

Research Article

Solving Ontology Metamatching Problem through Improved Multiobjective Particle Swarm Optimization Algorithm

Yikun Huang¹,¹ Yucheng Zhuang¹,² and Xingsi Xue³

¹Concord University College, Fujian Normal University, Fuzhou, Fujian 350117, China ²School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou, Fujian 350118, China ³Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian 350118, China

Correspondence should be addressed to Yikun Huang; fjnuhyk@163.com

Received 8 September 2022; Revised 4 October 2022; Accepted 10 October 2022; Published 22 November 2022

Academic Editor: Yifan Zhu

Copyright © 2022 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.

In recent years, knowledge representation in the Artificial Intelligence (AI) domain is able to help people understand the semantics of data and improve the interoperability between diverse knowledge-based applications. Semantic Web (SW), as one of the methods of knowledge representation, is the new generation of World Wide Web (WWW), which integrates AI with web techniques and dedicates to implementing the automatic cooperations among different intelligent applications. Ontology, as an information exchange model that defines concepts and formally describes the relationships between two concepts, is the core technique of SW, implementing semantic information sharing and data interoperability in the Internet of Things (IoT) domain. However, the heterogeneity issue hampers the communications among different ontologies and stops the cooperations among ontology-based intelligent applications. To solve this problem, it is vital to establish semantic relationships between heterogeneous ontologies, which is the so-called ontology matching. Ontology metamatching problem is defined as a multiobjective optimization model in this work, and a multiobjective particle swarm optimization (MOPSO) with diversity enhancing (DE) (MOPSO-DE) strategy is proposed to better trade off the convergence and diversity of the population. The well-known benchmark of the Ontology Alignment Evaluation Initiative (OAEI) is used in the experiment to test MOPSO-DE's memory consumption.

1. Introduction

In recent years, knowledge representation in the Artificial Intelligence (AI) domain is able to help people understand the semantics of data and improve the interoperability between diverse knowledge-based applications. Semantic Web (SW) [1–4], as one of the methods of knowledge representation, is the new generation of World Wide Web (WWW), which integrates AI with web techniques and dedicates to implementing the automatic cooperations among different intelligent applications. Ontology, as an information exchange model that defines concepts and formally describes the relationships between two concepts, is the core technique of SW, implementing semantic information sharing and data interoperability in the Internet of Things (IoT) domain [5]. However, one concept may be described with different terminologies in various specific fields, yielding the heterogeneity problem among different ontologies [6]. The heterogeneity issue hampers the communications among different ontologies and stops the cooperations among ontology-based intelligent applications. If there is incapability of sharing knowledge for the sake of the final goal of SW, machines could not cooperate with each other. Therefore, the ontology heterogeneity issue directly affects the development of SW. To solve this problem, it is vital to establish semantic relationships, such as classes and properties for correspondences between heterogeneous ontologies so as to find the identical entity mappings, which is the so-

called ontology matching. Similarity measures, which represent the core technology of the ontology matching [7], are used to compute the similarity value between two entities. The selection, combination, and tuning are three components of the similarity integration. Among them, selection is of importance for the integration, for some contradictory results are not possible to be integrated. Since only using a single similarity measure fails to ensure the confidence on all heterogeneous scenarios, various similarity measures are integrated to obtain a satisfactory alignment [8]. Ontology metamatching problem is aimed at how to choose the decent similarity measures, assign appropriate weights for them, and how to verify the alignment by removing the incorrect correspondences to enhance the quality of matching results, which is commonly a complex optimization problem with many local optima [9].

Although there have been many studies on solving the ontology matching problems with single-objective optimization strategies [10], it only optimizes one of the two conflicting objectives, namely, recall or precision. Optimizing recall (or precision) will result in decreasing precision (or recall), resulting in bias improvement. Due to the alignment's fmeasure that comprehensively considers these two objectives, it is necessary to trade off two conflicting objectives at the same time to achieve better results. However, the research on the multiobjective ontology matching technology is still in its infancy. To this end, MOPSO is used in this work to improve the alignment's quality. Most MOPSO algorithms use an external archive to save local and global best particles, greatly increasing the memory consumption when there are multiple local optimal solutions on the Pareto front, and sometimes these solutions cannot fully converge to the real Pareto fronts. For MOPSO, how to enable the solutions to fast converge to the Pareto fronts (PF) and how to make solutions evenly distributed on the true PF are two fundamental issues to be addressed. The search space of the solutions is greatly cut down by the local and global best particles used to guide the update of solutions at current generation, and MOPSO tends to be trapped into local optima due to its fast convergence and uneven distribution. Therefore, enhancing the population diversity to reduce the probability of premature convergence is vital to the performance of MOPSO. In recent years, decomposition-based archiving approach for MOPSO has become a popular method to balance convergence and diversity of the population [11], but it still faces great challenges and increases memory consumption in dealing with complex optimization problems with multiple local optima, such as the ontology metamatching problem. Multiobjective particle swarm optimization for feature selection with fuzzy cost [12] also has good performance in improving the diversity of population. This method defines a fuzzy crowding distance measure to save candidate solutions and determine the global best particles in the archive. To better balance the convergence and diversity of the population and save the algorithm's memory consumption, an improved MOPSO algorithm for enhancing population diversity in MOPSO is proposed in this work to better guide the update of the solutions in the swarm. Since the particle swarm optimization (PSO) is a popular strategy to solve the ontology

matching problem [13], this paper proposes a multiobjective PSO based on diversity enhancing strategy (MOPSO-DE) to solve it. In particular, MOPSO-DE uses a diversity enhancing strategy to efficiently improve the alignment's quality. To be specific, the contributions of this paper are as follows: (1) a multiobjective optimization model is constructed for the ontology metamatching problem; (2) a general multiobjective optimization framework is presented for solving the ontology metamatching problem to evaluate the alignment's quality; and (3) a MOPSO-DE is proposed to solve the ontology metamatching problem efficiently.

The rest of the paper is organized as follows: Section 2 presents the development of the existing swarm intelligence algorithm-based ontology matching techniques. Section 3 describes the related concepts of the ontology matching problem and the mathematical optimization model. Section 4 elaborates the implementation details of MOPSO-DE, Section 5 presents the experimental results and analysis, and Section 6 summarizes the work of this paper and determines the direction of following work.

2. Swarm Intelligence Algorithm-Based Ontology Matching Technique

Due to the complex optimization problem in the ontology matching domain, the Swarm algorithms (SI) algorithms, such as Brain Storm Optimization (BSO) algorithm [14], Parallel Compact Differential Evolution (PCDE) algorithm [15], Compact Genetic Algorithm (CGA) [16], Artificial Bee Colony (ABC) algorithm [17], and Evolutionary Algorithm (EA) [18], have become popular methods to integrate heterogeneous ontologies.

Martinez-Gil and Aldana-Montes are the first to propose Genetics for Ontology Alignments (GOAL) [19] to determine the suitable aggregating weights generated by each similarity measure using the Evolutionary Algorithm (EA). Alexandru-Lucian and Iftene [20] optimize both parameters and the threshold in the matching process to further filter unauthentic results in the matching process. Acampora et al. [21] improve the quality of the solution and the convergence speed of EA through a local search process. Xue and Wang [22] utilize a new metric to determine the weight required by several pairs of ontologies in the matching process at a time in order to approximately measure the alignment's f-measure and guide the search direction of the algorithm. He et al. [23] propose an optimization method named Artificial Bee Colony (ABC) algorithm to integrate diverse similarity measures in the matching process to improve the alignment's quality. Xue et al. [24] use NSGA-III [25] to integrate diverse similarity measures in the matching process. However, as the scale of the similarity measures increases, the quality decreases. These proposals need distributions of memory space to store the similarity matrices determined by similarity measures, which increases the space complexity of the algorithm and thus cannot obtain high-quality alignment. To solve this problem, the Genetic Algorithm based Ontology Matching (GAOM) [26] uses EA to find the optimal entity matching pairs to achieve high-quality matching results. Alves et al. [27]

propose a hybrid GA, which combines the EA with a local search strategy to match instance information and determine the optimal concept mapping. Recently, Chu et al. [28] propose model ontology in vector space and propose a Compact Evolutionary Algorithm (CEA) to determine the optimal alignments.

