Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **3**, August **2023**

Knee Solution-Driven, Decomposition-Dased Multi-Objective Particle Swarm Optimization for Ontology Meta-Matching

Wen-Bin Tan

College of Electrical and Power Engineering
Taiyuan University of Technology
Yingjersey Street, Wanblin District
wbtan_21@163.com

Qing Lv*

College of Electrical and Power Engineering
Taiyuan University of Technology
Yingjersey Street, Wanblin District
lvqing_2006@163.com

Bao-Zhong Zhao

College of Electrical and Power Engineering
Taiyuan University of Technology
Yingjersey Street, Wanblin District
zbz892950125@163.com

Qi Wu

College of Electrical and Power Engineering
Taiyuan University of Technology
Yingjersey Street, Wanblin District
w19985757@163.com

Yi-Kun Huang

Concord University College of Fujian Normal University
Fujian Normal University
Cangshan District
fjnuhyk@163.com

*Corresponding author: Qing Lv

Received March 6, 2023, revised April 26, 2023, accepted June 14, 2023.

Abstract. *As the latest information exchange model, ontology is favored by information systems, but due to the different backgrounds of ontology engineers, heterogeneity often arises between different ontologies, and these heterogeneity seriously affect the interaction and cooperation between these systems. but due to the different backgrounds of ontology engineers, heterogeneity often arises between different ontologies, and these heterogeneity seriously affect the interaction and cooperation between these systems. Ontology matching is considered an effective method to solve the ontology heterogeneity problem whose kernel technology is a similarity measure. Since there are three aspects of ontology heterogeneity, i.e. lexical-based, linguistics-based and structure-based*, a single measure cannot achieve satisfactory ontology alignments. To this end, integrating different similarity measures is necessary. First of all, due to the difference in user preferences for alignment quality, the ontology matching problem is modeled as a continuous multi-objective optimization model. Particle Swarm Optimization (PSO) is suitable for solving continuous optimization problems and previous studies have found that decomposition-based methods are more suitable for solving ontology matching. Then, considering the user's preference, a knee solution-driven, decomposition-based multi-objective particle swarm algorithm (K-MOPSO/D) is designed to solve the ontology matching. Finally, the effectiveness of our proposed method is verified by standard test cases from the well-known OAEI (Ontology Alignment Evaluation Initiative), and its performance is compared with the state-of-the-art matching methods.

Keywords: ontology; PSO; multi-objective optimization; user preference; OAEI.

1. **Introduction.**

As the kernel technology of Semantic Web (SW), ontology plays an important role. At present, ontology, the formal expression of information, has been widely used in various fields such as Artificial Intelligence (AI) [1], Semantic Sensor Web (SSW) [2, 3], and Biomedical [4, 5]. Due to the subjectivity of different ontology designers [6] and different representations of the same domain knowledge (e.g., different terms for the same concept in different ontologies [7]), there is often heterogeneity among ontologies. Performing ontology matching is an effective solution to realize semantic interaction between two ontologies [8]. Figure 1 shows the result of performing ontology matching for two heterogeneous ontologies. Matched pairs of concepts are connected by arrows, e.g. "chairman" and "chair".

Since manual execution of ontology matching is time-consuming and has a high error rate [9], automatic matching systems have shown great potential [10]. The core of ontology matching is the similarity measure technique, which measures the similarity between different ontology entities. However, since there are three heterogeneous cases of ontology entities, namely lexical heterogeneity, semantic heterogeneity, and structural heterogeneity, a single similarity measure cannot truly reflect the similarity between entities [11], and it is imperative to integrate multiple similarity measures. Therefore, how to find appropriate similarity measures, integration weights, and filtering thresholds to obtain high-quality ontology alignment is called the ontology meta-matching problem [12]. Figure 2 illustrates the optimization model for ontology meta-matching. Firstly, the information about ontologies is analyzed by Jena [1] (a tool of ontology), and multiple similarity matrices($M_i(i = 1, 2, ..., n)$) are obtained by different similarity measures, secondly, multiple similarity matrices are aggregated into one(M) by weighted sum, and finally, the final similarity matrix(M_T) is obtained by threshold(T) filtering, and matching result is extracted.

Since the evaluation indicators [13](recall and precision) of ontology matching, are contradictory to a certain extent, it is difficult to find a perfect solution that can make both

[1]https://jena.apache.org/

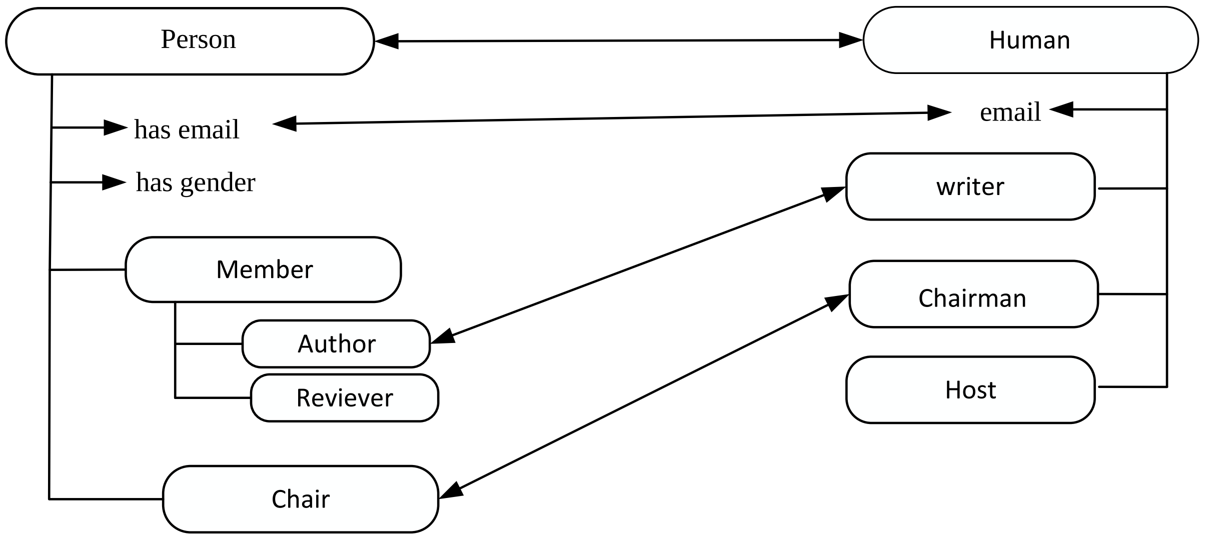

Figure 1. The result of performing ontology matching for ontologies with heterogeneity.

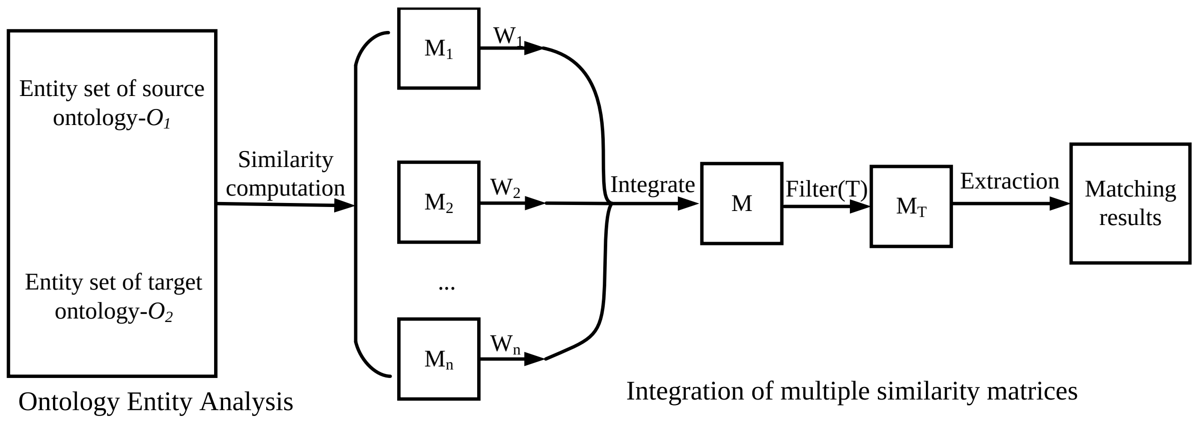

Figure 2. The optimization model for ontology meta-matching

recall and precision optimal, so we model the ontology matching problem as a multi-objective continuous optimization problem, and use the Pareto front as a set of trade-off solutions to satisfy the needs of different users. It is worth noting that the knee solution is the largest trade-off between the objectives. A small improvement in any of the objectives of the knee solution will be a large deterioration of the objectives. Therefore, in this paper, the knee solutions are used as the output of the model to satisfy different users' preferences.

Many ontology matching systems rely on reference alignment to evaluate the quality of the ontology alignment, which is called ontology matching with reference alignment. However, obtaining expert reference matching is a very difficult task, especially for large-scale ontology. To solve this problem, we propose three approximate evaluation functions as evaluation metrics, which is called ontology matching without reference alignment.

Considering that the optimization objectives recall and precision of the ontology meta-matching problem have no gradient information, the meta-heuristic algorithm has attracted widespread attention. Particle swarm optimization algorithm [14] is one of the representative algorithms of meta-heuristics, which is a global stochastic optimization algorithm based on population intelligence. It imitates the foraging behavior of birds, analogizes the search space of the problem to the flight space of birds, considers each bird as a particle to represent a candidate solution to the problem, and the optimal solution to be found is equivalent to the food to be found. Since the particle swarm optimization algorithm is highly parallel and converges quickly, it is well suited to solve the ontology meta-matching problem.

In the past decades, there are two main types of strategies to deal with multi-objective optimization problems, namely, Pareto-based algorithms, and decomposition-based algorithms. Regarding the Pareto-based algorithms, there are often these two drawbacks [15]: (1) the performance of the algorithms is very sensitive to the parameter settings. (2) In the iterative update, the convergence priority criterion is used to determine the updated position of the particles, which often leads to poor population diversity. Regarding the decomposition-based algorithms, the multi-objective problem is decomposed into several sub-problems, and all sub-problems are optimized simultaneously using the neighborhood relationship between different sub-problems. Therefore, the decomposition-based algorithms all find a finite number of Pareto optimal vectors that are uniformly distributed along the PF and thus well represent the whole Pareto Front(PF) [16]. In addition, since each sub-problem is optimized by using information from only a few of its neighboring sub-problems, this makes the decomposition-based algorithms have lower computational complexity than the Pareto-based algorithms. Therefore, this paper proposes a knee solution-driven, decomposition-based multi-objective particle swarm optimization algorithm(K-MOPSO/D) to handle the ontology element matching problem. To facilitate the presentation of the main contributions of this paper, they are summarized as follows:

- The ontology matching problem is converted into the multi-objective optimization model.
- Three approximate evaluation functions are proposed as the objective function in the optimization model to overcome the drawback of introducing reference alignment.
- The K-MOPSO/D is used to solve the multi-objective optimization model.

