



Article ID 1007-1202(2022)03-0240-15

DOI <https://doi.org/10.1051/wujns/2022273240>

Improving the Efficiency of Multi-Objective Grasshopper Optimization Algorithm to Enhance Ontology Alignment

□ LV Zhaoming, PENG Rong[†]

School of Computer Science, Wuhan University, Wuhan 430072, Hubei, China

© Wuhan University 2022

Abstract: Ontology alignment is an essential and complex task to integrate heterogeneous ontology. The meta-heuristic algorithm has proven to be an effective method for ontology alignment. However, it only applies the inherent advantages of meta-heuristics algorithm and rarely considers the execution efficiency, especially the multi-objective ontology alignment model. The performance of such multi-objective optimization models mostly depends on the well-distributed and the fast-converged set of solutions in real-world applications. In this paper, two multi-objective grasshopper optimization algorithms (MOGOA) are proposed to enhance ontology alignment. One is ε -dominance concept based GOA (EMO-GOA) and the other is fast Non-dominated Sorting based GOA (NS-MOGO). The performance of the two methods to align the ontology is evaluated by using the benchmark dataset. The results demonstrate that the proposed EMO-GOA and NS-MOGO improve the quality of ontology alignment and reduce the running time compared with other well-known metaheuristic and the state-of-the-art ontology alignment methods.

Key words: ontology alignment; multi-objective grasshopper optimization algorithm; ε -dominance; fast non-dominated sorting; knowledge integration

CLC number: TP 182

Received date: 2022-03-18

Foundation item: Supported by the Ministry of Education-China Mobile Joint Fund Project (MCM2020J01)

Biography: LV Zhaoming, male, Ph. D. candidate, research direction: knowledge engineering, ontology matching and swarm intelligence. E-mail: zhaominglv@whu.edu.cn

[†] To whom correspondence should be addressed. E-mail: rongpeng@whu.edu.cn

0 Introduction

Ontology is a clear formal specification for sharing concepts and is widely used to support knowledge integration and interoperability such as intelligent transportation systems^[1], industrial megaprojects analytics^[2], product lifecycle management^[3], and cognitive and robotic systems^[4]. In recent years, more and more ontologies have been developed by using the Semantic Web standard technologies in the same domain of interest. However, different ontology design experts may use different terms to describe the concept of the same thing, which emerges the heterogeneity problem between ontologies. Therefore, ontology-based data management is becoming more and more complex. In order to apply these knowledge widely, ontology alignment is a key process. The goal of ontology alignment is to find a set of semantic correspondences between different ontologies for further handled knowledge-based systems. In the past few years, many ontology alignment methods or techniques have been introduced in the literature. Especially the method of weighted aggregation for combining multiple basic matchers through swarm and evolutionary algorithms has demonstrated to be an effective method. These methods utilize weights to combine lexical-based, linguistic-based, and structure-based matchers for finding more correspondences between entities to make full use of the information contained within context of an ontology. Weights based methods

are divided into two categories: fixed weights and dynamic weights. However, the fixed weights manually configured according to experimental results have certain limitations, which may reduce the quality of alignment. Therefore, adaptive weights allocation are an effective method for combining matchers adequately.

In recent years, the successful application of meta-heuristic algorithm is very noteworthy. Swarm intelligence algorithm has successfully solved the ontology alignment problem and obtained the better ontology alignment quality in the form of a single objective. However, these methods only apply the inherent advantages of meta-heuristics and rarely consider the execution efficiency, especially the multi-objective ontology alignment model. Further, the performance of such multi-objective optimization models mostly depends on the well-distributed and the fast-converged set of solutions in specific applications. In order to improve the efficiency and effect of multi-objective ontology alignment methods, in this work, the grasshopper optimization algorithms are improved to perform multi-objective ontology alignment tasks more efficiently. Grasshopper Optimization Algorithm (GOA) is a population based meta-heuristic algorithm inspired by the behavioral characteristics of grasshopper swarms to simulate exploration and exploitation processes of the swarm intelligence algorithm^[5]. According to the claim of Bock *et al*^[6], ontology alignment as an optimization problem needs to maintain the number of correspondences and a set of correct correspondences at the same time. Therefore, the ontology alignment can be regarded as a multi-objective optimization problem with two objectives: the number of correspondences and the set of correct correspondences. Further, GOA balances the exploration and exploitation in the search process through an adaptive parameter. The high repulsive force between grasshoppers gives GOA the advantages of high exploration and local optimal avoidance. The attraction between grasshoppers makes GOA algorithm have strong exploitation and convergence performance. These characteristics make the GOA algorithm more adaptable to complex multi-objective search problems and surpass other algorithms such as particle swarm optimization (PSO), bat optimization algorithm (BA), Bonobo Optimizer (BO), and firefly algorithm (FA). These advantages motivate us to develop an ontology alignment system based on multi-objective GOA. In this paper, two different multi-objective GOA are proposed to perform ontology align-

ment tasks in a more efficient way. One is ε -dominance concept based GOA (EMO-GOA) and the other is fast non-dominated sorting based GOA (NS-MOGOA). The proposed EMO-GOA approach uses the grid technology based on ε -dominance concept to keep the diversity of the non-dominated solution in the archive, and reduce the time to converge to the Pareto optimal solution. The NS-MOGOA uses a fast non-dominated sorting method by assigning ranking and crowding distance to the population, which speeds up the completion of the ontology alignment task.

The remainder of this paper is organized as follows. Section 1 describes the related work of this paper. Section 2 describes the related concepts underlying our research work. Section 3 gives the mathematical model of the multi-objective ontology alignment problem. The modified EMO-GOA is proposed in Section 4. Section 5 proposes the NS-MOGOA. Section 6 initializes the conditions required for the experiments and the results are discussed. Section 7 concludes with an overview of our future work.

1 Related Works

In the past few years, evolutionary computing technology based ontology alignment methods have been proposed to address the problem of combining multiple basic matchers. These methods are divided into two categories: direct optimization alignment and the ontology meta-matching technique. The method of directly optimizing an alignment is to search for a correspondence set in the candidate correspondences. This set is also called an alignment. Furthermore, each confidence value for a correspondence is calculated by combining multiple basic matchers. The most famous method of this type is MapPSO, which directly optimizes an alignment evaluated by a fitness function with two objectives^[6]. The method minimizes the aggregate values of lexical-based, linguistic-based, and structure-based matchers and evaluates candidate alignments by using fitness functions. The optimization process is performed by using discrete particle swarm optimization. In detail, a candidate solution is randomly selected from the Class, Data property and Object property space. The update process deletes some correspondences when the fitness function decreases, and some new correspondences are added. When the termination condition is reached, a sub-optimal alignment is found. The limitation of this

method is that the weight value of the combined matchers is fixed. This similar work is GAOM, which uses an evaluation function to evaluate candidate alignments in the optimization process of genetic algorithms^[7]. In recent years, Mohammadi *et al*^[8] proposed a method to directly optimize ontology alignment based on simulated annealing algorithm from the perspective of reducing the time of calculation and memory. The authors model the ontology alignment problem as a function of optimizing a state, and an optimal state is obtained by performing a simulated annealing algorithm. This method uses several matchers to calculate a similarity matrix. The confidence of each correspondence is taken from the maximum value calculated by these matchers which utilize all information of a pair of entities such as name, label and comment.