Although the above ontology matching methods based on SI to integrate heterogeneous ontologies can ensure the quality of alignment, the convergence speed of the algorithm is slow. Compared with the SI mentioned above in the ontology matching domain, PSO has the advantage that there is only one-way information flow; that is, all particles can converge quickly. MapPSO [29] addresses the ontology matching problem by using PSO and introduces a new metric based on statistical results to approximately evaluate the alignment's quality. Huang et al. [30] propose a compact PSO (cPSO) for sensor ontology matching on Artificial Internet of Things (AIoT). However, these proposals all have the drawback of premature convergence so that the global optimal solution cannot be found to obtain a high-quality alignment. To overcome this drawback, multiobjective PSO (MOPSO) is introduced to the ontology matching domain. Xue et al. [31] propose a compact MOPSO applied in large-scale biomedical ontology domain.

There exist a number of local optimal solutions in the ontology metamatching problem, and infinite candidate solutions can be found in the search space [9]. One of the main concerns in the existing MOPSO in terms of the ontology metamatching domain is how to enhance population diversity to improve the alignment's quality. To this end, this work puts forward an improved MOPSO to get rid of an external archive with a high computational cost, and it utilizes a diversity enhancing strategy to strike a balance between convergence and diversity.

3. Preliminaries

3.1. Ontology and Ontology Alignment

Definition 1. An ontology is defined as a 3-tuple [32].

$$O = (C, P, I), \tag{1}$$

where C represents a collection of objects targeted at a certain domain, e.g., "book" can be interpreted as a class of all book objects in a library; *P* represents the set of relationships between two objects, e.g., "has author" is a relationship between the "book" object and the "author" object; and I represents the collection of specific individual instances corresponding to instance objects, e.g., "Computer Book" is an instance of the class 'book." Figure 1 presents an example of two ontologies under alignment. The classes, properties, and instances of ontology are called entities. The alignment of two ontologies is the process of finding correspondences between entities. In addition, rounded rectangles stand for classes, e.g., "Vehicle," and one-way arrows are properties between two objects, e.g., "is a." To ensure the semantic interoperation between different systems, ontology is needed to describe semantic relations. Due to ontology designers'

subjectivity, the concepts of ontology may have different descriptions. Thus, the heterogeneity problem of ontology is presented. To solve this problem, it is necessary to find out the mappings between the concepts of ontology, i.e., the so-called ontology alignment.

Definition 2. Ontology alignment is a set of correspondences between entities, which is defined as a 4-tuple.

$$C = \left(e, e', \operatorname{sim}, \operatorname{rel}\right),\tag{2}$$

where e and e' are, respectively, the entities of source and target ontologies, sim is the similarity in the range [0,1], rel is the relation between e and e'. In Figure 1, the double-sided arrows establish correspondences between the two entities. For example, a connection between "Wheeled" in ontology 1 and "Wheeled Vehicle" in ontology 2 is established with a similarity of 0.82. The value of similarity represents the confidence, for example, 0 means the nonequivalence relation and 1.0 is the equivalence relation.

Definition 3. The process of ontology alignment is usually defined as a function.

$$A = \varphi(O_1, O_2, \text{RA}, p, r), \tag{3}$$

where A is the final matching result; O_1 and O_2 are source ontology and target ontology to be matched; RA is the reference alignment; p is the set of parameters such as weights and thresholds; and r is an external resource, such as an external dictionary, e.g., WordNet.

Figure 2 shows the process of ontology alignment: in the matching process, only the matching results' confidence value greater than the threshold $t \in [0, 1]$ are considered to be authentic, so how to filter the unauthentic results is the key to the ontology alignment process, and the filtering process is finished by the similarity measures.

3.2. Similarity Measures. Generally, there are three kinds of similarity measures in the ontology matching field, i.e., linguistic-based, string-based, and structure-based.

Linguistic-based similarity measure uses the external digital dictionary, such as WordNet [33] to calculate the similarity value of two words by comparing their relation of hypernymy or whether they are synonymous. Wu and Palmer [34] is a classic similarity measure used in WordNet, which calculates the correlation of two words by considering the depth of the two synonym sets and the depth of the Least Common Subsumer (LCS). The formula of Wu and Palmer is defined as follows:

$$sim_{l}(s_{1}, s_{2}) = \frac{2 \times dep(LCS(s_{1}, s_{2}))}{dep(s_{1}) + dep(s_{2})},$$
(4)

where s_1 and s_2 are the strings to be matched. LCS (s_1, s_2) represents the closest common parent concept of s_1 and s_2 ; and dep (s_1) , dep (s_2) , and dep $(LCS(s_1, s_2))$ are depth of s_1 and s_2 and LCS (s_1, s_2) , respectively.



FIGURE 1: An example of two ontologies under alignment.



FIGURE 2: The process of ontology alignment.

tance calculation algorithms, such as N-gram [35], SMOA

[20], Levenshtein [36], and Jaro-Winkler [37] distance.

Because of the superior performance of N-gram in compar-

ing the similarity of two strings [35], this paper uses N-gram

to calculate the similarity value of strings in the field of

String-based similarity measure utilizes different dis-

ontology alignment, and its formula is as follows:

$$\operatorname{sim}_{\operatorname{str}}(s_1, s_2) = \frac{2 \times \operatorname{com}(s_1, s_2)}{N_{s_1} + N_{s_2}},$$
(5)

where $com(s_1, s_2)$ is the amount of common substrings of s_1 and s_2 and N_{s_1} and N_{s_2} are the number of substrings, respectively. According to the paper [38], the best performance can be achieved when the number of substrings is 3.

Structure-based similarity measure mainly calculates the similarity value of two entities through the relationship between superclasses and subclasses. Matching entities are considered structurally similar if they have the same amount of superclasses and subclasses. In this paper, similarity is calculated based on the amount of superclasses and subclasses of entities in different ontologies, and the relevant formula is defined as follows:

$$sim_{struct}(s_1, s_2) = \begin{cases} 0, \text{ if two entities have the same amount of superclasses and subclasses,} \\ 1, \text{ if two entities have different amount of superclasses and subclasses.} \end{cases}$$
(6)

3.3. Ontology Metamatching Problem. Since one similarity measure cannot guarantee the matching quality in all circumstances, various similarity measures are usually integrated to enhance the quality of matching results and get the final similarity value. In this work, the weighted average strategy is chosen to integrate different similarity measures, which is defined as follows:

$$sim(s_1, s_2) = w_1 \times sim_l(s_1, s_2) + w_2
\times sim_{str}(s_1, s_2) + w_3 \times sim_{struct}(s_1, s_2),$$
(7)

where $w_i \in [0, 1]$, $i = 1, 2, 3, \sum w_i = 1$.

How to assign the appropriate weights to similarity measures and filtering weights of the ontology matching

result for ontology alignment is referred to the ontology metamatching problem [19]. In addition, the weight tuning process requires a trade-off between two conflicting objectives, i.e., the alignment's recall and precision, which are as follows:

$$\operatorname{recall} = \frac{|R \cap A|}{|R|},\tag{8}$$

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

where R and A are reference alignment and the final alignment, respectively, formulated by experts in specific fields.



