Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **9**, Number **3**, August **2024**

Efficient Ontology Meta-Matching Based on Anti-Distance Approximation Model Assisted Evolutionary Algorithm

Qi Wu, Qing Lv*

College of Electrical and Power Engineering
Taiyuan University of Technology, Taiyuan 030002, China
w19985757@163.com, lvqing_2006@163.com

Ziliang Yu

Faculty of Engineering, Architecture and Information Technology
The University of Queensland, Brisbane QLD 4072 Australia
ziliang.yu@uqconnect.edu.au

*Corresponding author: Qing Lv

Received October 10, 2023, revised January 13, 2024, accepted April 1, 2024.

Abstract. *Ontology plays a critical role for the representation and knowledge sharing. And the ensuing ontology heterogeneity problem often affects the interoperability among ontologies. Ontology Matching(OM) using similarity measures is a cutting-edge method for addressing the ontology heterogeneity problem. Ontology meta-matching(OMM) is proposed for optimising the aggregated multiple similarity measures' weights for improving the ontology alignment's quality. However, OMM based on Evolutionary Algorithm (EA) requires comparison with the reference alignment during the fitness evaluation, thus decreases the efficiency of the algorithm. For reducing EA's computing cost, an Anti-Distance Approximation Model(ADAM) assisted EA is proposed for forecasting the individuals' fitness. The model accomplishes the approximation of fitness values based on the distance relationship between individuals in the decision space. First, grid sampling method is used to collect solutions in the feasible domain, which will be evaluated by fitness function. Sampling solutions can not only help to analyze the fitness landscape of a local region and improve the accuracy of the approximation, but also can be used to construct ADAM. In the experiment, we used Benchmark and Anatomy provided by Ontology Alignment Evaluation Initiative to test EA-ADAM's performance comparing with EA and other cutting-edge systems. Experimental results demonstrate the algorithm's efficiency and effectiveness.*
Keywords: Ontology Meta-matching, Ontology Alignment, Evolutionary Algorithm, Expensive Optimization, Fitness Approximation

1. **Introduction.** Ontology describes domain knowledge and expresses the specification of the terms in the vocabulary [1], which is the modeling tool of Semantic Web(SW). According to [2], an ontology consists mainly of classes: describe objects with common characteristics in a specific domain(For example, "birds" can represent a class of all bird objects in animals); properties: describe the entities' relationship; instances: describe a specific object of the class(For example, "English" is an instance of the "Language" class); and the relationship between the elements in the ontology. Elements mentioned above are collectively called entities. With the development of the SW and the Knowledge

Graph, a large number of ontologies have emerged in various domain [3, 4]. However, different developers construct ontologies from their own perspective, which leads to the same knowledge's different representations, i.e. ontology heterogeneity problem [5].

Previous studies have shown that OM can find correspondences between heterogeneous ontologies to solve the ontology heterogeneity problem [6]. Specifically, OM utilizes similarity measures to evaluate the similarity and obtain the correspondence between entities from different ontologies, finally, output the matching result [7]. As the key of OM, similarity measure dedicates to evaluating the similarity of the entities from two ontologies [8]. However, only one similarity measures is difficult to provide the satisfactory results, which shows the necessity for aggregating multiple similarity measures to optimise OM's results [9]. OMM investigates how different similarity measures can be combined and debugged to determine high-quality matching results [10]. OMM first identifies different similarity matrices through similarity measures, and then assigns appropriate weights and thresholds to these matrices to obtain final matching results [11]. Since the weights and thresholds take real numbers in the range of [0,1], OMM problem is usually modeled as a class of continuous optimization problems. According to the previous studies, EA [12, 13] has been an effective method for addressing the OMM problem [14].

For OMM base on EA, since population requires a large number of fitness evaluation [15], which will deteriorate the efficiency of the algorithm. The most popular approach is to use surrogate-models instead of the expensive fitness evaluation process while guaranteeing the optimization ability of the algorithm, i.e., Surrogate-Assisted Evolutionary Algorithm(SAEA). Mainstream surrogate models such as Support Vector Machine(SVM) [16], Gaussian process regression (GPR) [17], and Radial Basis Function(RBF) [18] have been widely used in solving high cost problems. However, these models are based on the idea of regression, which often requires complicated training processes and high computational complexity. In contrast, similarity-based model utilizes inter-individual interrelationships to directly obtain an approximate mapping of the fitness function, which avoids the complex training process and is an efficient generalized model.