Another category is the ontology meta-matching technology, which can be considered to automatically select or configure appropriate weights and filtering thresholds during optimization process^[9]. The earliest meta-matching method is GOAL^[10]. Instead of directly optimizing the candidate alignment between two ontologies, this method uses a genetic algorithm to set the weights, and to find a suboptimal alignment. This algorithm searches for different weight configurations, i.e., a candidate solution vector, and finds the best set of weights in infinite search space. The evaluation metric of a solution is to use the precision and recall of the reconciliation. Similarly, Naya *et al*^[11] used the genetic algorithm to find out appropriate weights for aggregating different similarity metrics into a final measure. Each individual represents a specific combination of metrics, and the method finds the combination and provides the best alignment. The genetic algorithm was also used in Ref. [12] for optimizing a suboptimal alignment. The difference is that the weight and the threshold are coded in the chromosome at the same time. Xue *et al*^[13] utilized multi-objective genetic algorithm (NSGA-II) to align ontology. In addition, also in Ref. [14], the multi-objective evolutionary algorithm based on decomposition (MOEA/D) was utilized for solving ontology alignment problem. For these two multi-objective approaches, precision and recall are used as two objectives. Although these meta-matching methods have obtained good alignment quality, they rarely consider the execution efficiency of meta-heuristics algorithms, especially the multi-objective on-

tology alignment model.

In recent years, researchers have proposed different methods based on newly developed meta-heuristic algorithms to solve the ontology meta-matching problem, and these methods do not require a priori knowledge in the alignment process. For example, Acampora *et al*^[15] proposed an ontology alignment process based on a hybrid evolutionary algorithm, which can effectively aggregate a variety of basic matchers without a priori knowledge. In the end, a suboptimal alignment was obtained between the given two ontologies. In addition, also in Ref. [16], the NSGA-II was utilized to solve the ontology meta-matching problem and the problem of prior knowledge. Ryma *et al*^[17] proposed a hybridization method by combining the Hill climbing local search technique into the genetic algorithm for large ontology alignment without a priori knowledge. Forsati *et al*^[18] proposed an effective method based on Harmony Search (HSOMap). This method uses a weighted harmonic-mean method to aggregate various matchers into a single confidence among all possible correspondences between two ontologies. Lv *et al*^[19] proposed a novel approach by using grasshopper optimization algorithm to solve the ontology meta-matching problem. This method makes use of various matchers and does not require a priori knowledge in the alignment process. As each basic matcher can capture the information of an entity from a different perspective, the aggregation technique of multiple matchers can enhance the matching effect of the algorithm. For deep reviews of other meta-heuristics based methods, consider the following articles: utilizing Memetic algorithm through partial reference alignment to evaluate candidate solution^[20], using Memetic algorithm through MatchFmeasure and Unanimous improvement ratio to evaluate candidate solution^[21], using the Tabu search improved compact evolutionary algorithm to match sensor ontologies^[22], and matching ontology based on instance by using Memetic algorithm^[23]. These methods also only focus on solving the problem of relying on prior knowledge and improving the quality of ontology alignment without considering the execution efficiency of meta-heuristic algorithms.

In this paper, the well-distributed and the fast-converged set of solutions for multi-objective meta-heuristic algorithms are considered. Two different GOA are proposed for multi-objective ontology alignment.

2 Preliminaries

In this section, the ontology, ontology alignment, and two basic concepts related to MOOP are defined as follows.

2.1 Ontology and Ontology Alignment

Definition 1 (Ontology) An ontology is formalized as $O = \langle C, P, I, V, R \rangle$, where:

- 1) C represents a collection of classes;
- 2) $P = \{P_1, P_2, \dots, P_n\}$ is a collection of properties;
- 3) $I = \{I_1, I_2, \dots, I_n\}$ represents instance information of concepts;
- 4) V indicates the annotations information of concepts;
- 5) $R = \{R_1, R_2, \dots, R_n\}$ represents axioms of concepts.

Definition 2 (Ontology alignment)

Ontology alignment is a task of finding a set of pair of entities between different ontologies. In this work, a pair of entities is also called correspondence, matching or mapping. The algorithm functioned to achieve this task can be defined as follows. This algorithm takes three parameters O_1, O_2 , and P as input and returns an alignment M , which is a set of finding correspondences. The O_1 and O_2 are two ontologies, and P is a set of parameters^[24].

$$M = F(O_1, O_2, P) \quad (1)$$

Definition 3 (Pareto dominance)

Assuming \mathbf{x} and \mathbf{y} are any two solution vectors in the evolutionary population, then if \mathbf{x} is called to dominate \mathbf{y} (denoted by $\mathbf{x} \prec \mathbf{y}$), it must meet two conditions: For all sub-objectives, \mathbf{x} is not worse than \mathbf{y} and at least one sub-objective makes that \mathbf{x} is better than \mathbf{y} . This mathematical model is defined as follows.

$$\begin{aligned} & (\forall k \in \{1, 2, \dots, r\}: f_k(\mathbf{x}) \leq f_k(\mathbf{y})) \wedge \\ & (\exists l \in \{1, 2, \dots, r\}: f_l(\mathbf{x}) < f_l(\mathbf{y})) \end{aligned} \quad (2)$$

Definition 4 (Pareto optimal solution) A solution \mathbf{x} is said a Pareto optimal solution because it is not dominated by any vector in the decision space Ω . The Pareto optimal solution is mathematically defined as follows:

$$\nexists \mathbf{y} \in \Omega : \mathbf{y} \prec \mathbf{x} \quad (3)$$

Definition 5 (Pareto optimal solution set) A Pareto optimal solution set (POS) is composed of multiple Pareto optimal solutions^[25].

$$\text{POS} = \{ \mathbf{x} \mid \nexists \mathbf{y} \in \Omega : \mathbf{y} \prec \mathbf{x} \} \quad (4)$$