FIGURE 3: The framework of multiobjective particle swarm optimization algorithm based on diversity enhancing strategy for solving ontology metamatching problem.

When recall equals 1, all correct matching pairs in the correspondences are found, while when precision equals 1, all matching pairs in the correspondences found are correct. To be specific, the multiobjective optimization model of the ontology metamatching problem in this paper is defined as follows:

$$\max f(W, T) = (\operatorname{recall}(W, T), \operatorname{precision}(W, T)),$$

s.t. $W = (w_1, w_2, \cdots)^T, T \epsilon[0, 1],$
$$\sum_{i=1}^N w_i = 1, w_i \epsilon[0, 1],$$

(10)

where W and T represent the weight sets and the filtering threshold, respectively; w_i is the aggregating weights of the *i*-th similarity measure; N is the amount of similarity measures; and recall(W, T) and precision(W, T) calculate the recall value and precision value of the ontology metamatching results under the parameters W and T.

4. Multiobjective Particle Swarm Optimization Algorithm Based on Diversity Enhancing Strategy

To effectively solve the ontology metamatching problem, a framework of multiobjective particle swarm optimization algorithm based on diversity enhancing strategy for solving the ontology metamatching problem is proposed. In order to simplify the structure of the algorithm and reduce the influence of external archive on the performance of solving complex optimization problems, such as the ontology metamatching problem, a diversity enhancing strategy is utilized in this framework to search for the optimal solutions to obtain perfect alignment. The framework of MOPSO-DE for ontology metamatching is given in Figure 3, where s_1 , s_2 , and s_3 represent the similarity measures; M_1 , M_2 , and M_3 represent the corresponding similarity matrix; and w_1 , w_2 , and w_3 are the aggregating weights for the similarity matrix, respectively. M is the matrix aggregated by M_1, M_2 and M_3 ; Threshold is the filtering threshold to exclude unauthentic results; the multiobjective processing module, representing the core of the framework for the ontology metamatching problem, utilizes a MOPSO-DE algorithm to trade off the two conflicting objectives simultaneously, i.e., recall and precision, to evaluate the alignment's quality.

In Figure 3, the final goal of MOPSO-DE for ontology metamatching is to obtain the corresponding similarity matrix for each similarity measure, assign the right weight



FIGURE 4: An example of encoding aggregating weights.

and find a suitable threshold to filter unauthentic alignment, to trade-off the two conflicts evaluation metrics, namely, recall and precision, and then, with the help of the use of the diversity enhancing strategy, to ensure the perfect alignment.

4.1. Encoding Mechanism. In this paper, a decimal coding scheme [13] is used to encode each solution, where each particle consists of the weight set and a threshold. Figure 4 shows an example of encoding aggregating weights.

 r_1' and r_2' are the cut points used to get the weights. Two cut points are needed to get three subintervals on one interval. n - 1 cut points are used to represent the aggregating weights of n similarity measures, and the sum of the weights is equal to 1, which ensures the efficiency of the algorithm. This coding mechanism ensures the allocation of the weight sets. In particular, the encoding process is as follows: (1) randomly generate n real numbers in [0,1], marked as $r_1, r_2, \ldots, r_{n-1}$, and r_n , respectively; (2) sort the first n - 1 cut points r_1, r_2, \ldots , and r_{n-1} in ascending order to get r_1', r_2', \ldots , and r_{n-1}' , and r_n is the threshold to filter invalid results; and (3) calculate the aggregating weights according to the following equation:

$$w_{i} = \begin{cases} r_{1}', & t = 1, \\ r_{i}' - r_{i-1}', & 1 < i < n, \\ 1 - r_{n-1}', & i = n. \end{cases}$$
(11)



FIGURE 5: The framework of diversity enhancing strategy.

4.2. Diversity Enhancing Strategy. The traditional MOPSO algorithm requires the external archive to save information of particles, which increases the memory consumption of the algorithm [13]. To improve the algorithm's efficiency, in this work, we use the diversity enhancing strategy to execute the evolutionary process. Diversity enhancing strategy is introduced to trade off convergence and diversity. In the proposed MOPSO-DE, pairwise particles at each generation guide the update of the particles instead of the local and global best particles. In particular, two particles are first randomly selected from the elite set to compete in pairs, and then, the particles with better recall and precision value are marked as the winner particles, which guide the update of the loser particles at each generation. To better illustrate the principle of the diversity enhancing strategy, Figure 5 is given to present its framework.

MOPSO cannot solve the ontology metamatching problem well to obtain good alignment because there are many optimal solutions which affect the performance of the algorithm. Therefore, the diversity enhancing strategy guides the update of particles, which no longer relies on optimal solutions and external archive, simplifying the structure of the algorithm, greatly reducing memory consumption, and ensuring the quality of the final alignment. Assuming that $X_{w,k}(t)$ is the position variable of the winning particle and $V_{l,k}(t)$ and $X_{l,k}(t)$ are the velocity variable and position variable of the loser particle, respectively. For the *i*-th particle in generation *t*, the update formula on the loser particles in the *k*-round competition is as follows:

$$V_{l,k}(t+1) = \varphi_1 V_{l,k}(t) + \varphi_2 (X_{w,k}(t) - X_{l,k}(t)), \qquad (12)$$

$$X_{l,k}(t+1) = X_{l,k}(t) + V_{l,k}(t+1),$$
(13)

where φ_1 and φ_2 are two weight vectors. As can be seen from the update formula of loser particles in the *k*-round competition, the position update formula of particles adopts the position update formula of the classic PSO, and the diversity enhancing strategy affects the update of velocity of loser particles in the competition. The first part of the velocity update formula $\varphi_1 V_{l,k}(t)$ is consistent with the inertia term of the classic PSO algorithm. The second part $\varphi_2(X_{w,k}(t) - X_{l,k}(t))$ indicates that loser particles learn from winner particles through DE to update instead of being guided by the optimal solutions in the population. In addition, because DE can guide the loser particles to update, it can further ensure the performance of the algorithm and get better ontology metamatching results.

4.3. The Pseudocode of Multiobjective Particle Swarm Optimization Algorithm Based on Diversity Enhancing Strategy. Given the source and target ontologies O_1 and O_2 , number of iteration N, population size n, particle's current position X, particle's current velocity V, elite particle set L, current generation t, particle's fitness value P_{recall} and $P_{\text{precision}}$, the problem dimension dim, and the winner particle $P_{\rm win}$ and the loser particle $P_{\rm loser}$, the pseudocode of MOPSO-DE is presented in Algorithm 1. MOPSO-DE first initializes the velocity and position of the particles and calculates the two objective function values of the particles, e.g., P_{recall} and $P_{\text{precision}}$, as fitness value. Since the selection of elite particles should have good convergence and distribution, this paper combines nondominated sort with the calculation of the crowding distance of the solution from the fronts' solution set to obtain the elite particle set L. The calculation of crowding distance requires sorting all solutions in the population in descending order. Specifically, the crowding distance of the first and last solution is set to a maximum. The crowding distance of the *i*-th solution is the product of the absolute value of the objective function of the i-1-th and i+1-th solution. Then, the recall and precision values of particles a and b are compared according to the pairwise competition strategy. If both recall and precision of particle *a* are greater than that of particle b, then a is the winner particle. Among them, the winner particle is defined as P_{win} , and it guides the update of the particle P_{loser} .