The rest of the paper is organized as follows: Section 2 describes the background of ontology matching; Section 3 describes the concepts related to ontology matching; Section 4 describes the details of K-MOPSO; Section 5 presents the experimental configuration and results; finally, Section 6 concludes and gives future work.

2. **Related Work.**

Ontology matching, as the core technology for interaction between different ontologies, can be divided into three main categories, namely, manual matching systems, semi-automatic matching systems, and automatic matching systems. Since manual processing of ontology matching is very time-consuming and labor-intensive, semi-automatic or automatic matching systems are generally used at this stage.

2.1. **Semi-Automatic Matching System.**

As the core technology of semi-automatic or automatic ontology matching, the similarity measures are used to determine the confidence of similarity between two entities in the ontology. The similarity value indirectly reflects the confidence of matching pairs. In general, using only one similarity measure cannot yield satisfactory results [17]. Mult-similarity measures are integrated into various most notable matching systems, e.g. COMA [18], COMA++ [19], QuickMig [20], OntoBuilder [21], and MapPSO [22], but these matching systems require experts to provide weights in advance to obtain matching results. Therefore, they belong to the semi-automatic system. These systems are automated only to a certain extent, but when the heterogeneity of ontology is more complex, it is difficult to assign weights to each similarity measure.

2.2. **Automatic Matching System.** Different from the semi-automatic method, the automatic ontology matching technique set the weights of the similarity measures through some intelligent techniques (e.g. meta-heuristic algorithm, machine learning). For example, GOAL [23] obtains the weights of different similarity measures indirectly through the genetic algorithm (GA), and then these similarity measures were integrated to find out suitable matching pairs. In this process, the integration of weights is automated and does not require the involvement of experts. Similarly, GA is used by GAOM [24], but it only takes into account classes and not properties in the ontology. However, the system, GOAL, is not perfect, because the reference alignment (RA) which is the standard answer obtained by an expert is introduced into GOAL in advance. Therefore, the method of GOAL is more theoretical value but lacks realistic significance. The bold attempt of GOAL overcomes some defects of semi-automatic matching, but a new challenge arises, which is how to overcome the defects of introducing reference alignment. To overcome the issue, the matching system with partial reference alignment (PRA) was proposed. Partial reference alignments need to be provided in advance as training samples in the PRA-based system, where the sample data reflect the overall character of the ontology. The most famous of the PRA-based systems is SAMBO [25], which uses some of the results as anchors to filter out incorrect matching pairs. The PRA is also used in LSD [26], in which a group of learners is trained by machine learning to get all the matching pairs. In recent years, many researchers have developed ontology matching systems based on PRA, such as Xue [27], etc. However, the current PRA technology is still not mature enough, because it is very difficult to find a group of representative samples reasonably, especially when the ontology has a large scale.

To overcome the drawbacks of overcoming the introduction of reference alignment, Ontology matching with no reference alignment(NRA) has achieved many results. Depending on the number of optimization objectives, NRA can also be divided into two categories, i.e., single-objective no-reference alignment(SO-NRA) and multi-objective no-reference alignment(MO-NRA). Concerning SO-NRA, Lv [28] designs an approximate evaluation function as the object to gain the matching pairs. similar works have also been mentioned in Xue [29] using firefly algorithm, Lv [30] using grasshopper optimization, and Jiang [31] using genetic algorithm, et. al. Various heuristic algorithms are used in matching methods and strive to balance the exploration and exploitation performance of algorithms in the process of ontology matching. Regarding MO-NRA, Acampora et al. [32] and Xue et al. [33] modeled the ontology matching problem as a multi-objective optimization problem for the first time and obtained better results than traditional EA [34] matching using the NSGA-II strategy [35], in addition, Xue et al. [36] proposed using a decomposition-based strategy to solve the ontology matching problem, and the results showed that the decomposition-based strategy was more suitable than NSGA-II to solve the ontology matching problem. However, the presence of evaluation errors in NRA can affect the performance of the algorithm.

Unlike evolutionary algorithm(EA) [37], PSO boasts the characteristics of fast searching speed, high efficiency, and simplicity, and is very suitable for real-valued continuous optimization problems (i.e. ontology matching) [38]. There are several potencies of the implementation of PSO in ontology matching, e.g. processing the large input, providing a general framework, scaling parallel computing, and having flexible anytime behavior [22]. In this paper, first, we model the ontology matching problem as a continuous multi-objective optimization problem. Second, three functions are proposed to evaluate the ontology alignment quality. Finally, we propose a knee solution-driven, decomposition-based multi-objective particle swarm optimization algorithm to solve this continuous multi-objective optimization problem.

3. Preliminaries.

3.1. Ontology and Ontology Alignment.

The word, "ontology", originates from the field of philosophy and is defined as the essence and structure of things. With the development of computer technology, ontology is used to describe the knowledge in the computer field [39]. One of the most authoritative explanations, proposed by Gruber in 1993, is that ontology is a "formalized, explicit and detailed explanation of shared concepts" [40]. Ontology provides a shared vocabulary, the existing object types or concepts and properties and interrelationships, which are structured and therefore suitable for use in computer systems. For the convenience of work in the following, ontologies are defined as follows [41]: **Definition 1 (Ontology)**: Ontology O is defined as a triple $O = (C, P, I)$, where:

- C denotes the set of classes or concepts, in which each class is the abstract expression of objects.
- P denotes the set of properties or relationships, in which each property describes the relationship between the domain and range.
- I denotes the set of instances, in which each instance describes the concrete object in the real world.

Definition 2 (Ontology Matching Result): ontology matching result A' is the set of matching elements, in which each element is 5-tuple (id, e_1, e_2, r, c). where:

- id is the identifier of the matched pair.
- e_1 and e_2 are the entities of different ontologies respectively.
- r is the relationship between e_1 and e_2 (usually equivalent).
- c is the confidence the matched pair ($c \in [0, 1]$).

3.2. Similarity Measures.

3.2.1. *lexical-based similarity measures.*

The similarity measures based on lexical is used to calculate the morphological similarity of the textual contents. Generally speaking, this kind of measure is realized by such operations: adding, deleting, and replacing. Several lexical-based similarity measures have emerged, such as N-gram, SMOA, and Levenshtein. According to these papers [42, 43], N-gram and SMOA have excellent performance in solving ontology matching problems, so these methods are used in this paper. The calculation of N-gram is shown as follows:

$$\text{N-gram}(s_1, s_2) = \frac{2 \times comm(s_1, s_2)}{N_{s_1} + N_{s_2}} \tag{1}$$

where s_1 and s_2 are two strings to be compared, each of them is cut into several substrings, in which three characters form a group. $comm(s_1, s_2)$ represents the number of identical substrings between s_1 and s_2. N_{s_1} and N_{s_2} represent the number of substrings of s_1 and s_2 respectively.

Unlike N-gram, SMOA [44] takes into account not only the same part of the two strings but also the differences between the two strings, which is defined as follows:

$$SMOA(s_1, s_2) = com(s_1, s_2) - dif(s_1, s_2) + winklerImpr(s_1, s_2) \tag{2}$$

where $com(s_1, s_2)$ is the number of largest public string between s_1 and s_2 by recursive way, and the largest public string will be removed from s_1 and s_2 respectively everytime the largest public substring is found. The $dif(s_1, s_2)$ is based on the length of the substring that did not match of $com(s_1, s_2)$ in the first iteration. $winklerImpr(s_1, s_2)$ is the improvement method proposed by Winkler.

3.2.2. *Linguistic-based similarity measure.* The linguistic-based similarity measure is used to determine the semantic distance between hypernym and hyponym. The introduction of external resources is required in this measure. This Wup measure [45], based on WordNet (an electronic dictionary widely used in the field of NLP), is proposed by Wu and Palmer and widely used in ontology matching [46]. It is based on the principle that the closer the semantic depth of two entities in the WordNet dictionary is to their common parent, the more similar the two entities are. The definition of this method is as follows:

$$Wup(s_1, s_2) = \frac{2 \times depth(LCA(s_1, s_2))}{depth(s_1) + depth(s_2)} \tag{3}$$

where $LCA(s_1, s_2)$ is the closest common parent concept between the s_1 and s_2, and $depth(s_i)$ represent the depth of the hierarchy of s_i in WordNet.

3.2.3. *Structure-based similarity measure.* The structure-based matcher is used to compute the distance between two entities from the structural level. There are two main ideas of the structure-based similarity measure we designed: (1) if the two entities have similar kinship(father and children), the two entities are similar; (2) if the two entities have similar hierarchical relationships (tree diagrams) respectively, the two entities are similar as well.

The similarity score of $kinSim(e_1, e_2)$, shown in Equation (4), is obtained by considering the relationship between the sub-entity and the super-entity. The similarity, $fatSim(e_1, e_2)$, is expressed as the similarity between the respective super-entities, and $sonSim(e_1, e_2)$ is about sub-entities.

$$kinSim(e_1, e_2) = \frac{fatSim(e_1, e_2) + sonSim(e_1, e_2)}{2} \tag{4}$$

Considering the feature of single inheritance in ontology, that is, each entity has only one super-entity. To get the similarity with high confidence between the two super-entities, the average of the similarity measures (i.e. N-gram, SMOA, and Wup) is used. $fatSim(e_1, e_2)$ is similarity between the two super-entities from e_1 and e_2 respectively.

Given that there are multiple sub-entities of an entity, the Algorithm 1 is used to compute the similarity of sub-entity. Where, the score with perfect matching in this algorithm is the highest average of Ngram, SMOA, and Wup.

The $hieSim(e_1, e_2)$ is based on the consistency of the tree structure. If two entities have the same tree structure, the $hieSim(e_1, e_2)$ is equal to 1, otherwise to 0. The calculation is shown in Equation (5):

$$hieSim(e_1, e_2) = \begin{cases} 1 & \text{if entities } e_1,\ e_2 \text{ have the same tree structure} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

As shown in Figure 3, “tree 1” and “tree 2” have the same number of nodes both parent and child, so the two trees are consistent in the point of form. Inversely, tree2 and tree3 have the same number of parent nodes, but the number of child nodes is different, so there is no structure consistency between the two trees.