2.2 Multi-Objective Grasshopper Optimization Algorithm

In 2018, Mirjalili *et al*^[26] proposed a multi-objective grasshopper optimization algorithm inspired from the behavior of grasshoppers swarms. In order to solve the optimization problem, a mathematical model was proposed by simulating the foraging behavior of grasshopper swarms. In this model, a social interaction function realizes the attraction and repulsion between grasshopper swarms. Furthermore, this repulsive force realizes the exploration process of GOA in a search space, and this attraction force realizes the exploitation and convergence process. In order to balance these two search processes, an adaptive comfort zone coefficient leads to an appropriate balance between exploration and exploitation. In the multi-objective version, the Pareto dominance, an archive and target selection techniques are introduced into the algorithm to optimize a Pareto optimal solution set. Further, the Pareto optimal dominance is integrated into GOA to compare solutions with multiple objectives. In order to obtain a non-dominated solution set, i.e., Pareto optimal solution set, an external archiving technique is used to store the non-dominated solutions and is updated after the population evolves. The mathematical model is defined as follows:

$$\mathbf{X}_i = c_w \left(\sum_{\substack{j=1 \\ j \neq i}}^N c_n \frac{p-q}{2} S(d_{ij}) \widehat{d}_{ij} \right) + \widehat{T}_d \quad (5)$$

\mathbf{X}_i is the updated position vector of the i -th grasshopper; c_w is similar to the inertia weight of particle swarm optimization, which balances the exploration and exploitation; p and q are the upper and lower bounds of the search range; \widehat{T}_d is the currently searched optimal target value, and \widehat{d}_{ij} is a unit vector from the i -th grasshopper to j -th grasshopper; S is the attraction function which is defined as follows.

$$S(d_{ij}) = u e^{\frac{-d_{ij}}{\tau}} - e^{-d_{ij}} \quad (6)$$

where d_{ij} is the distance between the i -th and the j -th grasshopper; u is the strength of attraction; τ indicates the attractive length scale. The unit vector \widehat{d}_{ij} is defined as follows:

$$\widehat{d}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{d_{ij}} \quad (7)$$

where \mathbf{x}_j is the current position vector of the j -th grasshopper; \mathbf{x}_i is the current position vector of the i -th grasshopper. In equation (5), c_n is an internal parameter,

which is the decreasing coefficient to shrink the comfort zone, repulsion zone and attraction zone; The c_w and c_n is defined as follows:

$$c_w = c_n = c_{\max} - t \frac{c_{\max} - c_{\min}}{L} \quad (8)$$

where c_{\max} and c_{\min} indicate the maximum value and minimum value of the parameter c , respectively; t is the current iteration number; L is the maximum number of iterations.

3 Mathematical Model of the Multi-Objective Ontology Alignment Method

According to the above analysis, the approach using meta-heuristics needs to be modeled as multi-objective optimization problem to solve the ontology alignment problem in a more suitable way. A mathematical model for the multi-objective ontology matching problem is defined. Subsequently, the data structure of the grasshopper is represented, and two objective functions are introduced. The mathematical model of the multi-objective optimization problem is defined as follows:

Given a vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$, $\mathbf{X} \in \Omega$ and it satisfies the following conditions:

$$g_i(\mathbf{X}) \geq 0, \quad i = 1, 2, \dots, k$$

$$h_l(\mathbf{X}) = 0, \quad l = 1, 2, \dots, l$$

Considering a maximization problem for ontology alignment, there are k conflicting objectives to be optimized. This optimization function is defined as follows:

$$\text{Maximize } f(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_k(\mathbf{X})) \quad (9)$$

For a candidate solution in the population, each grasshopper position also represents a set of weights used in the basic matcher aggregation task and is associated with a candidate alignment in the meta-matching method. The value of each weight indicates the contribution for a basic matcher. In order to adequately capture the information of the ontology, the matchers with different capabilities are used in the proposed method. Therefore, a candidate solution vector \mathbf{X} is represented as follows.

$$\begin{cases} \mathbf{X} = (r_1, r_2, \dots, r_n) \\ r_i = \sum_{i=1}^n w_i \times m_i(c_e), \text{ s.t. } \sum_{i=1}^n w_i = 1 \end{cases} \quad (10)$$

where c_e is a correspondence, w_i is the weight of the i -th

matcher, m_i ($i \in [1, n]$) is the i -th matcher, n is the number of matchers.

Inspired by Bock *et al*^[6], the transformation of ontology alignment task into an optimization problem requires two objectives: 1) the number of correspondences; 2) a set of correct correspondences. Therefore, two evaluation functions were proposed. Furthermore, we consider the matching probability of correspondences for a candidate alignment found during perform. It is formulated to the first objective, which is formally defined as follows:

$$f_1(\mathbf{X}) = \frac{|\mathbf{X}|}{|O_1 \times O_2|} \quad (11)$$

In order to protect the elite individuals, it is necessary to maximize the average confidence of correspondences in an alignment, which is formally defined as follows:

$$f_2(\mathbf{X}) = \frac{\sum_{i=1}^{|\mathbf{X}|} r_i}{|\mathbf{X}|} \quad (12)$$

where $|\mathbf{X}|$ is the number of correspondences for an alignment found so far, r_i is the confidence weighted for a correspondence, and $|O_1 \times O_2|$ represents the total number of candidate correspondences between two ontologies.

In the next sections, two versions of the multi-objective grasshopper optimization algorithm are proposed to investigate the matching quality and calculation time for solving ontology alignment problem.

4 Proposed EMO-GOA Algorithm

Considering that the two essential components of a multi-objective optimization algorithm are the selecting of non-dominated solutions and the maintaining of the diversity of non-dominated solutions in the optimization process, we propose an archive update mechanism based on the ε -domination concept to improve the distribution and convergence speed. In the proposed EMO-GOA, a grid or hyper-box mechanism based on the ε -dominance concept is used to improve the original MOGOA which is only based on the Pareto dominance. The main reason for this modification is that in the steady-state MOEA experiment by ε -dominance, it has been proved to be a good compromise method in terms of approaching the Pareto optimal front, the diversity of the solution, and the well-converged^[27]. Furthermore, this mechanism

maintains the diversity of solutions in the archive by allowing only one solution to be displayed in each pre-allocated hyper-box. In EMO-GOA optimization process, the new solutions generated are compared with each member in the archive by using ε -dominance. First of all, an identification array $\{\mathbf{B}=(B_1, B_2, \dots, B_k)^T\}$ is assigned to each member in the archive, and it is defined as follows:

$$B_k(f) = \begin{cases} \lfloor \frac{f_k - f_k^{\min}}{\varepsilon_k} \rfloor, & \text{for minimizing } f_k \\ \lceil \frac{f_k - f_k^{\min}}{\varepsilon_k} \rceil, & \text{for maximizing } f_k \end{cases} \quad (13)$$

where k is the number of problem objectives, and f_k^{\min} is the minimum value of the k -th objective and ε_k is the allowable tolerance in the k -th objective.