5. Experiment

5.1. Experimental Configurations. In order to verify the effectiveness of MOPSO-DE, the well-known benchmark

Input:
two ontologies O_1 and O_2 , number of iteration N, population size n;
Particle's current position X, Particle's current velocity V, elite particle set L;
Output:
Winner particle <i>P</i> _{win} ;
Initialization:
1 initialize generation $t = 0$;
2 calculate particle's fitness value P_{recall} and $P_{\text{precision}}$;
3 for $(i = 0; i < \dim; i + +)$.
4 $V[i]$ =random(0, 1);
5 $X[i]$ =random(0, 1);
6 end for.
7 NonDominatedSort()
<pre>8 calculateCrowdingDistance();</pre>
9 sortFronts();
10 get elite particle set <i>L</i> ;
Evolution
11 get elite particle set <i>L</i> ;
12 while $t < N$ do
13 randomly select two particles <i>a</i> and <i>b</i> from the elite set <i>L</i> ;
14 P_{recall} and $P_{\text{precision}}$ of particles <i>a</i> and <i>b</i> are calculated, respectively;
15 $[P_{win}, P_{loser}] = compete(a, b);$
16 if $P_{\text{recall}}(a) > P_{\text{recall}}(b)$ and $P_{\text{precision}}(a) > P_{\text{precision}}(b)$ then
17 $P_{\text{win}} = a;$
18 else
19 $P_{\rm win} = b;$
20 end if.
21 Update:
22 $V_{P_{\text{loser}}}' \leftarrow \text{update } P_{\text{loser}}' \text{s velocity according to formula (12);}$
23 $X_{P_{loser}}' \leftarrow$ update P_{loser}' s position according to formula (13);
24 update particle's fitness value P_{recall} and $P_{\text{precision}}$;
25 $t = t + 1;$
26 end while
27 return $P_{\rm win}$.

ALGORITHM 1: The pseudocode of MOPSO-DE.

ID	Brief description
101-104	The matched source ontologies and target ontologies with same lexical, linguistic, and structural characters.
201-210	The matched source ontologies and target ontologies with same structural characters but with different lexical and linguistic characters.
221-247	The matched source ontologies and target ontologies with same lexical and linguistic features but with different structural characters.
248-266	The matched source ontologies and target ontologies with different lexical, linguistic, and structural characters.
301-304	The real ontologies.

provided by the Ontology Alignment Evaluation Initiative (OAEI) http://oaei.ontologymatching.org/ is used. Each dataset of benchmark consists of two ontologies and reference alignments, which are used to evaluate the performance of the different ontology matching systems. The reference alignments are provided by OAEI's benchmark to evaluate the quality of alignments, and Table 1 briefly describes the benchmarks for OAEI. In the experiment, MOPSO-DE is compared with PSO-based matching technique [29], MOPSO-based matching technique [13], and OAEI's participants, i.e., edna [39], AgrMaker [40], AROMA [41], ASMOV [42], CODI [43], Eff2Match [44], Falcon [45], GeRMeSMB [46], MapPSO [47], RiMOM [48], SOBOM [49], and Tax-oMap [50]. The experimental results of MOPSO-DE, PSO, and MOPSO in this paper are obtained from the mean

TABLE 2: Comparison among recall-driven and precision-driven PSO, MOPSO, and MOPSO-DE. In round parenthesis, there are recall, precision, and f-measure, respectively.

No.	PSO (R)	PSO (P)	MOPSO	MOPSO-DE
101	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00,1.00, 1.00)	(1.00, 1.00, 1.00)
103	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
104	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
201	(0.92, 1.00, 0.95)	(0.92, 1.00,0.96)	(0.92,1.00, 0.96)	(0.92, 1.00, 0.96)
202	(0.10, 0.45, 0.16)	(0.10, 1.00, 0.02)	(0.09,0.81, 0.16)	(0.09, 0.81, 0.16)
203	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
204	(0.98, 1.00, 0.99)	(0.80, 1.00, 0.88)	(0.98, 1.00, 0.99)	(0.98, 1.00, 0.99)
205	(0.92, 0.95, 0.94)	(0.08, 1.00, 0.15)	(0.92, 0.95, 0.94)	(0.92, 0.95, 0.94)
206	(0.91, 0.97, 0.94)	(0.10, 1.00, 0.18)	(0.71, 0.98, 0.82)	(0.86, 0.98, 0.92)
207	(0.92, 0.97, 0.95)	(0.10, 1.00, 0.18)	(0.58, 0.98, 0.73)	(0.87, 0.98, 0.92)
208	(0.83, 0.97, 0.90)	(0.70, 1.00, 0.82)	(0.80, 0.99, 0.88)	(0.83, 0.98, 0.90)
209	(0.44, 0.75, 0.55)	(0.09, 1.00, 0.16)	(0.35, 0.85, 0.48)	(0.35, 0.85, 0.49)
210	(0.52, 0.80, 0.63)	(0.32, 1.00, 0.49)	(0.48, 0.98, 0.65)	(0.50, 0.96, 0.66)
221	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
222	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
223	(0.96, 0.96, 0.96)	(0.08, 1.00, 0.15)	(0.96, 0.98, 0.96)	(0.94, 0.99, 0.96)
224	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
225	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
228	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
230	(1.00, 0.93, 0.96)	(0.11, 1.00, 0.19)	(0.91, 0.98, 0.95)	(0.97, 0.94, 0.95)
231	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
232	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
233	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
236	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
237	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
238	(0.96, 0.96, 0.96)	(0.08, 1.00, 0.15)	(0.95, 0.98, 0.96)	(0.93, 0.99, 0.96)
239	(1.00, 0.96, 0.97)	(0.06, 1.00, 0.12)	(0.93, 0.96, 0.92)	(1.00, 0.96, 0.98)
240	(0.90, 0.87, 0.89)	(0.15, 1.00, 0.26)	(0.74, 0.96, 0.82)	(0.81, 0.96, 0.88)
241	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
246	(1.00, 0.96, 0.98)	(0.06, 1.00, 0.12)	(1.00, 0.96, 0.98)	(1.00, 0.96, 0.98)
247	(0.90, 0.87, 0.89)	(0.15, 1.00, 0.26)	(0.81, 0.96, 0.88)	(0.81, 0.96, 0.88)
248	(0.05, 0.13, 0.07)	(0.04, 1.00, 0.07)	(0.04, 0.96, 0.07)	(0.05, 0.41, 0.09)
249	(0.10, 0.25, 0.14)	(0.01, 1.00, 0.02)	(0.09, 0.81, 0.16)	(0.09, 0.81, 0.16)
250	(0.17, 0.37, 0.24)	(0.03, 1.00, 0.05)	(0.15, 0.71, 0.25)	(0.15, 0.71, 0.25)
301	(0.75, 0.87, 0.81)	(0.18, 1.00, 0.31)	(0.70, 0.91, 0.79)	(0.70, 0.93, 0.80)
302	(0.68, 0.88, 0.77)	(0.20, 1.00, 0.34)	(0.64, 0.94, 0.76)	(0.65, 0.91, 0.76)
303	(0.85, 0.69, 0.76)	(0.02, 1.00, 0.04)	(0.75, 0.94, 0.83)	(0.79, 0.95, 0.86)
304	(0.96, 0.95, 0.96)	(0.03, 1.00, 0.07)	(0.88, 0.98, 0.93)	(0.96, 0.97, 0.96)

values of 30 independent runs. The experimental results are presented by recall, precision, and f-measure. Table 2 shows the alignment quality of recall-driven and precision-driven PSO and MOPSO as well as MOPSO-DE proposed in this paper running independently for 30 times. R and P represent recall-driven and precision-driven in the classic PSO, respectively. Tables 3 and 4 show the comparison of mean and standard deviation in terms of recall and precision among recall-driven and precision-driven PSO, MOPSO, and MOPSO-DE.