Due to the fact that class and property have different textual contents, the calculations for the class and property are different. The specific calculation process of class (as a type of entity) is as follows:

$$CstrSim(e_1, e_2) = \frac{kinSim(e_1, e_2) + hieSim(e_1, e_2)}{2} \tag{6}$$

Algorithm 1 Calculating similarity between sub-entities

```
Input: $E_1$: the set of sub-entities of $e_1$; $E_2$: the set of sub-entities of $e_2$;
Output: $sonSim(e_1, e_2)$
1: $n$ is the number of sub-entities of entity $e_1$(i.e. $|E_1|$)
2: $m$ is the number of sub-entities of entity $e_2$(i,e, $|E_2|$)
3: $sim_1 = 0$;
4: for (int $i = 1$; $i \leq n$; $i++$) do
5:     $max_1$=the score with the perfect matching between the $i$th sub-entity of $e_1$ and one entity from $E_2$;
6:     $sim_1 += max_1$;
7: end for
8: $simAverage_1$=$sim_1/n$
9: $sim_2 = 0$;
10: for (int $j = 1$; $j \leq n$; $j++$) do
11:     $max_2$=the score with the perfect matching between the $j$th sub-entity of $e_1$ and one entity from $E_1$;
12:     $sim_2 += max_2$;
13: end for
14: $simAverage_2$=$sim_2/n$
15: return $(simAverage_1 + simAverage_2)/2$
```

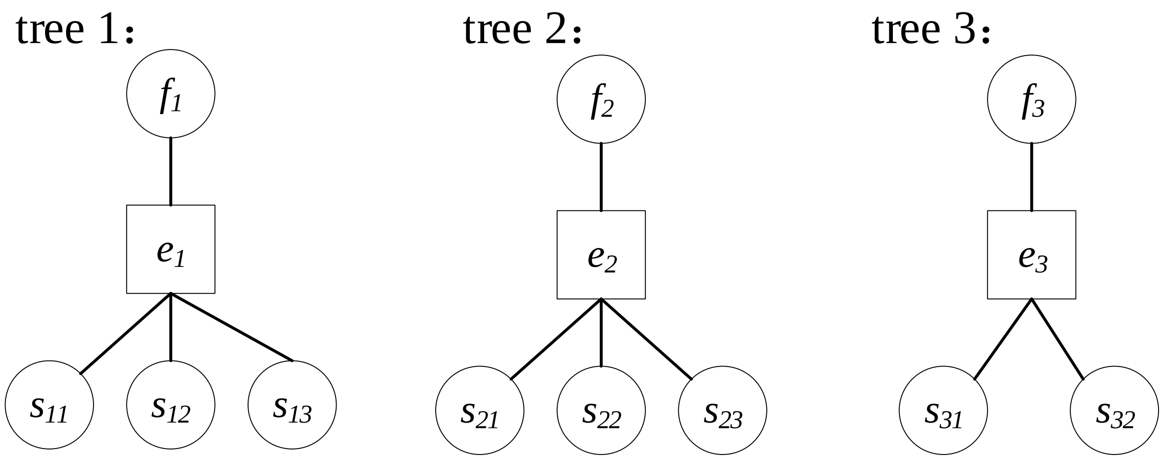

FIGURE 3. The consistency of tree structure

$StrSimC(e_1, e_2)$, a class-based structural similarity function, is composed of two parts. In the first part, the similarity of relatives is considered between two entities, i.e. $kinSim(e_1, e_2)$, and in the second part, the tree structure-$hieSim(e_1, e_2)$, is considered.

Besides these textual contents about the class entity, domain and range are used in property entity. The formula for calculating the similarity score of property entity is shown as follows:

$$PstrSim(e_1, e_2) = \frac{kinSim(e_1, e_2) + hieSim(e_1, e_2) + drSim(e_1, e_2)}{3} \tag{7}$$

The property entities also use the N-gram, SMOA, and Wup methods for calculating domain and range similarity. The similarity score of $drSim(e_1, e_2)$ is calculated as follows:

$$drSim(e_1, e_2) = \frac{domainSim(e_1, e_2) + rangeSim(e_1, e_2)}{2} \tag{8}$$

3.3. **The Multi-Object Optimization Model for Ontology Matching.** The establishment of an optimization model is a pre-requisition of addressing ontology matching. Generally, a complete optimization model consists of three vital parts: decision variables, objective function, and constraints.

3.3.1. *Decision Variables.* Usually, the decision variable is a set of unknowns and is denoted by n-dimensional vector $x = (x_1, x_2, ..., x_n)^T$, which is the solution to the problem. As shown in Figure 2, there are n weighting factors(i.e. $w_1, w_2, ..., w_n$) and one threshold T, forming decision variables $X = (x_1, x_2, ..., x_n, x_{n+1})^T$.

3.3.2. *Objective Function.* The objective function is a mathematical expression that needs to be optimized. It is a function of the decision variable X, which can be recorded as $f(X)$.

In the field of ontology matching, traditional systems need to introduce reference alignment in advance to optimize a set of metrics, namely recall, precision, and f-measure. The relevant evaluation metrics are shown as follows:

$$recall = \frac{|R \cap A|}{|R|} \tag{9}$$

$$precision = \frac{|R \cap A|}{|A|} \tag{10}$$

$$f - measure = \frac{recall \times precision}{\alpha \times recall + (1 - \alpha) \times precision} \tag{11}$$

Where R is a set of reference alignment that is usually the matching systems want to obtain and A is a set of found results. The function recall reflects the completeness of the found matching results, that is, the ratio of the correct matching results found by the matching system to the reference alignment. The function precision is marked as the accuracy of the found matching pairs, that is, the ratio of the correct found matching pairs to all the found matching results. The function f-measure is the summed average of recall and precision, usually α equal to 0.5.

But the three idealized quality metrics (i.e. recall, precision, f-measure) have some defects that the reference alignment R can not be acquired before the matching [47]. Therefore, the approximate evaluation functions are urgently needed without the introduction of set R. Three approximate evaluation functions(i.e. Equations (14), (15), and (17)) designed in our work are aimed to overcome the drawbacks of the traditional systems based on RA or PRA.

The function $comple(M)$ consists of two parts, i.e. $w(M)$ and $s(M)$. The purpose of the function $w(M)$, shown as Equation (12), is to find more matching pairs. Due to matching with one by one to be solved in this paper, the number of matching pairs found will not be greater than the number of entities (i.e. $|O_1|$ and $|O_2|$) in two ontologies respectively. The denominator of the function $w(M)$ is the minimum number of $|O_1|$ and $|O_2|$, and the numerator $|P|$ is the number of matching pairs that satisfy the condition of the threshold T.

$$w(M) = \frac{|P|}{min(|O_1|, |O_2|)} \tag{12}$$

The Algorithm 2 shows how to find the set P iteratively which is used to store the similarity scores.

Algorithm 2 The algorithm for finding set P

```
Input: the integrated matrix M, threshold $T$;
Output: $P$
1: Initialize $P = \emptyset$ , max = the largest element of matrix M;
2: while ($max > T$) do
3:     add $M_{ij}$ to $P$;
4:     set all the elements to 0 of the rows and columns corresponding to the $M_{ij}$;
5:     $max$ =the largest element of matrix M;
6: end while
7: return $P$;
```

Unlike the $w(M)$, $s(M)$ is used to select matching pairs with high confidence. The numerator of $s(M)$ denotes the sum of the similarity values of the matched pairs found, and the denominator denotes the number of matched pairs found. To better estimate the recall and to consider not making the found matching pairs perform particularly poorly in terms of precision, $s(M)$ is given a small weight and $w(M)$ is given a larger weight.

$$s(M) = \frac{\sum_{i=1}^{|P|} P_i}{|P|} \tag{13}$$

$$comple(M) = \frac{w(M) \times s(M)}{0.8 \times w(M) + 0.2 \times s(M)} \tag{14}$$

The function $sail(M)$ is used to approximate the precision metric.

$$sail(M) = \frac{d(M) \times w(M)}{0.8 \times d(M) + 0.2 \times w(M)} \tag{15}$$

where, $sali(M)$ is made up of two parts, i.e. $d(M)$ and $w(M)$. The function, $d(M)$, is used to calculate the dispersion of similarity between the most prominent matching pair and the other with lower confidence. Besides, the function $w(M)$ is integrated to make it possible to find as many matching pairs as possible with higher similarity scores.

The function $d(M)$ is as Equation (16), which is based on the principle that the more obvious the gap of similarity between "host" and "neighbors", the more advantage the "host" has. Where, "host" is the largest element in the matrix, "neighbors_*1*" is the set of all the elements in host's column except "host", while "neighbors_*2*" is in host's row.

A simple example, as shown in Figure 4, will serve to illustrate the point. A *5×5* similarity matrix is shown, and there is a "host" in the center of the matrix. All the elements except "host" itself in the third row and third column of the matrix are neighbors of the "host". The larger the difference between the scores of the "host" and those of the two groups of neighbors (i.e. neighbors_*1* and neighbors_*2*), the more superior the host is, and therefore the higher confidence it is.

$$d(M) = \frac{\sum_{i=1}^{|P|} f(P_i)}{|P|} \tag{16}$$

The algorithm of calculating P_i is shown in the Algorithm 3:

