excellent Talents in Fujian Province University (no. GY-Z18155), the Scientific Research Foundation of Fujian University of Technology (no. GY-Z17162), the Science and Technology Planning Project in Fuzhou City (no. 2019-G-40), and the Foreign Cooperation Project in Fujian Province (no. 2019I0019).

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.
[2] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: a tutorial,” Reliability Engineering & System Safety, vol. 91, no. 9, pp. 992–1007, 2006.
[3] 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.
[4] S. Bechikh, L. B. Said, and K. Ghedira, “Searching for knee regions of the pareto front using mobile reference points,” Soft Computing, vol. 15, no. 9, pp. 1807–1823, 2011.
[5] J. Branke, K. Deb, H. Dierolf et al., “Finding knees in multi-objective optimization,” in International Conference on Parallel Problem Solving from Nature, pp. 722–731, Springer, Berlin, Germany, 2004.
[6] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: a literature review," Expert Systems with Applications, vol. 42, no. 2, pp. 949–971, 2015.
[7] G. Acampora, U. Kaymak, V. Loia et al., “Applying NSGA-II for solving the ontology alignment problem,” in Proceedings of the 2013 IEEE International Conference on Systems, Man, and
[8] M. Biniz and R. El Ayachi, “Optimizing ontology alignments by using Neural NSGA-II,” Journal of Electronic Commerce in Organizations, vol. 16, no. 1, pp. 29–42, 2018.

[9] X. Xue, J. Chen, and X. Yao, “Efficient user involvement in semi-automated ontology matching,” IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 1–11, 2018.

[10] X. Xue, Y. Wang, and W. C. Hao, “Using memetic algorithm for instance coreference resolution,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 2, pp. 580–591, 2016.

[11] X. Xue, J. Chen, and X. Yao, “Using memetic algorithm for instance coreference resolution,” Knowledge-Based Systems, vol. 97, pp. 144–157, 2016.

[12] J. M. V. Naya, M. M. Romero, J. P. Loureiro et al., “Improving ontology alignment through a memetic aggregation of similarity measures,” Information Sciences, vol. 250, pp. 1–20, 2013.

[13] 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.

[14] S. Chu, X. Xue, J.-S. Pan et al., “Optimizing ontology alignment in vector space,” Journal of Internet Technology, vol. 21, no. 1, pp. 15–23, 2020.

[15] X. Xue, J. Chen, and X. Yao, “Interactive ontology matching based on compact interactive evolutionary algorithm,” Knowledge-Based Systems, vol. 137, pp. 94–103, 2017.

[16] G. Acampora and A. Vitiello, “A study on local search meta-heuristics for ontology alignment,” in Computational Intelligence for Semantic Knowledge Management, pp. 53–70, Springer, Cham, Switzerland, 2020.

[17] C.-H. Chen, “A cell probe-based method for vehicle speed estimation,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E103-A, no. 1, pp. 265–267, 2020.

[18] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, “A probability density function generator based on neural networks,” Physica A: Statistical Mechanics and Its Applications, vol. 541, Article ID 123344, 2020.

[19] G. A. Miller, “Wordnet: a lexical database for English,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.

[20] G. Sidorov, A. Gelbukh, H. Gómez -Adorno et al., “Soft similarity and soft cosine measure: similarity of features in vector space model,” Computación Y Sistemas, vol. 18, no. 3, pp. 491–504, 2014.