Tables 5 and 6 show the statistical analysis of *t*-test based on Tables 3 and 4, respectively. In terms of the memory consumption, MOPSO-DE is compared with MOPSO in Figure 6. In addition, recall and precision are averaged on 30 independent runs, and f-measure is calculated from them. Table 7 shows the comparison of MOPSO-DE with OAEI's participants in terms of the average matching results' quality. We choose f-measure as alignment's metric to trade off recall and precision, since f-measure comprehensively takes recall and precision into account.

No.	PSO (R)	PSO (P)	MOPSO	MOPSO-DE
101	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
103	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
104	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
201	0.92 (1.07 <i>E</i> – 15)	0.92 (1.07 <i>E</i> – 15)	0.92 (1.22 <i>E</i> – 15)	0.92 (1.08 <i>E</i> – 15)
202	0.10 (1.90 <i>E</i> – 16)	0.10 (1.58 <i>E</i> – 17)	0.09 (3.94 <i>E</i> – 3)	0.09 (1.57E - 2)
203	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
204	0.98 (2.21 <i>E</i> – 15)	0.80 (7.51 <i>E</i> – 16)	0.98 (2.22 <i>E</i> – 15)	0.98 (2.22 <i>E</i> – 15)
205	0.92 (1.07 <i>E</i> – 15)	0.08 (1.26 <i>E</i> – 16)	0.92 (2.81E - 1)	0.92 (4.09E - 1)
206	0.91 (9.12 <i>E</i> – 4)	0.10 (1.89 <i>E</i> – 16)	0.71 (1.14 <i>E</i> – 2)	0.86 (1.10 <i>E</i> – 1)
207	0.92 (1.07 <i>E</i> – 15)	0.10 (1.89 <i>E</i> – 16)	0.58 (2.26 <i>E</i> – 15)	0.87 (1.16E - 1)
208	0.83 (1.04 <i>E</i> – 15)	0.70 (5.27 <i>E</i> – 16)	0.80 (1.34 <i>E</i> – 2)	0.83 (7.49 <i>E</i> – 2)
209	0.44 (4.04 <i>E</i> – 16)	0.09 (1.59 <i>E</i> – 16)	0.35 (6.46 <i>E</i> – 4)	0.35 (3.29 <i>E</i> – 2)
210	0.52 (9.08E - 4)	0.32 (5.06E - 16)	0.48 (1.11E - 15)	0.50 (2.52E - 2)
221	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
222	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
223	0.96 (1.15 <i>E</i> – 15)	0.08 (1.26 <i>E</i> – 16)	0.96 (9.27 <i>E</i> – 3)	0.94 (1.51E - 2)
224	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)
225	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
228	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
230	$1.00 \ (0.00E + 0)$	0.11 (2.13 <i>E</i> – 16)	0.91 (1.34 <i>E</i> – 15)	0.97 (4.03E - 2)
231	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E+0)$	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$
232	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E+0)$	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$
233	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	1.00 (0.00E + 0)	1.00 (0.00E + 0)
236	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E+0)$	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$
237	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E+0)$	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$
238	0.96 (1.15 <i>E</i> – 15)	0.08 (1.26 <i>E</i> – 16)	0.95 (5.83 <i>E</i> – 3)	0.92 (5.08E - 15)
239	$1.00 \ (0.00E + 0)$	0.06 (1.18 <i>E</i> – 16)	0.93 (2.65 <i>E</i> – 1)	1.00 (4.51E - 2)
240	0.90 (1.42 <i>E</i> – 15)	0.15 (3.21 <i>E</i> – 16)	0.74(1.27E-1)	0.81 (1.33E - 1)
241	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$
246	1.00 (0.00E + 0)	0.06 (1.18 <i>E</i> – 16)	1.00 (4.33E - 1)	1.00(2.79E-1)
247	0.90 (1.42 <i>E</i> – 15)	0.15 (3.21 <i>E</i> – 16)	0.81 (3.02 <i>E</i> – 1)	0.81 (1.16E - 1)
248	0.05 (9.47 <i>E</i> – 17)	$0.04 \ (6.32E - 17)$	0.04 (5.20E - 4)	0.05 (4.83E - 3)
249	0.10 (5.81E - 4)	0.01 (1.58 <i>E</i> – 17)	0.09 (4.07E - 2)	0.09 (2.11E - 2)
250	0.17 (3.73 <i>E</i> – 16)	0.03 (2.17 <i>E</i> – 17)	0.15 (5.98 <i>E</i> – 2)	0.15 (2.23 <i>E</i> – 16)
301	0.75 (1.16 <i>E</i> – 15)	0.18 (2.87 <i>E</i> – 16)	0.70 (9.84E - 2)	0.70 (1.13 <i>E</i> – 1)
302	0.68 (0.00E + 0)	0.20 (2.13 <i>E</i> – 16)	0.64 (8.71 <i>E</i> – 16)	0.65 (1.59 <i>E</i> – 2)
303	0.85 (8.74 <i>E</i> – 16)	0.02 (2.76E - 17)	0.75 (1.03 <i>E</i> – 15)	0.79(3.45E-2)
304	0.96 (1.56 <i>E</i> – 15)	0.03 (4.58 <i>E</i> – 17)	0.88 (8.29E - 4)	0.96 (1.37 <i>E</i> – 1)

The configurations of MOPSO and MOPSO-DE are as follows:

- (i) The size of population: 20
- (ii) The number of iterations: 200
- (iii) Learning factor: $c_1 = 2$, $c_2 = 2$
- (iv) String-based similarity measure: N-gram

- (v) Linguistic-based similarity measure: Wu and Palmer method
- (vi) Structure-based similarity measure: out-in degree

Since the ontology metamatching problem is a relatively small-scale problem, the population size is set as 20. In particular, the configurations of PSO and MOPSO are determined according to the corresponding literatures [13, 29] to guarantee the alignment's quality.

TABLE 4: Comparison of mean and standard deviation of precision among recall-driven and precision-driven PSO, MOPSO, and MOPSO-DE.