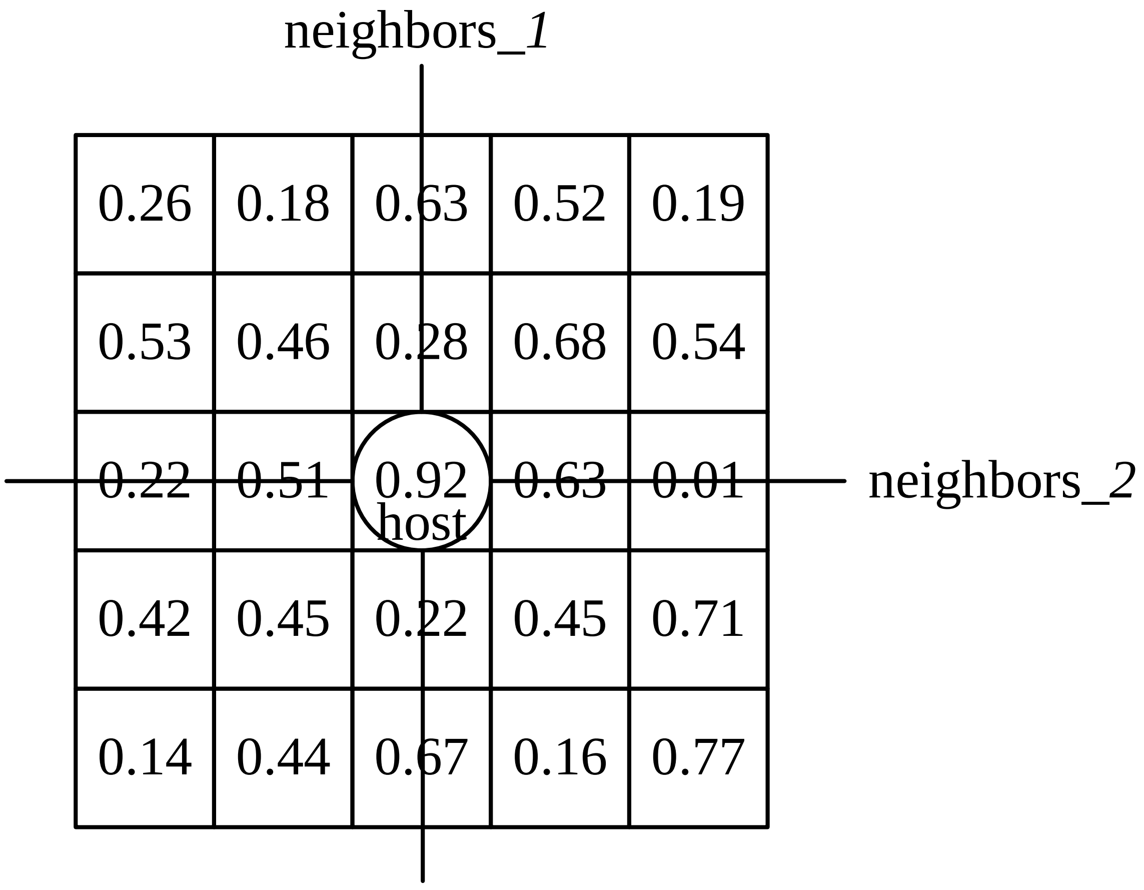

FIGURE 4. The advantage of the host

Algorithm 3 The algorithm for calculating

```
Input: similarity matrix $M$, the element $P_i$ , $L = \emptyset$;
Output: $f(P_i)$
1: Find the elements of $i$th row and $j$th column corresponding to $P_i$ in $M$, and add to
   $L$(don't include $P_i$)
2: double $d = 0.0$;
3: for (int $j = 1$; $j \leq |L|$; $j++$) do
4:     $d+ = (P_i - L_i)$;
5: end for
6: return $d/|L|$;
```

To approximate f-measure, a weighted average function $bal(M)$ is proposed, which is used to balance $comple(M)$ and $sali(M)$. It is defined as Equation (17):

$$bal(M) = \frac{2 \times comple(M) \times sail(M)}{comple(M) + sail(M)} \tag{17}$$

3.3.3. *Constraint Condition.* The constraints in an optimization model are determined by the decision variables $x(x \in \Omega)$, where Ω is the feasible region. A set of equations like $h_i(x) = 0(i = 1, 2, ..., m)$, and inequality, $g_j(x) \leq 0(j = 1, 2, ..., n)$, are respectively used to define the constraints. In the optimization model established in this paper, the weights $(w_1 \sim w_n)$ and thresholds T need to be restricted.

- Equation constraint: The n similarity measures used are required to be given weight respectively, so the sum of the n weighting factors is required to be 1.0, i.e., $\sum_{i=1}^{n} w_i = 1.0$.
- Inequality constraint: According to the characteristics of the weighting factors of similarity measures, the range of weight is $w_i \in [0,1](i = 1, ..., n)$. Due to similarity values in the $[0,1]$, the threshold should be satisfied $T \in [0,1]$.

4. Knee Solution Driven, Decomposition-Dased Multi-Objective Particle Swarm Algorithm.

Combined with the multi-objective optimization model, the strategy of selecting three representative solutions is designed to meet the user's preference. To solve this model, the novel decomposition based multi-objective PSO algorithm is designed.

4.1. Encoding and Decoding.

Considering the features of decision variables and constraints in the optimization model, the decision variables should be encoded and decoded rightly. Since real number encoding does not lose accuracy and is easier than binary encoding, we use real number encoding. In this work, the coding information includes the weights used to integrate similarity measures and the thresholds used to filter low-confidence pairs. Specifically, based on the characteristics of the weights in the optimization model, we replace the weights with breakpoints, which reduces the number of coding dimensions.

Suppose, n is the number of weighting factors, the set of these breakpoints is represented as $s = \{s_1, s_2, ..., s_{(n-1)}\}$. The first step of the decoding process is to sort the elements of s into a set $s' = \{s'_1, s'_2, ..., s'_{n-1}\}$ by ascending order, and then calculate these weights through the following Equation (18):

$$w_m = \begin{cases} s'_1, & m = 1 \\ s'_m - s'_{m-1}, & 1 < m < n \\ 1 - s'_{m-1}, & m = n \end{cases} \tag{18}$$

Due to the need for $n-1$ bit representing breakpoints and one-bit representing thresholds, the encoding length of an individual is n. Figure 5 shows the decoding process of a complete individual. Where s_1, s_2, and s_3 are the three breakpoints, and t is the threshold. The three breakpoints are sorted in ascending order (i.e. s'_1, s'_2, s'_3) except threshold t. Using this coding method not only satisfies the constraints in the optimization model but also reduces one dimension of encoding.

4.2. Traditional PSO algorithm.

In an N-dimension space, the information about particle i can be represented by two N-dimension vectors: the position of the particle i, i.e. $x_i = (x_{i1}, x_{i2}, ..., x_{iN})^T$, and its velocity, $v_i = (v_{i1}, v_{i2}, ..., v_{iN})^T$. After finding two optimal solutions, the global and local optimal solutions, the velocity and position particle update according to the Equation (19):

$$\begin{gathered} v_i^{k+1} = \omega v_i^k + c_1 \times rand_1 \times (Pbest_i^k - x_i^k) + c_2 \times rand_2 \times (Gbest^k - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{gathered} \tag{19}$$

where:

- i: is one particle in population;
- k: is iterations;
- ω: is the inertia factor, a non-negative number, which is used to controls the particle's speed;
- c_1 and c_2: learning factors, appropriate c_1, and c_2 can accelerate convergence and not easily fall into local optimization;

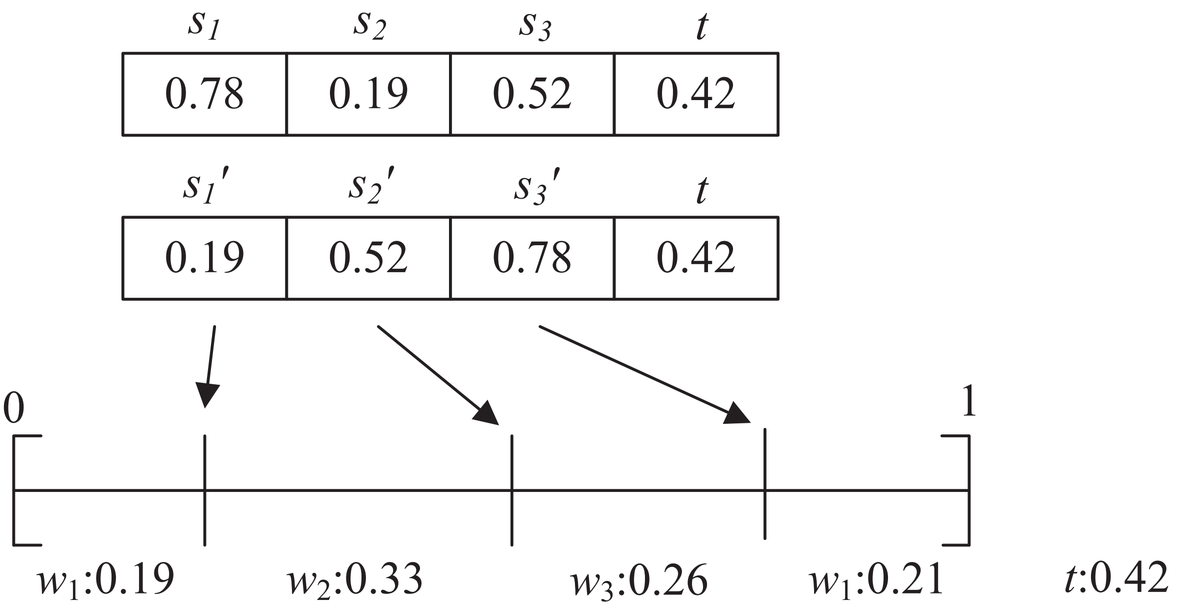

Figure 5. The encoding and decoding

- $rand_1$ and $rand_2$: are two random numbers between $[0, 1]$ respectively;
- $Pbest_i^k$: is the position of extremum of particle i;
- $Gbest^k$: is the position of extremum of population;
- v_{max}: determines the intensity of searching the problem space and the v_i^k is limited between $[-v_{max}, +v_{max}]$.

In this paper, the parameters are determined according to Shi and Eberhart's suggestions to balance exploration and exploitation [48]. The inertia weight ω to be changed linearly with the number of iterations as Equation (20):

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \times k_n \tag{20}$$

Where $\omega_{max} = 0.9$, $\omega_{min} = 0.4$, k_{max} is the maximum iterations, k_n is the number of current iterations. This LDW(Linearly Decreasing Inertia Weight) allows the algorithm to explore a larger area at the beginning and locate the approximate position quickly. The basic flowchart of the PSO algorithm is shown as Figure 6:

However, PSO algorithm is designed for the single objective optimization problem, it is not suitable for the optimization model proposed in this paper. In order to improve the quality of solutions and provide a variety of different non-dominated solutions at one time, a decomposition based multi-objective PSO algorithm (MOPSO/D) is proposed for solving the ontology matching problem.

4.3. **The Decomposing for Multi-Objective PSO.**

In the multi-objective optimization algorithm, the process of fast non-dominated sorting and crowded distance estimation is time and space-consuming [49]. However, it is necessary to determine the non-dominated solution set of multi-objective optimization. Under the decomposition strategy, the multi-objective optimization problem is decomposed into a series of single-objective optimization subproblems, and each Pareto optimal solution is the corresponding single objective optimization subproblem. In this paper, the decomposition strategy is used to transform the traditional PSO into a multi-objective algorithm.