Therefore, this mechanism divides the whole objective space into grid clusters. The identification array B_x of the new solution \mathbf{x} and the identification array B_a of each archive member a are calculated, respectively. This new solution \mathbf{x} is not accepted when the identification array B_a of any member a in the archive dominates B_x of the new solution \mathbf{x} . That is, the new solution \mathbf{x} is dominated by the archive member ε -dominance. Instead, this archive member a is deleted, and the new solution \mathbf{x} is accepted when the new solution B_x dominates the B_a of any archive member a . If neither of the above two situations occurs, it means that the new solution \mathbf{x} and the members in the archive are ε -non-dominance. Two cases need to be analyzed. If the new solution shares a hyper-box with an archive member, the general non-dominated relationships are used to detect them first. If this new solution \mathbf{x} dominates the archive member or it is non-dominated by the archive member and is closer to the \mathbf{B} vector than the archive member, then the new solution \mathbf{x} is retained and the archive member is removed from the archive. If the new solution \mathbf{x} does not share the hyper-box with any archive member, then the new solution \mathbf{x} is accepted [27].

The above analysis shows that each hyper-box is occupied by only one hyper-box at the Pareto optimal front. The ε -dominance concept have three advantages: 1) the maintenance of diversity; 2) consistency of the size of the final archive and the total number of Pareto optimal solutions; 3) the rapid convergence of Pareto optimal solutions. The EMO-GOA is summarized in Algorithm 1.

Algorithm 1: Summary of EMO-GOA

- 1: Load two ontologies O_1 and O_2
 - 2: Construct solution space by similarity measure
 - 3: Initialize maximum number of iterations(maxIter), c_{\max} , c_{\min} , the population $X_i (i = 1, 2, \dots, n)$;
 - 4: Calculate the fitness of each grasshopper by using Eqs. (11) and (12).
 - 5: Select the target T of population by using Eq.(2).
 - 6: **while** ($t < \text{maxIter}$)
 - 7: Update c using Eq. (8)
 - 8: **for** each search individual
 - 9: Normalize the distances between individuals in [1,4]
 - 10: Update the position of the current individual by using Eq. (5)
 - 11: **End for**
 - 12: **for each individual**
 - 13: Evaluate individual by using Eqs. (11) and (12)
 - 14: Update the archive by using the ε - dominance technique.
 - 15: **End for**
 - 16: Select the target from the archive by using the roulette wheel strategy.
 - 17: Update target position and fitness.
 - 18: $t=t+1$;
 - 19: **End while**
 - 20: **End**
-

5 Proposed NS-MOGOA Algorithm

In the proposed NS-MOGOA, a fast non-dominated sorting (FNS) algorithm is used to replace the calculation of non-dominated solutions in MOGOA, and to improve the performance of MOGOA. The simulation results of this FNS algorithm in NSGAI show that it can preserve the diversity of solutions and better convergence. This FNS algorithm mainly includes two operations: 1) Non-dominated ranking for the whole population; 2) Calculating the crowding distance of each individual [28]. The NS-MOGOA is summarized in Algorithm 2.

First, for the non-dominated ranking, there are two sub-operations that need to be performed. The first one is to count the number of dominating solutions p . It is also called the dominating number N_p ; the second operation saves the solutions dominated by this solution p . It is worth noting that the solution with a dominance number 0 is naturally called the first non-dominated level.

Algorithm 2: Summary of NS-MOGOA

```

1: Load two ontologies  $O_1$  and  $O_2$ 
2: Construct solution space by similarity measure
3: Initialize maximum number of iterations(maxIter),  $c_{max}$ ,  $c_{min}$ ,
the population  $X_i (i = 1, 2, \dots, n)$ ;
4: Calculate the fitness of each grasshopper by using Eqs. (11)
and (12).
5: Select the target  $T$  of population by using Eq.(2).
6: while ( $t < \text{maxIter}$ )
7:   Update  $c$  using Eq. (8)
8:   for each search individual
9:     Normalize the distances between individuals in [1,4]
10:    Update the position of the current individual by using
Eq. (5)
11:    Evaluate individual by using Eqs. (11) and (12)
12:   End for
13:   Update the population by using the fast non-dominated
sorting technique.
14:   Select the target from the individuals with lower
rankings by using the roulette wheel strategy.
15:   Update target position and fitness.// Update current target
16:    $t=t+1$ ;
17: End while
18: End

```

Next, the algorithm scans all solutions with $N_p=0$ and decrements their dominance numbers. When the dominance number of any solution q decreases by 0, the solution q is saved in a list Γ . The members from list Γ are considered the second non-dominant level. This operation continues until all levels are recognized.

For the calculation of crowding distance, this operation estimates the density around a particular solution in the population. Specifically, this operation sorts the population in ascending order according to each objective value. For each objective, the solution with the largest and smallest function value is assigned an infinite distance value. The intermediate solution is assigned a difference v , which is equal to the difference between the two neighboring solutions of this solution. This calculation is continued until all target values are reached. As a result, the crowding distance of each individual is the sum of each object difference v of each solution. Further, the crowding distance for the z -th solution is half the circumference of the cuboid. The method is depicted in Fig. 1.

In the selection process, the FNS algorithm prefer-

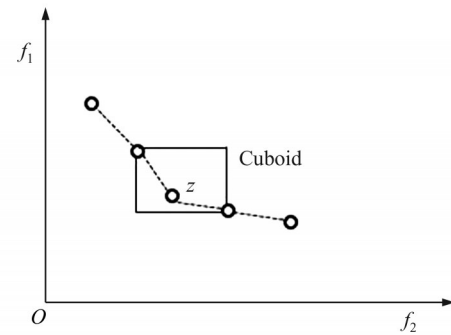


Fig. 1 The method of calculating for the crowding distance of the z -th solution

Note: The points linked by dashed lines indicate that they are at the same non-dominant level

entially selects the solution with a low non-dominated rank when two solutions with different non-dominated ranks. When these two solutions have the same ranking, the one in a spacious area (that is, a less crowded area) is selected first.

6 Experiments

In order to evaluate the performance of the improved algorithms in performing ontology alignment tasks, three experiments are performed. In the first experiment, the selection strategy for this target is studied. The second experiment compares traditional MOGOA with other commonly used meta-heuristic algorithms. In the third experiment, two improved methods were compared with the Ontology Alignment Evaluation Initiative (OAEI) methods.