No.	PSO (R)	PSO (P)	MOPSO	MOPSO-DE
101	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
103	1.00(0.00E+0)	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E + 0)$	$1.00 \ (0.00E + 0)$
104	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
201	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
202	0.45 (7.12 <i>E</i> – 16)	1.00 (0.00E + 0)	0.81 (8.70 <i>E</i> – 3)	0.81 (9.00E - 2)
203	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
204	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
205	0.95 (3.78 <i>E</i> – 16)	1.00 (0.00E + 0)	0.95 (1.45 <i>E</i> – 2)	0.95 (2.11 <i>E</i> – 2)
206	0.97 (1.68 <i>E</i> – 15)	$1.00 \ (0.00E+0)$	0.98 (1.91E - 4)	0.98 (5.86 <i>E</i> – 3)
207	0.97 (8.55E - 16)	1.00 (0.00E + 0)	0.98 (1.63 <i>E</i> – 15)	0.98 (5.21 <i>E</i> – 3)
208	0.97 (1.78 <i>E</i> – 15)	1.00 (0.00E + 0)	0.99 (2.81 <i>E</i> – 3)	0.98 (5.76 <i>E</i> – 3)
209	0.75 (9.35 <i>E</i> – 16)	1.00 (0.00E + 0)	0.85 (5.75 <i>E</i> – 3)	0.85 (5.63E - 2)
210	0.80 (1.96E - 4)	1.00 (0.00E + 0)	0.98 (1.58 <i>E</i> – 15)	0.96 (4.04 <i>E</i> - 2)
221	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
222	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
223	0.96 (1.15 <i>E</i> – 15)	1.00 (0.00E + 0)	0.98 (9.46 <i>E</i> – 3)	0.99 (1.02 <i>E</i> – 2)
224	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
225	1.00 (0.00E + 0)	$1.00 \ (0.00E+0)$	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
228	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
230	0.93 (2.02 <i>E</i> – 15)	1.00 (0.00E + 0)	0.98 (1.45 <i>E</i> - 15)	0.94 (2.38 <i>E</i> – 2)
231	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
232	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
233	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
236	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
237	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
238	0.96 (1.15 <i>E</i> – 15)	1.00 (0.00E + 0)	0.98 (5.63E - 3)	0.98 (1.64 <i>E</i> - 15)
239	0.96 (8.86E - 16)	1.00 (0.00E + 0)	0.96 (9.49 <i>E</i> – 3)	0.96 (1.61E - 2)
240	0.87 (2.29 <i>E</i> – 15)	1.00 (0.00E + 0)	0.96 (1.66E - 2)	0.96 (4.29 <i>E</i> – 2)
241	1.00 (0.00E + 0)	1.00 (0.00E + 0)	$1.00 \ (0.00E + 0)$	1.00 (0.00E + 0)
246	0.96 (8.86E - 16)	1.00 (0.00E + 0)	0.96 (1.55E - 2)	0.96 (1.00 <i>E</i> – 2)
247	0.87 (2.29 <i>E</i> – 15)	1.00 (0.00E + 0)	0.96 (1.61E - 2)	0.96 (1.69 <i>E</i> – 2)
248	0.13 (2.35 <i>E</i> – 16)	1.00 (0.00E + 0)	0.96 (4.35 <i>E</i> – 2)	0.41 (2.56 <i>E</i> – 1)
249	0.25 (3.20 <i>E</i> – 2)	1.00 (0.00E + 0)	0.81 (8.99E - 2)	0.81 (7.47 <i>E</i> – 2)
250	0.37 (0.01E + 0)	1.00 (0.00E + 0)	0.71 (1.41E - 1)	0.71 (2.41 <i>E</i> – 15)
301	0.87 (2.29 <i>E</i> – 15)	1.00 (0.00E + 0)	0.91 (1.05E - 2)	0.93 (2.52E - 2)
302	0.88 (8.23 <i>E</i> – 16)	1.00 (0.00E + 0)	0.94 (1.22 <i>E</i> – 15)	0.91 (3.02 <i>E</i> – 2)
303	0.69 (5.58 <i>E</i> – 16)	1.00 (0.00E + 0)	0.94 (1.12 <i>E</i> – 15)	0.95 (2.36 <i>E</i> – 2)
304	0.95 (9.91 <i>E</i> – 16)	$1.00 \ (0.00E + 0)$	0.98 (1.34 <i>E</i> – 5)	0.97 (7.33 <i>E</i> – 3)

5.2. Experimental Results. As can be seen from Table 2, since the ontology metamatching based on MOPSO-DE optimize two conflicting objectives simultaneously, i.e., recall and precision, better f-measure can be obtained compared with the classic PSO driven by only one objective. The single-objective PSO optimizes the quality of alignment by improving recall or precision, which leads to the sacrifice of another objective. MOPSO can better balance two objectives and achieve better results. In addi-

tion, thanks to the diversity enhancing strategy, the solutions can fast converge to the Pareto fronts and the distribution of Pareto fronts' solutions are evenly distributed to help the algorithm jump from local optimal solutions. Therefore, the proposed MOPSO-DE has better convergence and distribution than MOPSO; thus, better solutions can be found. It can be found that the fmeasure obtained by MOPSO-DE is superior to PSO and MOPSO.

TABLE 5: *t*-test statistical analysis for recall.

TABLE 6: T-test statistical analysis for precision.

No.	MOPSO-DE versus PSO (<i>R</i>)	MOPSO-DE versus PSO (P)	MOPSO-DE versus MOPSO	No.	MOPSO-DE versus PSO (<i>R</i>)	MOPSO-DE versus PSO (P)	MOPSO-DE versus MOPSO
101	0.00	0.00	0.00	101	0.00	0.00	0.00
103	0.00	0.00	0.00	103	0.00	0.00	0.00
104	0.00	0.00	0.00	104	0.00	0.00	0.00
201	0.00	0.00	0.00	201	0.00	0.00	0.00
202	-3.48	-3.48	0.00	202	21.90	-11.56	0.00
203	0.00	0.00	0.00	203	0.00	0.00	0.00
204	0.00	4.20	0.00	204	0.00	0.00	0.00
205	0.00	11.24	0.00	205	0.00	-12.97	0.00
206	-2.48	37.84	7.42	206	9.34	-18.69	0.00
207	-2.36	36.35	13.69	207	10.51	-21.02	0.00
208	0.00	9.50	2.15	208	9.50	-19.01	-8.54
209	-14.98	43.28	0.00	209	9.72	-14.59	0.00
210	-4.34	39.12	4.34	210	21.69	-5.42	-2.71
221	0.00	0.00	0.00	221	0.00	0.00	0.00
222	0.00	0.00	0.00	222	0.00	0.00	0.00
223	-7.25	239.51	-6.18	223	16.10	-5.36	3.93
224	0.00	0.00	0.00	224	0.00	0.00	0.00
225	0.00	0.00	0.00	225	0.00	0.00	0.00
228	0.00	0.00	0.00	228	0.00	0.00	0.00
230	-4.07	116.88	8.15	230	2.30	-13.80	-9.20
231	0.00	0.00	0.00	231	0.00	0.00	0.00
232	0.00	0.00	0.00	232	0.00	0.00	0.00
233	0.00	0.00	0.00	233	0.00	0.00	0.00
236	0.00	0.00	0.00	236	0.00	0.00	0.00
237	0.00	0.00	0.00	237	0.00	0.00	0.00
238	-4.20	9.05	-28.18	238	5.46	-6.67	0.00
239	0.00	114.15	8.48	239	0.00	-13.60	0.00
240	-3.70	27.18	2.08	240	11.49	-5.10	0.00
241	0.00	0.00	0.00	241	0.00	0.00	0.00
246	0.00	18.45	0.00	246	0.00	-21.90	0.00
247	-4.24	31.16	0.00	247	29.16	-12.96	0.00
248	0.00	11.34	11.27	248	5.99	-12.62	-11.60
249	-2.59	20.76	0.00	249	37.74	-13.93	0.00
250	-2.52	2.93	0.00	250	186.22	-6.59	0.00
301	-2.42	25.20	0.00	301	13.04	-15.21	4.01
302	-10.33	155.01	3.44	302	5.44	-16.32	-5.44
303	-9.52	122.24	6.35	303	60.34	-11.60	2.32
304	0.00	37.18	3.19	304	14.94	-22.41	-7.47

This work uses the *t*-test to measure the differences between different matching systems. The *t* value is the absolute value. The larger the *t* value, the more significant the performance difference between the two systems. Since the experiments are running 30 times independently on each testing case, the *t*values are compared with 2.045. Tables 3 and 4 present the mean and standard deviation of recall and precision among recall-driven and precision-driven PSO, MOPSO, and MOPSO-DE in benchmark track. In round parenthesis, there are the standard deviation values. As can be seen from Tables 3 and 4, the mean of recall and precision of MOPSO-DE is higher and the standard deviation is lower, which indicates the efficiency and stability of the MOPSO-DE. In addition, as can be seen from Tables 5 and 6, all *t* values are greater than 2.045, which indicate that MOPSO-DE is significantly different from the classical PSO and MOPSO in terms of performance.