In this paper, the idea of decomposing is used to select three representative solutions for users with different preferences. The N subproblems are defined as this form:

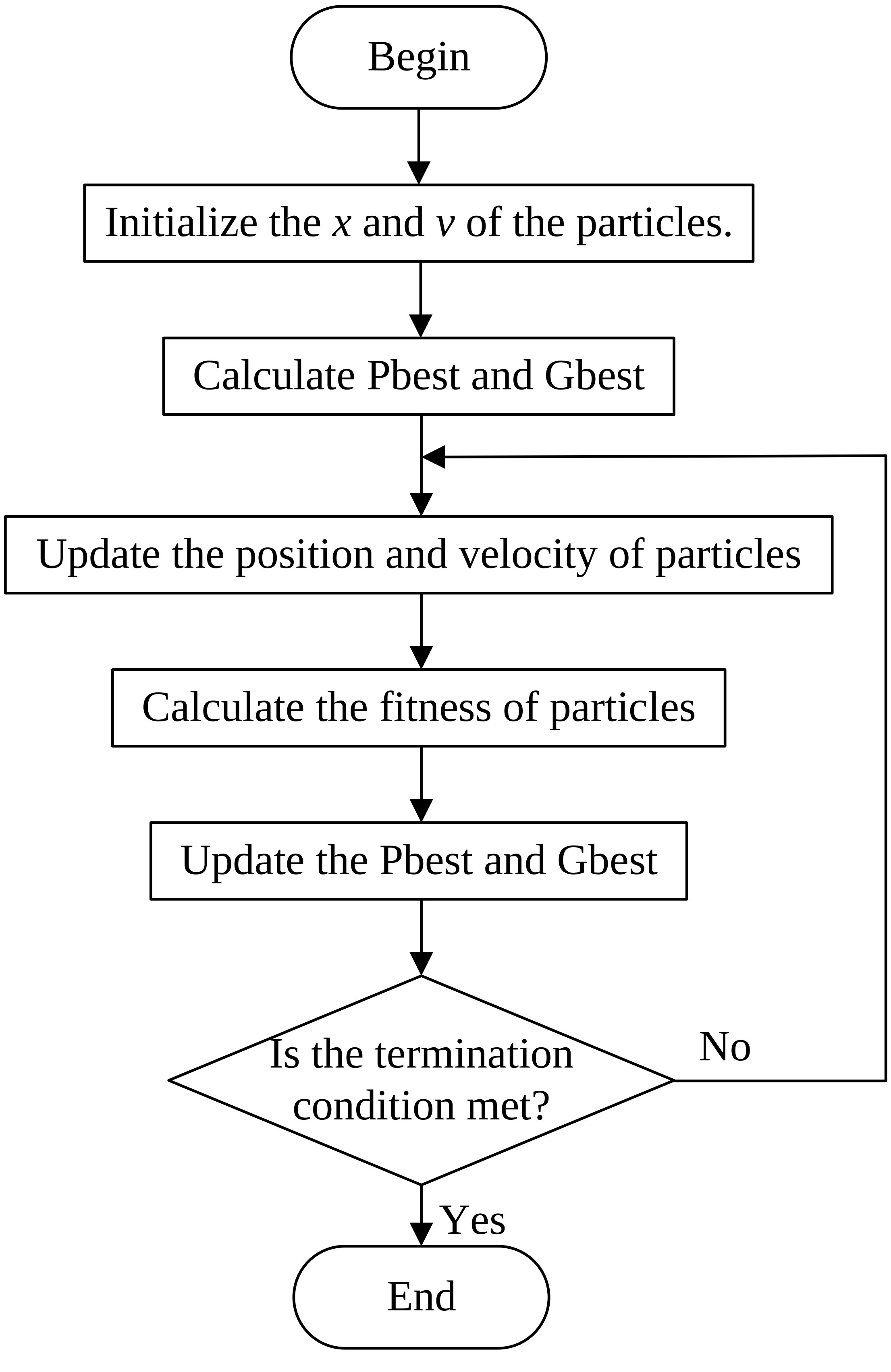
Begin
Initialize the x and v of the particles.
Calculate Pbest and Gbest
Update the position and velocity of particles
Calculate the fitness of particles
Update the Pbest and Gbest
Is the termination condition met?
No
Yes
End

Figure 6. The flow chart of PSO

$\frac{comple\times sail}{\alpha_i\times comple+\beta_i\times sail}$. The computation of $(\alpha_i,\beta_i)^T$ is as Equation (21):

$$(\alpha_i,\beta_j)^T=\begin{cases}(0,1)^T, & i=1\\ (\frac{i-1}{N-1},\frac{N-i}{N-1})^T, & other\end{cases} \tag{21}$$

The Euclidean distance between the two weight vectors $\lambda_m=(\alpha_m,\beta_m)$ and $\lambda_n=(\alpha_n,\beta_n)(m\neq n)$ is used to judge the closeness between two weight vectors. If λ_m and λ_n have a closer Euclidean distance, the two sub-problems corresponding to the weight vectors are neighbor problems, which can promote the optimization process of each other. The pseudo-code for the MOPSO/D algorithm is shown algorithm 4:

Algorithm 4 The pseudo-code of MOPSO/D

```
Input: MOP, termination condition, $N$-the number of a subpopulation, $m$-the number
    of individuals in each subpopulation, $T$-the number of neighbor vectors.
Output: Three representative solution collections: $EP$.
 1: Initialization:
 2: Step1: generate $N\times m$ individuals randomly;$\{x^1,...,x^N\}$ is represented as a set
    of $N$ subpopulations; $\{x^{11},...,x^{1m},x^{21},...,x^{2m},...,x^{N1},...,x^{Nm}\}$ is the set of $N\times m$
    individuals.
 3: Step2: generate $N$ uniform weights $\{\lambda_1,...,\lambda_N\}$, and determine the neighbor weights
    of each uniform weight. Take $\lambda_i$ as an example, its neighbor vector is: $B(i)=\{\lambda_{i1},...,\lambda_{iT}\}$.
 4: Step3: calculate the pbest and gbest for each subpopulation (the method uses aggre-
    gation functions).
 5: Step4: three representative solutions(i.e. $x^c,x^s,x^b$) are selected according to the
    inflection point solution selection principle. $EP=\{x^c,x^s,x^b\}$.
 6: Update:
 7: while (The termination condition is not met) do
 8:     for ($i=1,...,N$) do
 9:         step[1]: Each individual in the subpopulation $\{x^{i1},...,x^{im}\}$ is
10:         updated, and the update formula is the particle update formula in
11:         PSO. Calculate gbest and set $y=gbest$.
12:         step[2]: Update the neighbor's gbest,
13:         for (j=1,..., T) do
14:             if $g^{ws}(y|\lambda_i)\geq g^{ws}(gbest^j|\lambda_j)$, let $gbest^j=y$.
15:         end for
16:         step[3]: The $EP$ are updated by using the principle of inflection
17:         point solution selection.
18:     end for
19: end while
```

The input of the algorithm is a multi-objective optimization problem (MOP). The termination condition is iterations. N is the number of subpopulations, while m is the number of individuals in each subpopulation x^i. T is the number of neighbor vectors.

The first step, initialization operation, is aimed at generating an initialized population, encoded by Figure 5. In the second step, the uniform weights are determined for each subpopulation correspondingly. The third step looks for *pbest* and *gbest* in each subpopulation respectively. Due to the character of decomposition, some scalar optimization problems need to be aggregated and used for searching *pbest* and *gbest*. Finally, according to the strategy of finding representative solutions, (x^c,x^p,x^f) are selected from the

Table 1. The Brief Description of benchmark Testing Cases

ID	Lexical feature	Linguistic feature	Structural feature
$101 \sim 104$	$\surd$	$\surd$	$\surd$
$201 \sim 210$	$\times$	$\times$	$\surd$
$221 \sim 247$	$\surd$	$\surd$	$\times$
$248 \sim 266$	$\times$	$\times$	$\times$
$301 \sim 304$	$--$	$--$	$--$

Table 2. Determination of parameters using the control variable method.

ID	Fix value	Changed value
1	m=40,G=180	T=2,3
2	T=2,G=180	m=20,25,30,35,40,45
3	m=40,T=2	G=100,120,140,160,180,200

initialized population and placed in the set EP. The weighted sum approach functions, a combination of the different objectives, used to find *gbest* and *pbest* are shown in the following Equation (22):

$$g^{ws}(x|\lambda) = \sum_{i=1}^{m} \lambda_i f_i(x) \tag{22}$$

where, $\lambda = (\lambda_1, ..., \lambda_m)^T$ is a weight vector and subject to $\sum_{i=1}^{m} \lambda_i = 1$. A MOP can be represented as $F(x) = (f_1(x), ..., f_m(x))^T$, x is subject to Ω (the decision space).

5. Experiment.

5.1. Experimental Design.

Two tracks from the famous Ontology Alignment Evaluation Initiative(OAEI), benchmark and conference, were used to verify the effectiveness of the our approach. According to the heterogeneity, the benchmark tract can be divided into five groups. The details from the benchmark track is shown in Table 1, while conference is a set of weak informative ontologies.

5.2. Experiment Configuration.

The parameter configuration:

- $c_1 = 2; c_2 = 2.$
- $\omega_{max} = 0.9; \omega_{min} = 0.4.$
- $v_{max} = 0.2.$
- $x_{max} = 1.0; x_{min} = 0.0.$
- the number of sub-problem: 7;
- the number of iterations: 180;
- the number of neighboring solutions: $T = 2$

In this paper, parameter sensitivity experiments are performed to configure an optimal set of parameters for K-MOPSO/D. For K-MOPSO/D, the number of neighboring solutions T, the number of particles of the subpopulation m , and the number of iterations of the algorithm G. The values of these four parameters have a significant impact on the performance of the algorithm, so we fix tow of them as a way to verify the impact of different values of the last parameter on the performance of the algorithm. As shown in the Table 2, all the test results are obtained from the 304 test set validation. Table 3, Table 4, and Table 5 record the three cases of Table 4, respectively. Columns 2, 3, and 4 of the table show the recall (R), precision (P), and F-measure (F) corresponding to the

Table 3. Sensitivity test results for the number of neighboring solutions T.

T	R	P	F
2	**0.60(0.016)**	**1(0.000)**	**0.74(0.117)**
3	0.50(0.095)	1(0.000)	0.66(0.087)

Table 4. Sensitivity test results for particle number m of subpopulations

m	R	P	F
20	0.48(0.069)	1(0.000)	0.63(0.061)
25	0.52(0.059)	1(0.000)	0.68(0.052)
30	0.51(0.064)	1(0.000)	0.67(0.056)
35	0.52(0.048)	1(0.000)	0.68(0.043)
40	**0.62(0.016)**	**1(0.000)**	**0.74(0.117)**
45	0.51(0.091)	1(0.000)	0.66(0.085)

Table 5. Sensitivity test results for the number of iterations G

G	R	P	F
100	0.48(0.103)	1(0.000)	0.64(0.099)
120	0.46(0.096)	1(0.000)	0.62(0.090)
140	0.51(0.091)	1(0.000)	0.66(0.085)
160	0.52(0.071)	1(0.000)	0.68(0.060)
180	**0.60(0.016)**	**1(0.000)**	**0.74(0.117)**
200	0.60(0.091)	1(0.000)	0.74(0.088)

knee solution driven by the precision, respectively, with the numbers before and within the parentheses indicating the mean and standard deviation, and the best results in bold.

Number of neighbor solutions T:The number of neighbor solutions T is an important parameter of the decomposition strategy. If T is too small, it will result in the particle positions being very close before and after the update, thus making the algorithm lose the ability to explore other regions; however, if T is too large, it will generate a large computational complexity when updating the neighbor solutions. Table 3 shows the sensitivity test results for the number of neighbor solutions T, and in this paper, T is set to 2.