6.1 Evaluation Methods and Dataset

The result of ontology alignment is a set of correspondences, which are usually evaluated using three indicators: precision, recall and F -measure from the Information Retrieval field^[29]. Precision represents the percentage of correctly matched mappings in all mappings found. Recall represents the percentage of correctly matched mappings in all correct mappings from the reference alignment. It is worth noting that if a system only retrieves the correct correspondence, the number of correct correspondence retrieved is reduced, but the total number of retrieved correspondences will be significantly reduced, which leads to a high precision. When the number of correct correspondences decreases, the recall becomes smaller. Therefore, precision and recall are two conflicting indicators. In the actual evaluation, it is not complete to evaluate the performance of a system by only applying the precision and the

recall, and they must be well combined^[9]. For this reason, the F -measure is used, which represents the weighted harmonic mean of precision and recall. In this experiment, precision is expressed as $Prec$, recall is expressed as Rec , F -measure is represented as $F1$.

$$Precision = \frac{|R_{rel} \cap A_{ret}|}{|A_{ret}|} \quad (14)$$

$$Recall = \frac{|R_{rel} \cap A_{ret}|}{|R_{rel}|} \quad (15)$$

$$F\text{-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (16)$$

where R_{rel} is the relevant mappings given by OAEI, and A_{ret} is the mappings retrieved by the system.

In order to study the performance of the proposed EMO-GOA and NS-MOGOA methods, twenty-three ontology alignment tasks are performed, which come from the OAEI. Since the maximum number of executions for the OAEI-2016 systems is five, we performed each task independently 5 times and calculated the average value for the fairness. These test tasks are described in Table 1.

Table 1 Brief of benchmarks

Ontology	Brief description of characteristics
101	Conventional reference ontology
103	With other language generalization
104	With other language restriction
203	No entity names and comments
204	Different naming conventions
205	The labels with synonyms and comments suppressed
221	An ontology with no specialization
222	An ontology with flattened hierarchy
223	An ontology with expanded hierarchy
224	An ontology with no instance
225	An ontology with no instance
228	An ontology with no properties
231	Multiplying entities
232	An ontology with no hierarchy and instance
233	An ontology with no hierarchy and property
236	An ontology with no instance and property
237	An ontology with flattened hierarchy and no instance
238	An ontology with expanded hierarchy and no instance
239	An ontology with flattened hierarchy and no property
240	An ontology with expanded hierarchy and no property
241	An ontology with no hierarchy, property and instance
246	An ontology with flattened hierarchy and no property and instance
247	An ontology with expanded hierarchy and no property and instance

6.2 Parameter Settings

In order to maintain an ontology alignment system with high accuracy and applicability to input different ontologies, in this work, basic matchers with different capabilities are used to calculate the similarity for a pair

of entities. The matcher means Jero measure^[30], Vector Space Model^[31], WordNet^[32], Levhenstein^[33], Hierarchy distance^[15] and Numberedhierarchy distance^[15], respectively. The parameter settings of each method are shown in Table 2.

Table 2 Parameters configuration of each method

Algorithm	Name of parameter	Value
EMO-GOA	The size of population (N)	30
	Maximum number of iterations (maxIter)	100
	The maximum value of c (c_{\max})	1
	The minimum value of c (c_{\min})	0.004
	The slack variable ε	0.001
	The strength of attraction u ;	0.5
	The attractive length scale τ ;	1.5
NS-MOGOA	Has the same parameters as EMO-GOA except that there are no slack variables ε	
MOGOA	Has the same parameters as EMO-GOA except that there are no slack variables ε	
SPEA2	The size of population (N)	30
	Maximum number of iterations (maxIter)	100
	The probability of crossover (P_c)	0.9
	The probability of mutation (P_m)	0.1
	The crossover distribution index for SBX (σ)	20
	The mutation distribution index for PM (H)	20
GA	The size of population (N)	30
	Maximum number of iterations (maxIter)	100
	The mutation rate (R_m)	0.2
	The crossover rate (R_c)	0.9
PSO	The size of population (N)	30
	Maximum number of iterations (maxIter)	100
	Learning factor (c_1)	2
	Learning factor (c_2)	2
	Inertia weight (w)	1.4

6.3 Experiment and Comparison

6.3.1 Target selection strategy

The next position of the grasshopper is moved by learning the current position, the target position and the positions of all other grasshoppers. Therefore, the choice of the target is an important component, which improves the distribution of non-dominated solutions in the archive and guides the grasshopper population toward promising regions. However, since the choice of this target needs to choose one from the Pareto optimal solution set, the choice of this target is a challenge in multi-objective MOGOA. In this work, we experimented with two selection techniques: roulette wheel selection

(RWS) technique and square root distance (SRD) technique^[34]. Therefore, in our work, the roulette selection strategy and the square root distance strategy are verified to select targets from the archive. For roulette selection, the author counts the number of neighbors of each solution in the archive based on a fixed distance as crowding distance. Subsequently, the members in the archive are ranked according to the crowding distance. The archive ranked is employed by the roulette selection strategy to select a target. For the square root distance, we calculated the distance between each member in the archive and the current target by using Eq. (17). The member with the smallest SRD is selected as the new tar-

get.

$$\text{SRD}(x_1, x_2) = \sum_{k=1}^2 \sqrt{|f_k(x_1) - f_k(x_2)|} \quad (17)$$

where k is the number of the problem objective.

The results of this experiment are shown in Table 3. As Table 3 shown, the RWS strategy is better than the SRD strategy in terms of matching accuracy and execution time. Therefore, RWS strategy is used to select the target that is suitable for ontology matching.

6.3.2 Comparison between proposed methods and the other meta-heuristic algorithms

In order to investigate the performance of the proposed EMO-GOA and NS-MOGOA methods, we summarized the average F -measure and time, and compared

them with the classic single- and multi-objective meta-heuristic algorithms. Table 4 shows the comparison results with the multi-objective meta-heuristic algorithms. For the average $F1$, this modified EMO-GOA and NS-MOGOA achieve better alignment quality compared with other multi-objective algorithms. The main reason is that EMO-GOA maintains the diversity in the archive by allowing only one solution in each pre-allocated hyper-box of the Pareto optimal front. Another reason is that it emphasizes non-dominated solutions. Compared with the original MOGOA, the NS-MOGOA can maintain a better solution diversity and better converge in the obtained non-dominated front. The SPEA2^[27] obtains a poor alignment quality. Considering the precision for Eq.