As shown in Table 7, MOPSO-DE achieves the best fmeasure in the testing cases 1XX and 3XX, which shows that the multiobjective evolution mechanism can find better alignments than other ontology matching systems. In addition, the introduction of the diversity enhancing strategy can better balance the diversity and convergence to significantly improve alignment's quality. MOPSO-DE does not need subjective threshold given by experts in advance, which has strong robustness for various matching problems. About the testing cases 202, 209, 248, 249, and 250, our method does not require additional reasoning and repairing process, and less similarity measures are taken into consideration, which is a reasonable method for small-scale matching problems and guarantees the alignment's quality. MOPSO-DE achieves the good but not optimal f-measure than other ontology matching systems such as AgrMaker and ASMOV. While the results of other systems such as edna, AROMA, CODI, Falcon, MapPSO, and TaxoMap almost dropped or close to 0 in testing cases 202, 209, 248, 249, and 250. The reason for this is that heterogeneous characteristic is a complex problem with more local optimal solutions. For the testing cases 202 and 209, two ontologies under alignment with different lexical and linguistic characters only have the same structural characters, which makes the matching process difficult. For the testing cases 248, 249, and 250, two ontologies under alignment are highly heterogeneous, which makes alignment's quality relatively low. MOPSO-DE can achieve satisfactory results compared with the SI algorithms such as MapPSO in these cases due to the multiobjective evolution and the diversity enhancing strategy. MapPSO is a PSO-based approach, whose weights are manually set, making it easier to be trapped into local optima. However, MOPSO-DE combines multiobjective PSO and DE to avoid local optima and achieve highquality alignments.

Figure 6 compares MOPSO-DE with MOPSO on the memory consumption. The introduction of the diversity enhancing strategy updates the particles by pairwise particles at each generation instead of the local and global best particles in the archive. It can be seen that without the external archive in storing particles' information, MOPSO-DE can significantly reduce the memory consumption, which indicates that the diversity enhancing strategy can effectively reduce the computational cost. To sum up, MOPSO-DE is capable of efficiently solving the ontology metamatching problem.

TABLE 7: Comparison of MOPSO-DE with OAEI's participants in terms of f-measure.

No.	edna	AgrMaker	AROMA	ASMOV	CODI	Ef2Match	Falcon	GeRMeSMB	MapPSO	RiMOM	SOBOM	ТахоМар	MOPSO- DE
101	1.00	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
103	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
104	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
201	0.04	0.92	0.95	1.00	0.13	0.77	0.97	0.94	0.42	1.00	0.95	0.51	0.96
202	0.03	0.89	0.00	0.88	0.00	0.08	0.00	0.39	0.05	0.81	0.64	0.02	0.16
203	1.00	0.98	0.80	1.00	0.86	1.00	1.00	0.98	1.00	1.00	1.00	0.49	1.00
204	0.93	0.97	0.97	1.00	0.74	0.99	0.96	0.98	0.98	1.00	0.99	0.51	0.99
205	0.34	0.92	0.95	0.99	0.28	0.84	0.97	0.99	0.73	0.99	0.96	0.51	0.94
206	0.54	0.93	0.95	0.99	0.39	0.87	0.94	0.92	0.85	0.99	0.96	0.51	0.92
207	0.54	0.93	0.95	0.99	0.42	0.87	0.96	0.96	0.81	0.99	0.96	0.51	0.92
208	0.93	0.96	0.58	1.00	0.61	0.95	0.98	0.95	0.79	1.00	1.00	0.44	0.90
209	0.35	0.88	0.37	0.92	0.22	0.47	0.65	0.59	0.16	0.87	0.71	0.14	0.49
210	0.54	0.93	0.18	0.96	0.24	0.38	0.66	0.58	0.32	0.85	0.82	0.15	0.66
221	1.00	0.97	0.99	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
222	0.98	0.98	0.99	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	0.46	1.00
223	1.00	0.95	0.93	1.00	1.00	1.00	1.00	0.96	0.98	0.98	0.99	0.45	0.96
224	1.00	0.99	0.97	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.51	1.00
225	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
228	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.51	1.00
230	0.85	0.90	0.93	0.97	0.98	0.97	1.00	0.94	0.98	0.97	0.97	0.49	0.95
231	1.00	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.51	1.00
232	1.00	0.97	0.97	1.00	0.97	1.00	0.99	1.00	1.00	1.00	1.00	0.51	1.00
233	1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00
236	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
237	0.98	0.98	0.97	1.00	0.99	1.00	0.99	1.00	0.99	1.00	1.00	0.46	1.00
238	1.00	0.94	0.92	1.00	0.99	1.00	0.99	0.96	0.97	0.98	0.98	0.45	0.96
239	0.50	0.98	0.98	0.98	0.98	0.98	1.00	0.98	0.98	0.98	0.98	0.94	0.98
240	0.55	0.91	0.83	0.98	0.95	0.98	1.00	0.85	0.92	0.94	0.98	0.88	0.88
241	1.00	1.00	0.98	1.00	0.94	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00
246	0.50	0.98	0.97	0.98	0.98	0.98	1.00	0.98	0.98	0.98	0.95	0.94	0.98
247	0.55	0.88	0.80	0.98	0.98	0.98	1.00	0.91	0.89	0.94	0.98	0.88	0.88
248	0.03	0.72	0.00	0.87	0.00	0.02	0.00	0.37	0.05	0.64	0.48	0.02	0.09
249	0.03	0.88	0.02	0.88	0.02	0.08	0.00	0.35	0.05	0.78	0.64	0.02	0.16
250	0.02	0.56	0.00	0.62	0.00	0.11	0.00	0.06	0.03	0.73	0.20	0.00	0.25
301	0.59	0.59	0.73	0.86	0.38	0.71	0.78	0.71	0.64	0.73	0.84	0.43	0.80
302	0.43	0.32	0.35	0.73	0.59	0.71	0.71	0.41	0.04	0.73	0.74	0.40	0.76
303	0.00	0.78	0.59	0.83	0.65	0.83	0.77	0.00	0.00	0.86	0.50	0.36	0.86
304	0.83	0.86	0.84	0.95	0.74	0.95	0.94	0.77	0.72	0.94	0.91	0.52	0.96

6. Conclusions

This paper is devoted to solving the ontology metamatching problem. The cooperation between intelligent application systems needs to share the semantic information of data, and the term description of different fields leads to semantic heterogeneity. The heterogeneity issue hampers the communications among different ontologies and stops the cooperations among ontology-based intelligent applications. In order to build a semantic bridge between different application systems, ontology is needed to describe semantic relations. To solve this problem, it is necessary to find the mappings between ontology entities, that is, to perform the ontology metamatching process. The ontology metamatching problem is firstly defined as a multiobjective optimization problem in this work, and a MOPSO-DE is proposed to solve it. MOPSO-DE uses a diversity enhancing strategy to efficiently improve the alignment's quality. Diversity enhancing strategy enables the solutions to converge quickly to the Pareto fronts and to be uniformly distributed on the real PF. The experiment uses OAEI's benchmark to test MOPSO-DE's performance. Experimental results prove that MOPSO-DE can obtain the perfect alignment and reduce the MOPSO's memory consumption.

However, according to the experimental results, the alignment's quality of MOPSO-DE is not good when dealing with relatively complex datasets, such as 202, 209, 248,249, and 250. Therefore, we will further improve the diversity enhancing strategy later. The performance of MOPSO-DE in this paper is good when dealing with problems with small dimensions, so it can be considered to apply MOPSO-DE to solve large-scale problems in the future work.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province (No. 2022J01644 and 2020J01875), the National Natural Science Foundation of China (No. 62172095), the Fujian Province Young and Middle-Aged Teachers Education Research Project (No. JAT210647, JAT210651), the Industry-School Cooperative Education Program of Ministry of Education (No. 202102100004), and the Research Innovation Team of Concord University College Fujian Normal University in 2020 (No.2020-TD-001).