Size of subpopulations m: The population size m has a great influence on the algorithm; if m is too small, it will lead to a sharp decrease in the search ability of the algorithm, and conversely, if m is too large, it will make it difficult for the optimal individual to guide the evolutionary direction of the whole population, resulting in a difficult convergence of the algorithm. Table 4 shows the sensitivity test results of the subpopulation particle number m, and the experimental data show that the best ontology matching results are achieved when the particle number of subpopulation is set to 40.

Number of iterations G: The number of iterations G also has a great influence on the experimental results; if G is too small, it makes it difficult for the algorithm to converge, and conversely, if G is too large, it leads to a large consumption of time and memory while the experimental results are almost constant. Table 5 shows the sensitivity test results for the number of iterations G. When G is larger than 180, the experimental results are not significantly improved, and therefore, G is set to 180.

Table 6. Comparison between recall-driven and precision-driven single-target PSO and recall-driven and precision-driven K-MOPSO/D.

ID	PSO (R)	PSO (P)	NRA-based K-MOPSO/D(R)	NRA-based K-MOPSO/D(P)
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	(0.81, 0.98, 0.89)	(0.08, 1.00, 0.15)	**(0.99, 1.00, 0.99)**	(0.99, 1.00, 0.99)
202	(0.90, 1.00, 0.95)	(0.90, 1.00, 0.95)	**(0.91, 1.00, 0.95)**	(0.91, 1.00, 0.95)
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, 0.98)	(0.98, 1.00, 0.98)	**(1.00, 1.00, 1.00)**	(1.00, 1.00, 1.00)
205	(0.92, 0.95, 0.94)	(0.08, 1.00, 0.15)	**(0.97, 0.98, 0.97)**	(1.00, 1.00, 1.00)
206	(0.91, 0.97, 0.94)	(0.10, 1.00, 0.18)	**(0.94, 1.00, 0.97)**	(0.94, 1.00, 0.97)
207	(0.92, 0.97, 0.95)	(0.10, 1.00, 0.18)	**(0.95, 1.00, 0.97)**	(0.95, 1.00, 0.97)
208	(0.83, 0.97, 0.90)	(0.70, 1.00, 0.82)	**(0.99, 1.00, 0.99)**	(0.99, 1.00, 0.99)
209	(0.44, 0.75, 0.55)	(0.09, 1.00, 0.16)	**(0.60, 0.94, 0.73)**	(0.60, 0.94, 0.73)
210	(0.52, 0.80, 0.63)	(0.32, 1.00, 0.49)	**(0.71, 0.95, 0.81)**	(0.69, 0.96, 0.80)
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)	**(0.99, 0.99, 0.99)**	(0.99, 0.99, 0.99)
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)	(1.00, 0.95, 0.97)	(1.00, 0.95, 0.97)
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)
238	(0.96, 0.96, 0.96)	(0.08, 1.00, 0.15)	(0.98, 0.98, 0.98)	(0.98, 0.98, 0.98)
239	(1.00, 0.96, 0.97)	(0.06, 1.00, 0.12)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
240	(0.90, 0.87, 0.89)	(0.15, 1.00, 0.26)	**(0.97, 0.97, 0.97)**	(0.97, 0.97, 0.97)
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, 1.00, 1.00)	(1.00, 1.00, 1.00)
247	(0.90, 0.87, 0.89)	(0.15, 1.00, 0.26)	(0.94, 0.94, 0.94)	(0.94, 0.94, 0.94)
248	(0.77, 0.97, 0.86)	(0.79, 1.00, 0.88)	**(0.86, 1.00, 0.92)**	(0.86, 1.00, 0.92)
249	(0.81, 1.00, 0.89)	(0.08, 1.00, 0.15)	**(0.90, 1.00, 0.95)**	(0.90, 1.00, 0.95)
250	(0.84, 1.00, 0.91)	(0.78, 1.00, 0.88)	**(0.91, 1.00, 0.95)**	(0.91, 1.00, 0.95)
251	(0.79, 1.00, 0.88)	(0.79, 1.00, 0.88)	**(0.87, 1.00, 0.93)**	(0.87, 1.00, 0.93)
252	(0.79, 1.00, 0.88)	(0.07, 1.00, 0.13)	**(0.88, 0.98, 0.92)**	(0.88, 0.98, 0.92)
253	(0.79, 1.00, 0.88)	(0.03, 1.00, 0.06)	**(0.86, 1.00, 0.92)**	(0.86, 1.00, 0.92)
254	(0.78, 1.00, 0.88)	(0.78, 1.00, 0.88)	**(0.79, 1.00, 0.88)**	(10.79, 1.00, 0.88)
257	(0.84, 1.00, 0.91)	(0.78, 1.00, 0.88)	**(0.94, 1.00, 0.97)**	(0.94, 1.00, 0.97)
258	(0.77, 0.97, 0.86)	(0.77, 1.00, 0.10)	**(0.88, 0.99, 0.93)**	(0.87, 1.00, 0.93)
259	(0.79, 0.98, 0.88)	(0.79, 1.00, 0.88)	**(0.88, 0.98, 0.92)**	(0.88, 0.98, 0.92)
260	(0.79, 0.95, 0.86)	(0.06, 1.00, 0.12)	**(0.86, 1.00, 0.93)**	(0.86, 1.00, 0.93)
261	(0.78, 0.89, 0.83)	(0.12, 1.00, 0.21)	**(0.79, 0.93, 0.85)**	(0.79, 0.93, 0.85)
262	(0.78, 1.00, 0.88)	(0.78, 1.00, 0.88)	**(0.79, 1.00, 0.88)**	(0.79, 1.00, 0.88)
301	(0.75, 0.87, 0.81)	(0.18, 1.00, 0.31)	**(0.81, 0.91, 0.86)**	(0.81, 0.94, 0.87)
302	(0.65, 0.82, 0.71)	(0.20, 1.00, 0.34)	(0.65, 0.84, 0.72)	(0.63, 0.86, 0.72)
304	(0.96, 0.95, 0.96)	(0.03, 1.00, 0.07)	**(0.97, 0.97, 0.97)**	(0.47, 1.00, 0.64)

Table 7. Mean and standard deviation of recall and precision for different matching systems

ID	MOP SO (R)	RA-based K-MOPSO/D (R)	NRA-based K-MOPSO/D (R)	MOP SO (P)	RA-based K-MOPSO/D (P)	NRA-based K-MOPSO/D (P)
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	0.90(0.02)	0.99(0.00)	0.99(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
202	0.88(0.02)	0.90(0.01)	0.91(0.01)	0.82(0.02)	1.00(0.00)	1.00(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	0.89(0.02)	1.00(0.00)	1.00(0.00)	0.96(0.03)	1.00(0.00)	1.00(0.00)
205	0.82(0.03)	0.97(0.01)	0.97(0.01)	0.89(0.03)	0.99(0.00)	0.99(0.00)
206	0.88(0.02)	0.94(0.00)	0.94(0.00)	0.92(0.02)	1.00(0.00)	1.00(0.00)
207	0.91(0.01)	0.95(0.00)	0.95(0.00)	0.98(0.01)	1.00(0.00)	1.00(0.00)
208	0.73(0.02)	0.99(0.00)	0.99(0.00)	0.99(0.00)	1.00(0.00)	1.00(0.00)
209	0.26(0.02)	0.60(0.01)	0.60(0.01)	0.89(0.02)	0.93(0.01)	0.93(0.01)
210	0.31(0.01)	0.71(0.00)	0.71(0.00)	0.94(0.01)	0.97(0.01)	0.97(0.01)
221	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)
223	0.89(0.03)	0.99(0.00)	0.99(0.00)	0.94(0.03)	0.99(0.00)	0.99(0.00)
224	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
225	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
228	0.97(0.02)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
230	1.00(0.00)	1.00(0.00)	1.00(0.00)	0.94(0.01)	0.95(0.00)	0.95(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)
232	0.98(0.02)	1.00(0.00)	1.00(0.00)	0.98(0.01)	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)
238	0.88(0.03)	0.99(0.00)	0.99(0.01)	0.94(0.02)	0.99(0.00)	0.99(0.01)
239	1.00(0.00)	1.00(0.00)	1.00(0.00)	0.97(0.02)	1.00(0.00)	1.00(0.00)
240	0.88(0.01)	0.97(0.01)	0.96(0.01)	0.91(0.02)	0.97(0.01)	0.96(0.01)
241	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
246	0.93(0.02)	1.00(0.00)	1.00(0.00)	0.96(0.01)	1.00(0.00)	1.00(0.00)
247	0.91(0.02)	0.96(0.01)	0.94(0.01)	0.88(0.01)	0.94(0.01)	0.94(0.01)
248	0.79(0.01)	0.86(0.01)	0.86(0.01)	1.00(0.00)	1.00(0.00)	1.00(0.00)
249	0.79(0.02)	0.90(0.00)	0.90(0.01)	1.00(0.00)	1.00(0.00)	1.00(0.00)
250	0.90(0.02)	0.93(0.01)	0.93(0.01)	0.98(0.01)	1.00(0.00)	1.00(0.00)
251	0.80(0.02)	0.87(0.00)	0.87(0.00)	0.99(0.01)	1.00(0.00)	1.00(0.00)
252	0.79(0.01)	0.88(0.01)	0.86(0.01)	0.97(0.02)	0.97(0.01)	1.00(0.00)
253	0.77(0.03)	0.86(0.01)	0.86(0.02)	0.97(0.01)	1.00(0.00)	1.00(0.00)
254	0.79(0.01)	0.79(0.00)	0.79(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
257	0.89(0.03)	0.93(0.02)	0.93(0.02	0.96(0.02)	1.00(0.00)	1.00(0.00)
258	0.77(0.03)	0.87(0.00)	0.87(0.01)	0.96(0.02)	1.00(0.00)	1.00(0.00)
259	0.80(0.03)	0.88(0.01)	0.87(0.01)	0.92(0.03)	0.97(0.01)	0.97(0.01)
260	0.78(0.02)	0.85(0.01)	0.85(0.01)	0.96(0.02)	1.00(0.00)	1.00(0.00)
261	0.79(0.02)	0.79(0.00)	0.79(0.00)	0.87(0.03)	0.93(0.00)	0.93(0.00)
262	0.79(0.01)	0.79(0.01)	0.79(0.01)	1.00(0.00)	1.00(0.00)	1.00(0.00)
301	0.75(0.03)	0.81(0.00)	0.81(0.00)	0.88(0.02)	0.92(0.01)	0.92(0.01)
302	0.62(0.02)	0.64(0.01)	0.64(0.01)	0.85(0.01)	0.86(0.00)	0.86(0.00)
303	0.76(0.03)	0.79(0.01)	0.79(0.01)	0.62(0.02)	0.69(0.01)	0.69(0.01)
304	0.91(0.03)	0.97(0.00)	0.97(0.00)	0.96(0.01)	0.99(0.01)	1.00(0.01)