Table 3 Comparison of target selection strategy

ID	SRD				RWS			
	Prec	$F1$	Rec	Time/ms	Prec	$F1$	Rec	Time/ms
101	1.000	1.000	1.000	2144.2	1.000	1.000	1.000	2027.6
103	1.000	1.000	1.000	2116.2	1.000	1.000	1.000	2002.2
104	1.000	1.000	1.000	2135.4	1.000	1.000	1.000	2098
203	1.000	1.000	1.000	1873.8	1.000	1.000	1.000	1830.8
204	1.000	0.997	0.994	2091.4	1.000	0.990	0.988	2185.2
205	0.914	0.771	0.668	2418.8	1.000	0.850	0.740	2340.6
221	1.000	1.000	1.000	2061.6	1.000	1.000	1.000	2014.6
222	1.000	1.000	1.000	2011.2	1.000	1.000	1.000	2007.8
223	0.983	0.975	0.967	2219	0.980	0.980	0.980	2240.8
224	1.000	1.000	1.000	2039.2	1.000	1.000	1.000	2025.4
225	1.000	1.000	1.000	2026	1.000	0.999	0.998	2027.2
228	1.000	0.997	0.994	1218.6	1.000	1.000	1.000	1153.8
231	1.000	0.999	0.998	2037.6	1.000	1.000	1.000	2082.2
232	1.000	0.994	0.988	2071.4	1.000	1.000	1.000	2056.4
233	1.000	1.000	1.000	1164.6	1.000	1.000	1.000	1194.6
236	1.000	1.000	1.000	1137.6	1.000	1.000	1.000	1170.2
237	1.000	1.000	1.000	2055.2	1.000	1.000	1.000	2121.6
238	0.988	0.984	0.981	2322	0.986	0.983	0.981	2304.6
239	0.967	0.983	1.000	1413	0.967	0.983	1.000	1368.6
240	0.941	0.955	0.970	1704.2	0.940	0.955	0.970	1649.8
241	1.000	1.000	1.000	1150.8	1.000	1.000	1.000	1153
246	0.967	0.983	1.000	1339.4	0.970	0.980	1.000	1372.4
247	0.912	0.925	0.939	1635.8	0.910	0.925	0.939	1651.2
Avg	0.986	0.981	0.978	1842.9	0.989	0.985	0.982	1829.5

Table 4 Comparison with other multi-objective metaheuristic algorithms on 23 matching tasks

ID	EMO-GOA			NS-MOGOA			MOGOA			SPEA2		
	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec
101	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.820	0.846	0.870
103	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.830	0.870	0.920
104	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.810	0.850	0.900
203	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.770	0.810	0.850
204	0.960	0.960	0.960	1.000	0.990	0.988	0.920	0.920	0.920	0.810	0.860	0.910
205	0.940	0.890	0.850	1.000	0.850	0.740	0.850	0.820	0.790	0.370	0.380	0.390
221	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.850	0.890	0.940
222	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.830	0.870	0.920
223	0.980	0.980	0.970	0.980	0.980	0.980	0.980	0.980	0.980	0.820	0.860	0.910
224	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.840	0.880	0.930
225	1.000	1.000	1.000	1.000	0.999	0.998	1.000	1.000	1.000	0.840	0.880	0.930
228	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	0.860	1.000
231	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.840	0.880	0.930
232	1.000	1.000	1.000	1.000	1.000	1.000	0.996	0.996	0.996	0.840	0.890	0.930
233	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.770	0.870	1.000
236	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	0.860	1.000
237	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.820	0.860	0.920
238	0.980	0.980	0.980	0.986	0.983	0.981	0.980	0.980	0.980	0.800	0.840	0.890
239	0.970	0.980	1.000	0.967	0.983	1.000	0.970	0.980	1.000	0.690	0.810	0.970
240	0.940	0.960	0.970	0.940	0.955	0.970	0.940	0.960	0.970	0.680	0.780	0.910
241	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.770	0.870	1.000
246	0.970	0.980	1.000	0.970	0.980	1.000	0.970	0.980	1.000	0.670	0.790	0.950
247	0.910	0.930	0.940	0.910	0.925	0.939	0.910	0.930	0.940	0.680	0.770	0.900
Avg	0.985	0.985	0.986	0.989	0.985	0.982	0.979	0.980	0.982	0.767	0.829	0.907

(14), if we select as many correct correspondences as possible, that is, the precision is high, then only the most likely correct correspondences will be selected. In this case, the total number of correspondences selected by the algorithm will be reduced, and the correct correspondences selected by the algorithm will be correspondingly

reduced, but the denominator is reduced faster so that the precision becomes higher. The precision of EMO-GOA and NS-MOGOA are 0.985 and 0.989, respectively, which shows that, in most cases, the performance of the proposed methods to find the correct correspondence is enhanced.

Comparison results with the single-objective algorithm in Table 5 shows that the proposed EMO-GOA and NS-MOGOA outperform the GA-based and PSO-based methods. The main reason is that multi-objective optimization algorithms can maintain a better solution diversity and better converge in the obtained non-dominated front. Furthermore, the multi-objective optimization algorithm has better performance than single-objective for ontology alignment problem.

Table 6 shows the average running time of each algorithm. It is clear that the time of the proposed

EMO-GOA is much less than NS-MOGOA, MOGOA, SPEA2, and GA. The main reason is that EMO-GOA employs the technology of ε -dominance to update the archive. This mechanism can reduce the time to converge to the Pareto optimal solution set and find a high-quality alignment. Although the running time of NS-MOGOA is higher than that of EMO-GOA, it is lower than MOGOA. Furthermore, this NS-MOGOA modified by employing a fast non-dominated solution sorting can speed up the convergence of non-dominated solutions and better maintain the diversity of solutions.

Table 5 Comparison results with other single-objective metaheuristic algorithms on 23 matching tasks

ID	EMO-GOA			NS-MOGOA			GA			PSO		
	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec
101	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.970	0.950	1.000	1.000	1.000
103	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.740	0.600	1.000	1.000	1.000
104	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.970	0.950	1.000	1.000	1.000
203	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.970	0.950	1.000	1.000	1.000
204	0.960	0.960	0.960	1.000	0.990	0.988	1.000	0.940	0.90	1.000	1.000	1.000
205	0.940	0.890	0.850	1.000	0.850	0.740	1.000	0.710	0.560	0.870	0.870	0.860
221	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.830	0.720	1.000	1.000	1.000
222	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.950	0.92	0.970	0.970	0.970
223	0.980	0.980	0.970	0.980	0.980	0.980	0.980	0.860	0.79	0.970	0.970	0.970
224	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.920	0.860	1.000	1.000	1.000
225	1.000	1.000	1.000	1.000	0.999	0.998	1.000	0.850	0.770	1.000	1.000	1.000
228	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.997	0.994	1.000	1.000	1.000
231	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.998	1.000	1.000	1.000
232	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.850	0.740	1.000	1.000	1.000
233	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.810	0.760	1.000	1.000	1.000
236	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.997	0.994	1.000	1.000	1.000
237	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.970	0.950	0.970	0.970	0.970
238	0.980	0.980	0.980	0.986	0.983	0.981	0.990	0.890	0.820	0.986	0.986	0.986
239	0.970	0.980	1.000	0.967	0.983	1.000	0.960	0.950	0.930	0.890	0.900	0.920
240	0.940	0.960	0.970	0.940	0.955	0.970	0.970	0.900	0.850	0.940	0.950	0.960
241	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.920	0.850	1.000	1.000	1.000
246	0.970	0.980	1.000	0.970	0.980	1.000	0.960	0.950	0.940	0.880	0.890	0.910
247	0.910	0.930	0.940	0.910	0.925	0.939	0.940	0.900	0.870	0.890	0.900	0.920
Avg	0.985	0.985	0.986	0.989	0.985	0.982	0.991	0.906	0.855	0.972	0.974	0.977