Most of the similarity-based methods face two main problems. First, such models rely on the distance of individuals in the solution space, i.e., the closer the Euclidean distance between two individuals, the more approximate their fitness. However, existing analyses of Fitness Landscape(FL) have shown that when the dimensionality of the solution space increases significantly, the accuracy of fitness predictions using distance relationships will be difficult to guarantee [19]. In some regions, it is possible that the fitness values of two neighboring individuals may vary dramatically, even if they are close in solution space. Therefore, in the process of estimating an individual using similarity-based models, it is necessary to analyze the FL in which the individual is located. One of the effective analytical methods is to extract sampling information in the feasible domain using sampling methods and then discretize the FL to represent it. Second, the individuals used to construct the model need to compute their fitness values, which is often done in an algorithm update, and these computations likewise deteriorate the algorithm's running time.

To perform the OMM task more efficiently while keeping the match quality as high as possible, a kind of similarity-based model, i.e., ADAM based on grid sampling is proposed, which is based on the distance relationship between the neighboring sampling individuals and the interested individuals in the solution space to accomplish fitness estimation, thus does not require a complex training process. In particular, we use grid sampling to divided the feasible domain into uniform grids. Each sampling point's fitness value will be calculated using fitness function. Sampling points will not only help analyze the roughness of the FL within a local region (i.e., relatively significant fitness variations within localized

regions), but they can also be used for fitness estimation. After that, the feasible domain is divided into different regions centered on each sampling point. For a new individual, first determine its region in the feasible domain, when it located in a region with a relatively rough fitness landscape (large fitness difference), then use the fitness function to calculate it. Otherwise, use the neighborhood sampling points to estimate it. On the one hand, it will reduce the running time of the algorithm, On the other hand, it can theoretically improve the accuracy of the estimation. Since the sampling process is done before the algorithm update, the problem of needing to accurately compute additional individuals to build the model during the algorithm update is solved. The main contributions are as follows:

- the FL is analyzed using grid sampling, when the FL where the individual is located is considered relatively rough, calculate it using the fitness function. Otherwise it will be estimated using ADAM;
- an ADAM based on grid sampling is presented to forecast the fitness of the individuals;
- an ADAM assisted EA is proposed for efficiently addressing the OMM problem.

The rest of this paper is organized as follows: Section 2 introduced the related work of this paper; Section 3 presents some basic conceptions of OM; Section 4 shows the construction of Anti-Distance Approximation Model(ADAM) assisted EA; Section 5 presents relevant results of experiments; Section 6 introduces the conclusion.

2. **Related Work.**

In this section, we will review the research process of OMM techniques and surrogate models used to solve expensive optimization problems in chronological order, respectively.

2.1. **Reviews of OMM Techniques.**

As the number of ontologies proliferates, the task of OM becomes more complex, so OMM combined multiple similarity measures has become a cutting-edge technology. The most popular approach is to model the OMM problem as an optimization problem and solve it by meta-heuristics [20].

Martinez-Gil et al. [15] proposed genetics for ontology alignments, which firstly used genetic algorithms to automatically establish the optimal weights of different similarity measures. After that Ginsca et al. [21] used EA to optimize three different types of similarity measures, i.e. syntactic-based, semantic-based and taxonomy-based, then used threshold screening for final alignment to improve the result's quality. In addition, for reducing the algorithm's running time, Xue et al. [22] optimized the ontology alignment using a Compact Evolutionary Algorithm (CEA) and obtained the experiment results with acceptable correctness and completeness. In order to prevent EA from falling into precocious convergence, Lv et al. [11] proposed a kind of adaptive pressure selection operator for optimizing EA-based ontology meta-matching. Besides EA, Particle Swarm Optimization(PSO) [23] has also become a common method for solving OMM problems due to its excellent performance and robustness. Semenova et al. [24] utilizes PSO to find appropriate weights for semantic measure metrics, enabling automatic tuning of parameters. In order to improve the quality of matching, Zhu et al. [25] proposed a Simulated Annealing(SA) PSO to optimize the parameters of multiple similarity measures and use SA strategy to help the algorithm jump out of the local optimal solution. Since researchers have different requirements on the focus of the matching results (e.g., the requirements on recall and precision in different research contexts), the single-objective OMM method is difficult to meet the matching requirements, and thus it is necessary to model the OM as a multi-objective OMM optimization problem. Xue et al. used NSGA-II [26] and MOEA/D [27] for maximizing three optimization objectives of OM. Semenova et