Table 8. T-test statistical analysis for recall and precision

ID	NRA-based MOPSO/D(R) VS MOPSO(R)	NRA-based K-MOPSO/D(R) VS RA-based K-MOPSO/D(R)	NRA-based K-MOPSO/D(P) VS MOPSO(P)	NRA-based K-MOPSO/D(P) VS RA-based K-MOPSO/D(P)
101	0.00	0.00	0.00	0.00
103	0.00	0.00	0.00	0.00
104	0.00	0.00	0.00	0.00
201	24.65	0.00	0.00	0.00
202	7.35	0.00	49.30	0.00
203	0.00	0.00	0.00	0.00
204	30.12	0.00	7.30	0.00
205	25.98	0.00	18.26	0.00
206	16.43	0.00	21.91	0.00
207	21.91	0.00	10.95	0.00
208	71.20	0.00	0.00	0.00
209	83.28	0.00	9.80	0.00
210	219.09	0.00	11.62	0.00
221	0.00	0.00	0.00	0.00
222	0.00	0.00	0.00	0.00
223	18.26	0.00	9.13	0.00
224	0.00	0.00	0.00	0.00
225	0.00	0.00	0.00	0.00
228	8.22	0.00	0.00	0.00
230	0.00	0.00	5.48	0.00
231	0.00	0.00	0.00	0.00
232	5.48	0.00	10.95	0.00
233	0.00	0.00	0.00	0.00
238	19.05	0.00	13.69	0.00
239	0.00	0.00	8.22	0.00
240	30.98	-3.87	12.25	-3.87
241	0.00	0.00	0.00	0.00
246	19.17	0.00	21.91	0.00
247	7.35	-7.75	23.24	0.00
248	27.11	0.00	0.00	0.00
249	42.60	0.00	0.00	0.00
250	7.35	0.00	10.95	0.00
251	19.17	0.00	5.48	0.00
252	27.11	-7.75	0.00	0.00
253	13.67	0.00	16.43	0.00
254	0.00	0.00	0.00	0.00
257	6.08	0.00	10.95	0.00
258	17.32	0.00	10.95	0.00
259	12.12	-7.75	8.66	0.00
260	17.15	0.00	10.95	0.00
261	0.00	0.00	10.95	0.00
262	0.00	0.00	0.00	0.00
301	10.95	0.00	9.80	0.00
302	4.90	0.00	5.48	0.00
303	5.20	0.00	17.15	0.00
304	10.95	0.00	11.62	0.00

Table 9. Results of benchmark: f-measure

ID	edna	AML	Cro Mat ch	Lily	Lo gM ap	Log Map Lt	XMap	Log Map Bio	NRA-based K-MOPSO /D
101	0.78	0.00	1.00	1.00	0.95	0.71	0.97	0.52	1.00
201	0.62	0.44	1.00	1.00	0.84	0.62	0.77	0.47	0.99
202	0.62	0.42	0.98	0.99	0.85	0.62	0.76	0.48	0.95
221	0.78	0.51	1.00	1.00	0.94	0.72	0.97	0.53	1.00
222	0.77	0.50	1.00	1.00	0.00	0.72	0.78	0.00	1.00
224	1.00	0.51	1.00	1.00	0.94	0.90	0.97	0.53	1.00
225	0.78	0.51	1.00	1.00	0.95	0.72	0.97	0.52	1.00
228	0.55	1.00	1.00	1.00	0.92	0.48	1.00	0.80	1.00
232	1.00	0.51	1.00	1.00	0.94	0.90	0.97	0.53	1.00
233	0.55	1.00	1.00	1.00	0.92	0.48	1.00	0.80	1.00
238	1.00	0.51	1.00	1.00	0.95	0.90	0.97	0.52	1.00
239	0.55	1.00	1.00	1.00	0.92	0.48	1.00	0.80	1.00
241	1.00	1.00	1.00	1.00	0.92	0.80	1.00	0.80	1.00
246	1.00	1.00	1.00	1.00	0.92	0.80	1.00	0.80	1.00
248	0.62	0.45	0.98	0.97	0.80	0.62	0.77	0.43	0.92
249	0.79	0.42	0.95	0.95	0.82	0.80	0.82	0.48	0.95
250	0.43	0.90	0.94	0.98	0.81	0.40	0.88	0.69	0.97
251	0.61	0.43	0.93	0.97	0.00	0.61	0.59	0.00	0.93
252	0.62	0.45	0.97	0.95	0.83	0.62	0.76	0.49	0.92
253	0.79	0.42	0.97	0.94	0.86	0.79	0.81	0.47	0.92
254	0.43	0.92	0.94	0.98	0.81	0.40	0.88	0.00	0.88
257	0.79	0.92	0.90	0.95	0.88	0.72	0.88	0.74	0.97
258	0.79	0.45	0.93	0.95	0.00	0.80	0.66	0.00	0.93
259	0.79	0.46	0.96	0.94	0.83	0.80	0.81	0.00	0.92
260	0.43	0.90	0.93	0.95	0.00	0.41	0.90	0.00	0.93
261	0.43	0.90	0.89	0.89	0.83	0.40	0.88	0.00	0.85
262	0.79	0.92	0.94	0.94	0.81	0.70	0.88	0.69	0.88
avg	0.72	0.65	0.97	0.98	0.75	0.66	0.88	0.45	0.96

5.3. **Results and Analysis.** Two tracks from the famous Ontology Alignment Evaluation Initiative(OAEI), benchmark and conference, were used to verify the effectiveness of the our approach. The contrast experiment is from three perspectives: (1)demonstrating the effectiveness of multi-objective optimization. (2)verifying the effectiveness of the proposed NRA optimization model; (3) Comparing the pros and cons between the proposed method and the state-of-the-art ontology matching system.

To verify the validity of our proposed multi-objective model, we compare K-MOPSO/D with single-objective PSO. As shown in Table 6, The first and third columns represent the optimal points of the single-target PSO driven by the recall and precision, respectively, with the corresponding recall, precision, and f-measure for that point inside the parentheses, and similarly, the second and fourth columns represent the optimal points of the K-MOPSO/D driven by the recall and precision, with the corresponding recall, precision, and f-measure for that point inside the parentheses. Regarding the recall, our proposed K-MOPSO/D performs better than single-target PSO on 26 test sets (the better data are bolded). Further, on some datasets, although there is no result improvement in K-MOPSO/D in terms of recall, it performs better in terms of precision. For example, on

Table 10. Results of conference:f-measure

ID	AML	Log Map	LogM apLt	XMap	NRA-based K-MOPSO/D
cmt-conference	0.59	0.62	0.42	0.00	0.72
cmt-confOf	0.69	0.45	0.48	0.58	0.62
cmt-edas	0.83	0.73	0.67	0.72	0.82
cmt-ekaw	0.63	0.63	0.50	0.67	0.67
cmt-iasted	0.89	0.89	0.89	0.89	0.89
cmt-sigkdd	0.92	0.91	0.76	0.87	0.88
conference-confOf	0.87	0.79	0.72	0.75	0.81
conference-edas	0.69	0.73	0.62	0.71	0.74
conference-ekaw	0.75	0.53	0.42	0.59	0.72
conference-iasted	0.50	0.64	0.42	0.45	0.64
conference-sigkdd	0.79	0.79	0.64	0.69	0.80
confOf-edas	0.71	0.62	0.58	0.67	0.74
confOf-ekaw	0.86	0.80	0.61	0.78	0.85
confOf-iasted	0.57	0.62	0.62	0.52	0.63
confOf-sigkdd	0.92	0.83	0.73	0.67	0.83
edas-ekaw	0.59	0.62	0.50	0.62	0.73
edas-iasted	0.60	0.52	0.52	0.48	0.81
edas-sigkdd	0.80	0.61	0.61	0.64	0.83
ekaw-iasted	0.78	0.74	0.60	0.64	0.84
ekaw-sigkdd	0.76	0.74	0.74	0.70	0.77
iasted-sigkdd	0.84	0.85	0.73	0.76	0.79
avg	0.74	0.70	0.61	0.64	0.77

the 230, both K-MOPSO/D and PSO have a recall of 1, but K-MOPSO/D has a precision of 0.95, which is higher than PSO's precision of 0.93. Concerning precision, in single target PSO, although a high precision can be obtained, the recall result deteriorates very badly, so that the result of blindly pursuing precision and ignoring recall is not available to users. For example, in 206, although the recall of PSO reaches 1, the precision is only 0.1, which indicates that although the correct rate of matching pairs found is high, the number of matching pairs found is too small. In contrast, our proposed K-MOPSO/D can optimize both recall and precision at the same time, which is more suitable for users to use. In summary, our proposed multi-objective optimization model is very effective.

To verify the advancement and effectiveness of our proposed NRA-based K-MOPSO/D, we compared the experimental results with those of MOPSO and RA-based K-MOPSO/D. As shown in Table 7, the first three columns indicate the means and standard deviations of the recalls for MOPSO, RA-based K-MOPSO/D, and NRA-based K-MOPSO/D on different data sets, respectively. Similarly, the last three columns denote the mean and standard deviation of MOPSO, RA-based K-MOPSO/D and NRA-based K-MOPSO/D on different data sets for precision, respectively. Means and standard deviations are derived from 30 independent operations. Since MOPSO, RA-based K-MOPSO/D, and NRA-based K-MOPSO/D are all non-deterministic algorithms, the t-test was used to verify whether there is a significant difference between the different methods with a significance level of 0.05. Specifically, the original hypothesis is that there is no significant difference in performance between NRA-based K-MOPSO/D and the other matchers, and the alternative hypothesis is that there is a difference in performance between NRA-based K-MOPSO/D and the other matchers. In this paper, since the total number of samples is 30 and the significance level is 0.05, it indicates a significant difference in the performance of the two matchers when $|t| \geq 2.045$ is satisfied, and further, the performance of NRA-based K-MOPSO/D is better when $t \geq 2.045$, otherwise, the performance of NRA-based K-MOPSO/D has a worse performance. As shown in Table 8, the first and second columns indicate the t-values of NRA-based K-MOPSO/D versus MOPSO and RA-based K-MOPSO/D on recall, respectively. Similarly, the third and fourth columns denote the t-values of NRA-based K-MOPSO/D versus MOPSO and RA-based K-MOPSO/D on precision, respectively. As can be seen from Table 8, regarding recall, NRA-based K-MOPSO/D has significantly higher statistical results than MOPSO on 28 datasets, while NRA-based K-MOPSO/D performs worse than RA-based K-MOPSO/D on only 4 datasets. regarding precision, NRA-based K-MOPSO/D has significantly higher statistical results than MOPSO on 25 datasets, while NRA-based K-MOPSO/D performs better than RA-based K-MOPSO/D on only 1 datasets. In summary, NRA- based K-MOPSO/D is clearly due to MOPSO and NRA- based K-MOPSO/D is not lower than RA-based K-MOPSO/D on most of the data sets, which proves the effectiveness of the proposed NRA optimization model.