Table 6 Comparison of the time with MOGOA, SPEA2, GA and PSO

ID	EMO-GOA	NS-MOGOAO	MOGOA	SPEA2	GA	PSO	ms
101	1 733.4	2 027.6	2 244.2	1 760.4	2 152.6	1 708.6	
103	1 940.6	2 002.2	2 416	1 850.6	2 063.2	1 613.2	
104	1 708.2	2 098	2 346	1 803.6	2 093.6	1 733.6	
203	1 494.2	1 830.8	2 099	1 814	2 018.4	1 549.6	
204	1 673	2 185.2	2 276.2	1 777.4	2 079.6	1 716.2	
205	1 925.4	2 340.6	2 246.6	1 748.4	2 101.6	1 689.8	
221	1 788.8	2 014.6	2 144.8	1 726.2	2 433.8	1 727.6	
222	1 745.2	2 007.8	2 302.2	1 734.2	1 982.6	1 709.2	
223	1 853	2 240.8	2 276.2	1 932.6	1 961	1 919	
224	1 671.2	2 025.4	2 287.2	1 813.8	1 821.8	1 735	
225	1 657.6	2 027.2	2 061.4	1 803	1 781.2	1 742	
228	1 012.2	1 153.8	1 240.2	1 398	1 059.4	1 092.2	
231	1 709.2	2082.2	2 193	1 792	1 945.4	1 663.4	
232	1 644.2	2 056.4	2 288.2	1 794.6	1966	1 739.2	
233	1 017	1 194.6	1 155.8	1 424.2	1 161	1 032.6	
236	1 017.4	1 170.2	1 427.8	1 377.6	1 128	1 046.4	
237	1 717.2	2 121.6	2 046.4	1 781	2 123.2	1 682	
238	1 877.6	2 304.6	2 403	1 868.6	2 227.4	1 928.6	
239	1 179	1 368.6	1 632.4	1 539.4	1 179.8	1 127.6	
240	1 425.8	1 649.8	1 807.4	1 715.6	1 475.2	1 359.2	
241	1 038.8	1 153	1 153.6	1 348	1 139.2	1 036.2	
246	1 173	1 372.4	1 542.6	1 518	1 323.4	1 126	
247	1 502.8	1 651.2	1 836.2	1 613	1 498.2	1 379.4	
Avg	1 543.69	1 829.50	1 975.06	1 692.79	1 770.24	1 524.2	

6.3.3 Comparison between proposed methods and the other OAEI methods

In order to further study the performance of the proposed methods, the top system AML^[35], LogMap^[36] and XMap^[37] in the OAEI-2016 competition are selected. It is worth noting that there are no results of tasks 103, 104, 203, 204, 205 and 231 on the OAEI official website. Therefore, these results are removed from our experimental results. Table 7 is the comparison of the proposed method with the top system of OAEI competition, and the results show that the proposed EMO-GOA and NS-MOGOAO methods significantly improve the quality of alignment.

7 Conclusion

The improvement of multi-objective evolutionary algorithms usually considers the two goals of maintaining the diversity of non-dominated solution sets and converging to the true Pareto optimal front. However, it is often ignored to achieve these two goals with rapid calculation. In this paper, the fast calculation manner for multi-objective meta-heuristic algorithms to align ontology are considered. Two different grasshopper optimization algorithms (GOA) are proposed for multi-objective ontology alignment. The performance of the proposed two methods is evaluated on the benchmark dataset. The results show that the proposed two methods outperform other meta-heuristic-based methods in terms of align-

Table 7 Comparison of the proposed method with the top system of OAEI competition

ID	AML			LogMap			XMap			NS-MOGOA			EMO-GOA		
	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1	Rec
101	1.00	0.00	0.00	0.94	0.95	0.96	0.94	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
221	1.00	0.51	0.34	0.94	0.94	0.95	0.94	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
222	1.00	0.50	0.33	0.00	0.00	0.00	0.94	0.78	0.67	1.00	1.00	1.00	1.00	1.00	1.00
223	1.00	0.51	0.34	0.94	0.94	0.95	0.94	0.97	1.00	0.98	0.98	0.98	0.98	0.98	0.97
224	1.00	0.51	0.34	0.94	0.94	0.95	0.94	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
225	1.00	0.51	0.34	0.94	0.95	0.96	0.94	0.97	1.00	1.00	0.999	0.998	1.00	1.00	1.00
228	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
232	1.00	0.51	0.34	0.94	0.94	0.95	0.94	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
233	1.00	1.00	1.00	0.85	0.92	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	0.85	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
237	1.00	0.50	0.33	0.00	0.00	0.00	0.94	0.78	0.67	1.00	1.00	1.00	1.00	1.00	1.00
238	1.00	0.51	0.34	0.94	0.95	0.96	0.94	0.97	1.00	0.986	0.983	0.981	0.98	0.98	0.98
239	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	0.967	0.983	1.00	0.97	0.98	1.00
240	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	0.94	0.955	0.97	0.94	0.96	0.97
241	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
246	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	0.97	0.98	1.00	0.97	0.98	1.00
247	1.00	1.00	1.00	0.85	0.92	1.00	1.00	1.00	1.00	0.91	0.925	0.939	0.91	0.93	0.94
Avg	1.00	0.71	0.63	0.79	0.82	0.86	0.97	0.96	0.96	0.99	0.99	0.98	0.99	0.99	0.99

ment quality. In particular, the EMO-GOA method achieves the best performance. For the running time, compared with SPEA2, GA, PSO, despite the long execution time of NS-MOGOA, the alignment quality has been improved. This EMO-GOA is only 0.019 49 seconds higher than PSO. It can be concluded that the two proposed multi-objective ontology alignment methods are successful for improving execution efficiency. In future work, we plan to further improve the performance of the proposed two methods in large ontology alignment.