References

- M. Rhayem, F. Mhiri, and F. Gargouri, "Semantic web technologies for the Internet of things: systematic literature review," *Internet of Things*, vol. 11, article 100206, 2020.
- [2] G. Antoniou and F. H. Van, A Semantic Web Primer, MIT press, 2004.
- [3] S. A. Mcilraith, T. C. Son, and H. Zeng, "Semantic web services," *Intelligent Systems*, vol. 16, no. 2, pp. 46–53, 2006.
- [4] T. Bernerslee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [5] G. Acampora and A. Vitiello, "A study on local search metaheuristics for ontology alignment," *Computational Intelligence* for Semantic Knowledge Management, vol. 837, pp. 53–70, 2020.
- [6] I. Osman, S. B. Yahia, and G. Diallo, "Ontology integration: approaches and challenging issues," *nformation Fusion*, vol. 71, pp. 38–63, 2021.
- [7] J. J. Lastra-Díaz, J. Goikoetxea, M. A. H. Taieb et al., "A large reproducible benchmark of ontology-based methods and word embeddings for word similarity," *Information Systems*, vol. 96, article 101636, 2021.
- [8] N. Ferranti, J. Souza, and S. Srosáriofurtadosoares, "An experimental analysis on evolutionary ontology meta-matching," *Knowledge and Information Systems*, vol. 63, no. 11, pp. 2919–2946, 2021.

- [9] J. Martinez-Gil and J. F. Aldana-Montes, "An overview of current ontology meta-matching solutions," *The Knowledge Engineering Review*, vol. 27, no. 4, pp. 393–412, 2012.
- [10] 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.
- [11] D. Cai, W. Yuping, and Y. Miao, "A new multi-objective particle swarm optimization algorithm based on decomposition," *Information Sciences*, vol. 325, pp. 541–557, 2015.
- [12] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 874–888, 2020.
- [13] A. V. Semenova and V. M. Kureychik, "Multi-objective particle swarm optimization for ontology alignment," in 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–7, Baku, Azerbaijan, 2016.
- [14] X. Xue, "Complex ontology alignment for autonomous systems via the compact co-evolutionary brain storm optimization algorithm," *ISA Transactions*In Press.
- [15] X. Xue and J. Chao, "Matching sensor ontologies with multi-Context similarity measure and parallel compact differential evolution algorithm," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24570–24578, 2021.
- [16] X. Xue and W. Liu, "Integrating heterogeneous ontologies in Asian languages through compact genetic algorithm with annealing re-sample inheritance mechanism," *Transactions* on Asian and Low-Resource Language Information Processing, pp. 1–21, 2022.
- [17] C. Rangel, J. Aguilar, M. Cerrada, and J. Altamiranda, "An approach for the emerging ontology alignment based on the bees colonies," in *Proceedings on the International Conference* on Artificial Intelligence (ICAI), pp. 536–541, 2015.
- [18] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, article 107343, 2021.
- [19] 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.
- [20] G. Alexandru-Lucian and A. Iftene, Using a Genetic Algorithm for Optimizing the Similarity Aggregation Step in the Process of Ontology Alignment, Roedunet International Conference, 2010.
- [21] 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.
- [22] 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.
- [23] Y. He, 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.
- [24] X. Xue, J. Lu, and J. Chen, "Using NSGA-III for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.

- [25] H. Seada and K. Deb, U-NSGA-III: A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives: Proof-of-Principle Results, Springer International Publishing, 2015.
- [26] J. Wang, Z. Ding, and C. J. Jiang, "GAOM: genetic algorithm based ontology matching," in 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06), pp. 617–620, Guangzhou, China, 2006.
- [27] A. Alves, K. Revoredo, and F. Baio, Ontology Alignment Based on Instances Using Hybrid Genetic Algorithm, Citeseer, 2012.
- [28] 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.
- [29] J. Bock and J. Hettenhausen, "Discrete particle swarm optimisation for ontology alignment," *Information Sciences*, vol. 192, no. 6, pp. 152–173, 2012.
- [30] Y. Huang, X. Xue, and C. Jiang, "Semantic integration of sensor knowledge on artificial internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8815001, 8 pages, 2020.
- [31] X. Wu, J. Chen, and X. Xue, "Optimizing biomedical ontology alignment through a compact multiobjective particle swarm optimization algorithm driven by knee solution," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 4716286, 10 pages, 2020.
- [32] N. Guarino, D. Oberle, and S. Staab, "What is an ontology? Handbook on ontologies," in *Handbook on Ontologies*, International Handbooks on Information Systems, S. Staab and R. Studer, Eds., pp. 1–17, Springer, Berlin, Heidelberg, 2009.
- [33] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1994.
- [34] Z. Wu and M. Palmer, "Verb semantics and lexical selection," https://arxiv.org/abs/cmp-lg/9406033.
- [35] H. Tan and P. Lambrix, SAMBO Results for the Ontology Alignment Evaluation Initiative 2007, Research Gate, 2007.
- [36] A. Maedche and S. Staab, *Measuring Similarity between Ontologies*, Springer, Berlin Heidelberg, 2002.
- [37] J. Euzenat, "An API for ontology alignment," in *International Semantic Web Conference*, S. A. McIlraith, D. Plexousakis, and F. Harmelen, Eds., vol. 3298 of Lecture Notes in Computer Science, pp. 698–712, Springer, Berlin, Heidelberg, 2004.
- [38] V. Mascardi, A. Locoro, and P. Rosso, "Automatic ontology matching via upper ontologies: a systematic evaluation," *IEEE Transactions on Knowledge & Data Engineering*, vol. 22, no. 5, pp. 609–623, 2010.
- [39] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, and O. Zamazal, "Results of the ontology alignment evaluation initiative 2017," *CEUR Workshop Proceedings*, vol. 2032, pp. 61– 113, 2016.
- [40] I. F. Cruz, C. Stroe, F. Caimi et al., "Using AgreementMaker to align ontologies for OAEI 2010," *ISWC International Workshop on Ontology Matching (OM)*, vol. 814, pp. 114–121, 2010.
- [41] J. David, "AROMA results for OAEI 2011," Ontology Matching, vol. 122, 2011.
- [42] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Journal of Web Semantics*, vol. 7, no. 3, pp. 235–251, 2009.
- [43] J. Huber, T. Sztyler, J. Noessner, and C. Meilicke, "CODI: combinatorial optimization for data integration-results for OAEI 2011," *Ontology Matching*, vol. 134, 2010.

- [44] W. Wei, K. Chua, and J. Kim, "Eff2Match results for OAEI 2010," Ontology Matching, vol. 150, pp. 105-106, 2010.
- [45] H. Wei, J. Chen, C. Gong, and Y. Qu, "ObjectCoref and Falcon-AO: results for OAEI 2010," *Ontology Matching*, vol. 2010, 2010.
- [46] C. Quix, A. Gal, T. Sagi, and D. Kensche, "An integrated matching system: GeRoMeSuite and SMB-results for OAEI 2010," Ontology Matching, vol. 166, 2010.
- [47] J. Euzenat and P. Shvaiko, *Overview of Matching Systems*, Springer, Berlin Heidelberg, 2013.
- [48] Z. Wang, Z. Xiao, H. Lei, Z. Yue, and T. Jie, "RiMOM results for OAEI 2010," *Ontology Matching*, vol. 195, 2010.
- [49] P. Xu, Y. Wang, C. Liang, and T. Zang, "Alignment results of SOBOM for OAEI 2010," Ontology Matching, vol. 203, pp. 7–11, 2010.
- [50] A. Xe, B. Hamdi, N. Safar, B. Niraula, and C. Reynaud, "Taxo-Map alignment and refinement modules: results for OAEI 2010," *Ontology Matching*, vol. 212, 2010.