Table 9 shows the performance among OAEI participants and our proposed NRA-based K-MOPSO/D on the f-measure. From the average of the last line, it can conclude that CroMatch and Lily are the most successful, followed by our method. Compared to CroMatch, our approach is 0.01 behind, and 0.02 behind Lily. The fourth-ranked matching system, XMap, is 0.12 behind our system, while other matching systems are even further behind. The two matching systems, CroMatch and Lily are untouchable in the benchmark test, despite some OAEI participants emerging in recent years. In the case of severe heterogeneity, such as $248 \sim 262$, the two systems tend to obtain relatively stable results. Because the f-measure is the harmonic average of recall and precision and our approach boasts lower recall, the averages of CroMatch and Lily are better than ours in the f-measure.

Table 10 shows the results of the OAEI participants and NRA-based K-MOPSO/D on the conference track. AML, LogMap, LogMapLt, and XMap are excellent systems in this track. As a weak informative ontology track, the experimental results of CroMatch and Lily are not outstanding enough like the benchmark track. The mean value of the f-measure shows that the NRA-based K-MOPSO/D performed the best compared to the other four participants. Compared to CroMatch and Lily, the NRA-based K-MOPSO/D parses only significant parts of the ontology and makes reasonable use of some similarity metrics. As a result, the performance is slightly inferior to that of CroMatch and Lily on the benchmark, but very good on the conference. From the experimental results.

6. **Conclusion and Future Work.**

To solve the heterogeneity of ontology, and consider the different preference habits of users, the ontology matching problem is modeled as a multi-objective optimization model. In this optimization model, an inflection point selection strategy is designed to satisfy different preferences. To solve this model, the K-MOPSO/D algorithm is designed, and its performance is verified by two tracks in OAEI. Through the experimental results, our method has achieved good results only by using a small amount of ontology information. However, there are also some shortcomings in our approach, such as the lower recall in some test cases in the benchmark track. The main reason for this imperfection is that much of the confidence between entities is lower, and the chosen matchers do not overcome this problem. At present, one of the obstacles in this domain is that the similarity measure can not be improved effectively. In addition, the method of this paper can not be reused for large-scale matching directly, because the operation of the similarity matrix is very time-consuming in our approach. With the rise of the Internet of Things, the size of ontology become increasingly larger correspondingly. Finding the right way to deal with this large-scale ontology matching problem presents a serious challenge. Based on the above problems, we will devote ourselves to overcoming the following challenges in the future: (1) Based on the existing matcher, new matchers will be trained through machine learning to improve the performance; (2) New technologies will be designed in large-scale ontology matching to overcome time-consuming and labor-consuming.

7. **Acknowledgement.** This work is supported by the Natural Science Foundation of Fujian Province(NO. 2022J01644)

REFERENCES

[1] Y.-R. Ma, Y.-J. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent and Fuzzy Systems*, vol. 43, pp. 1–13 , 2022.

[2] F.-Q. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G.-Y. Ding, P. Mei, and L.-Y. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access* , vol. 8, pp. 104555–104564, 2020.

[3] X.-S. Xue, and J.-F. Chen, "Matching biomedical ontologies through compact differential evolution algorithm with compact adaption schemes on control parameters," *Neurocomputing*, vol. 458, pp. 526–534, 2021.

[4] X.-S. 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, pp. 107343, 2021.

[5] X.-S. Xue, and J.-H. Liu, "Collaborative ontology matching based on compact interactive evolutionary algorithm" *Knowledge-Based Systems*, vol. 197, pp. 94–103, 2017.

[6] X.-S. Xue, and C. Jiang, "Matching sensor ontologies with multi-context similarity measure and parallel compact differential evolution algorithm," *IEEE Sensors Journal*, vol. 21, pp. 24570–24578, 2021.

[7] X.-S. Xue, and W.-Y. Liu, "Integrating heterogeneous ontologies in Asian languages through compact genetic algorithm with annealing re-sample inheritance mechanism," in *Transactions on Asian and Low-Resource Language Information Processing*, ACM New York, NY, 2022.

[8] X.-S. Xue, X.-J. Wu, and C. Jiang, G.-J. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, 2021.

[9] A. Bento, A. Zouaq, and M. Gagnon, "Ontology matching using convolutional neural networks," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 5648–5653.

[10] M. Guia, R.-R. Silva, and J. Bernardino, "A hybrid ontology-based recommendation system in e-commerce," *Algorithms*, vol. 12, pp. 239, 2019.

[11] X.-S. Xue, and Z.Y. Tang, "An evolutionary algorithm based ontology matching system," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, pp. 551–556, 2017.

[12] X.-S. Xue, Y.-Q, Wang, and W.-C. Hao, "Using MOEA/D for optimizing ontology alignments," *Soft Computing*, vol. 18, pp. 1589–1601, 2014.

[13] X.-S. Xue, and J.-w. Lu, "A compact brain storm algorithm for matching ontologies," *IEEE Access*, vol. 8, pp. 43898–43907, 2020.

[14] D.-S. Wang, D.-P. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, pp. 387–408, 2018.

[15] W.-f. Gao, W. Xu, M.-G. Gong, and G.-G. Yen, "A decomposition-based evolutionary algorithm using an estimation strategy for multimodal multi-objective optimization," *Information Sciences*, vol. 606, pp. 531–548, 2022.

[16] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, pp. 440–462, 2016.

[17] X.-S. Xue, and J.-F. Chen, "Using compact evolutionary tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[18] Y. Zhu, and D. Shasha, "VLDB'02: Proceedings of the 28th International Conference on Very Large Databases," Elsevier Amsterdam, The Netherlands, 2002.

[19] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with COMA++," in *AProceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005, pp. 906–908.

[20] C. Drumm, M. Schmitt, H.-H. Do, and E. Rahm, "Quickmig: automatic schema matching for data migration projects,"in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, 2007, pp. 107–116.

[21] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi, "A framework for modeling and evaluating automatic semantic reconciliation," *The VLDB Journal*, vol. 14, pp. 50–67, 2005.

[22] J. Bock,and J. Hettenhausen, "Discrete particle swarm optimisation for ontology alignment," *Information Sciences*, vol. 192, pp. 152–173, 2012.

[23] J. Martinez-Gil, E. Alba, and J. Aldana-Montes,, "Optimizing ontology alignments by using genetic algorithms," in *Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web. Karlsruhe, Germany*, 2008.

[24] J.-L. Wang, Z.-J. Ding, and C.-J. Jiang, "GAOM: genetic algorithm based ontology matching," in *2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*, 2006, pp. 617–620.

[25] P. Lambrix, and H. Tan, "SAMBO—a system for aligning and merging biomedical ontologies," *Journal of Web Semantics*, vol. 4, pp. 196–206, 2006.

[26] A. Doan, P. Domingos, and A.-Y. Halevy, "Reconciling schemas of disparate data sources: A machine-learning approach," in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, 2001, pp. 509–520.

[27] X.-S. Xue, and X. Yao, "Interactive ontology matching based on partial reference alignment," *Applied Soft Computing*, vol. 72, pp. 355–370, 2018.

[28] Q. Lv, C.-C. Jiang, and H. Li, "Solving ontology meta-matching problem through an evolutionary algorithm with approximate evaluation indicators and adaptive selection pressure," *IEEE Access*, vol. 9, pp. 3046–3064, 2020.

[29] X.-S. Xue, and J.-F. Chen, "Optimizing sensor ontology alignment through compact co-firefly algorithm," *Sensors*, vol. 20, pp. 2056, 2020.

[30] Z.-M. Lv, and R. Peng, "A novel meta-matching approach for ontology alignment using grasshopper optimization," *Knowledge-Based Systems*, vol. 201, pp. 106050, 2020.

[31] C. Jiang, and X.-S. Xue,, "A uniform compact genetic algorithm for matching bibliographic ontologies," *Applied Intelligence*, vol. 51, pp. 7517–7532, 2021.

[32] G. Acampora, U. Kaymak, V. Loia, and A. Vitiello, "Applying NSGA-II for solving the ontology alignment problem," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 1098–1103.

[33] X.-S. Xue, Y.-P. Wang, W.-C. Hao, and J. Hou, "Optimizing ontology alignments through NSGA-II without using reference alignment," *Computing and Informatics*, vol. 33, pp. 857–876, 2014.

[34] X.-S. Xue, and Z.-Y. Tang, "An evolutionary algorithm based ontology matching system," in *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, 2017, pp. 551–556.

[35] X.-S. Xue, and Y.-P. Wang, "Ontology alignment based on instance using NSGA-II," *Journal of Information Science*, vol. 41, pp. 58–70, 2015.

[36] X.-S. Xue, and J.-H. Liu, "Optimizing ontology alignment through compact MOEA/D," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, pp. 1759004, 2017.

[37] C.-M. Chen, S. Lv, J. Ning, and J.-M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, pp. 124, 2023.

[38] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial IoT and smart city services," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, IEEE, 2020, pp. 1–6.

[39] A. Wong, F. Yip, P. Ray, and N. Paramesh, "Towards semantic interoperability for IT governance: an ontological approach," *Computing and Informatics*, vol. 27, pp. 131–155, 2008.

[40] T.-R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, pp. 199–220, 1993.

[41] 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, pp. 89–216, 2012.

[42] X.-S. Xue, and Y.-P. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 580–591, 2015.

[43] X.-S. Xue, and Z. Hang,and A.-Y. Tang,, "Interactive biomedical ontology matching," *PloS One*, vol. 14, pp. e0215147, 2019.

[44] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," in *The Semantic Web–ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings 4*, 2005, pp. 624–637.

[45] Z.-B. Wu, and M. Palmer, "Verb semantics and lexical selection," *ArXiv Preprint Cmp-lg/9406033*, 1994.

[46] M. Biniz, and R. El Ayachi, "Optimizing ontology alignments by using Neural NSGA-II," *Journal of Electronic Commerce in Organizations (JECO)*, vol. 16, pp. 29–42, 2018.

[47] X.-S. Xue, "Complex ontology alignment for autonomous systems via the Compact Co-Evolutionary Brain Storm Optimization algorithm," *ISA Transactions*, vol. 132, pp. 190–198, 2023.

[48] R.-C. Eberhart, and Y.-H. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, vol. 1, 2020, pp. 84–88.

[49] Q.-F. Zhang, and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, 2007.