References

- [1] Fernández S, Ito T. Semantic integration of sensor data with SSN ontology in a multi-agent architecture for intelligent transportation systems[J]. *IEICE Trans Inf Syst*, 2017, **12**: 2915-2922.
- [2] Zangeneh P, McCabe B. Ontology-based knowledge representation for industrial megaprojects analytics using linked data and the semantic web[J]. *Adv Eng Informatics*, 2020, **46**: 101164.
- [3] Talhi A, Fortineau V, Huet J C, *et al*. Ontology for cloud manufacturing based product lifecycle management[J]. *J Intell Manuf*, 2019, **30**(5) : 2171-2192.
- [4] Azevedo H, Belo J P R, Romero R A F. Using ontology as a strategy for modeling the interface between the cognitive and robotic systems[J]. *J Intell Robotic Syst*, 2020, **99**(3): 431-449.
- [5] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: Theory and application[J]. *Advances in Engineering Software*, 2017, **105**: 30-47.
- [6] Bock J, Hettenhausen J. Discrete particle swarm optimisation for ontology alignment[J]. *Inf Sci*, 2012, **192**: 152-173.
- [7] Wang J L, Ding Z J, Jing C J. GAOM: Genetic algorithm based ontology alignment[C]// *Asia-Pacific Services Computing Conference (APSCC)*. Washington D C: IEEE, 2006: 617-620.
- [8] Mohammadi M, Hofman W, Tan Y H. Simulated annealing-based ontology alignment[J]. *ACM Trans Manag Inf Syst*, 2019, **10**(1): 3:1-3:24.

- [9] Gil J M, Montes J F A. Evaluation of two heuristic approaches to solve the ontology meta-matching problem[J]. *Knowl Inf Syst*, 2011, **26**(2): 225-247.
- [10] Gil J M, Alba E, Montes J F A. Optimizing ontology alignments by using genetic algorithms [C]// *7th International Semantic Web Conference (ISWC)*. Berlin Heidelberg: Springer-Verlag, 2008: 1-15.
- [11] Naya J M V, Romero M M, Loureiro J P, *et al*. Improving ontology alignment through genetic algorithms[C]// *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*. Hershey: IGI Global, 2010: 240-259.
- [12] Ginsca A L, Iftene A. Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment[C]// *9th Roedunet International Conference (RoEduNet)*. Washington D C: IEEE, 2010: 118-122.
- [13] Xue X S, Wang Y P. Improving the efficiency of NSGA-II based ontology aligning technology[J]. *Data Knowl Eng*, 2017, **108**: 1-14.
- [14] Xue X S, Wang Y P, Hao W C. Using MOEA/D for optimizing ontology alignments[J]. *Soft Comput*, 2014, **18**(8): 1589-1601.
- [15] Acampora G, Loia V, Vitiello A. Enhancing ontology alignment through a memetic aggregation of similarity measures [J]. *Inf Sci*, 2013, **250**: 1-20.
- [16] Acampora G, Kaymak U, Loia V, *et al*. Applying NSGA-II for solving the ontology alignment problem[C]// *International Conference on Systems, Man, and Cybernetics(SMC)*. Washington D C: IEEE , 2013: 1098-1103.
- [17] Ryma G, Kholadi M K. Genetic algorithm with hill climbing for correspondences discovery in ontology mapping[J]. *J Inf Technol Res*, 2019, **12**(4):153-170.
- [18] Forsati R, Shamsfard M. Symbiosis of evolutionary and combinatorial ontology mapping approaches[J]. *Inf Sci*, 2016, **342**: 53-80.
- [19] Lv Z M, Peng R. A novel meta-matching approach for ontology alignment using grasshopper optimization[J]. *Knowl Based Syst*, 2020: 201-202. Article ID: 106050.
- [20] Xue X S, Wang Y P, Ren A H. Optimizing ontology alignment through memetic algorithm based on partial reference alignment[J]. *Expert Syst Appl*, 2014, **41**(7): 3213-3222.
- [21] Xue X S, Wang Y P. Optimizing ontology alignments through a memetic algorithm using both MatchFmeasure and unanimous improvement ratio[J]. *Artif Intell*, 2015, **223**: 65-81.
- [22] Xue X S, Wang Y P. Using compact evolutionary Tabu Search algorithm for matching sensor ontologies[J]. *Swarm Evol Comput*, 2019, **48**: 25-30.
- [23] Xue X S, Wang Y P. Using memetic algorithm for instance coreference resolution[J]. *IEEE Trans Knowl Data Eng*, 2016, **28**(2): 580-591.
- [24] Euzenat J, Shvaiko P. *Ontology Matching*[M]. Berlin Heidelberg: Springer-Verlag, 2013.
- [25] Das A K, Nikum A K, Krishnan S V, *et al*. Multi-objective Bonobo Optimizer (MOBO): An intelligent heuristic for multi-criteria optimization[J]. *Knowl Inf Syst*, 2020, **62**(11): 4407-4444.
- [26] Mirjalili S Z, Mirjalili S, Saremi S, *et al*. Grasshopper optimization algorithm for multi-objective optimization problems[J]. *Appl Intell*, 2018, **48** (4): 805-820.
- [27] Deb K, Mohan M, Mishra S. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions[J]. *Evol Comput*, 2005,**13**(4): 501-525.
- [28] Deb K, Agrawal S, Pratap A. *et al*. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Trans Evol Comput*, 2002, **6**(2): 182-197.
- [29] Yates R A B, Neto B A R. *Modern Information Retrieval*[M]. New York: ACM Press / Addison-Wesley ,1999.
- [30] Cohen W W, Ravikumar P, Fienberg S E. A comparison of string distance metrics for name-matching tasks [C]// *Proc of the KDD Workshop on Data Cleaning and Object Consolidation*. Washington D C: IEEE, 2003, **3**: 73-78.
- [31] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. *Commun ACM*, 1975, **18**(11): 613-620.
- [32] Miller G A. WordNet: A lexical database for English[J].*Communications of the ACM*, 1995, **38**(11): 39-41.
- [33] Scherbina A, Kuznetsov S. Clustering of Web sessions using Levenstein metric[J]// *Industrial Conference on Data Mining* . Berlin: Springer-Verlag, 2004: 127-133.
- [34] Saxena N, Mishra K K. Improved multi-objective particle swarm optimization algorithm for optimizing watermark strength in color image watermarking[J]. *Appl Intell*, 2017, **47**(2): 362-381.
- [35] Faria D, Pesquita C, Balasubramani B S, *et al*. OAEI 2016 results for AML[C]// *Proceedings of the 11th International Workshop on Ontology Matching*. Berlin: Springer-Verlag, 2016: 154-160.
- [36] Ruiz E J, Grau B C, Cross V V. LogMap family participation in the OAEI 2016[C]// *Proceedings of the 11th International Workshop on Ontology Matching*. Berlin: Springer-Verlag, 2016: 201-203.
- [37] Djeddi W E, Khadir M T, Yahia S B. XMap: results for OAEI 2016[C]// *Proceedings of the 11th International Workshop on Ontology Matching*, Berlin: Springer-Verlag, 2016: 222-226.

□