al. [28] proposed MOPSO-based OMM system. In addition, other meta-heuristic methods such as grasshopper algorithm [29] and firefly algorithm [30] have been successively applied to solve OMM problem. Besides meta-heuristics, Machine Learning(ML)[31] has also become an advanced method. Xue et al. [32] proposed a new OMM framework based on Reinforcement Learning(RL) [33] to obtain high-quality alignment.

It can be seen that most of the meta-heuristic algorithms around OMM are aimed at the quality of the alignment, while ignoring the time consumption of the matching system, which makes it difficult for conventional techniques to perform well. Therefore, surrogate-based optimization has become a popular research, which tends to reduce the number of real fitness calculations to improve the efficiency of the algorithm while ensuring excellent optimization capabilities of the algorithms.

2.2. **Surrogate-based Optimization.** Many scholars have made great contributions in the field of surrogate-based optimization, proposing a series of efficient surrogate-models.

Classical surrogate-models include Polynomial Regression(PR) [34], SVM, Kriging [35], RBF, and Neural Network(NN) [36, 37]. These models are based on the idea of regression, which greatly reduces the computational cost of the algorithms in solving costly problems. Bernaridno et al. [38] applied a linear regression method with weights to a clone selection algorithm to improve its performance in high-cost problems. Ciccazzo et al. [39] applied SVM to the simulation of circuit analysis with satisfactory results. Liu et al. [40] proposed a gaussian process surrogate model assisted EA to solve the medium-scale expensive problem. RBF models for large-scale optimization problems with constraints greatly greatly improves the efficiency of the evolutionary algorithm [41]. Graning et al. [42] introduced NN to EA and verified the performance improvement. Meanwhile, since each model has its own advantages and disadvantages [43], some studies such as [44] have used multiple surrogate-models simultaneously to enhance the robustness of the model. Unlike the regression-based model described above, similarity-based models utilize interrelationships between individuals to accomplish the forecast. The most representative of these is Fitness Inheritance(FI) [45]. FI was first proposed by Smith et al., since a new individual evolves from its parent individual, then in some cases the approximate fitness value of the new individual can be obtained from its parent individuals. Inspired by Smith, many models based on interrelationships between individuals have been produced, such as [46], [47]. This type of model is highly dependent on the similarity between individuals and is often widely applied to localized regions of the problem.

As can be seen above, research on OMM techniques based on meta-heuristic algorithms dominates. However, for classical OMM techniques based on EA, excessive computational consumption in fitness evaluation process will deteriorate the efficiency of the algorithm. Existing surrogate-models such as RBF, NN, etc. often require data for training, which causes extra computationally expensive. As for the existing similarity-based models, over-reliance on similarity between individuals makes forecast accuracy difficult to guarantee. To improve the efficiency of EA-based OMM, an EA-ADAM has been proposed, which utilizes the neighborhood solutions in the feasible domain to construct ADAM to estimate new individuals in order to reduce the number of fitness calculations. In particular, we use grid sampling to obtain the neighborhood solutions and use these solutions' fitness to analyze the roughness of FL and then determine whether to estimate it or not, which will theoretically improve the accuracy of the estimation.

3. **Preliminaries.**

3.1. **Ontology and Ontology Alignment.** Ontologies describe a specific domain's knowledge by defining concepts, properties, instances, and the relationships of them. An ontology can be defined as follows:

Definition 3.1 (Ontology)**.** *An ontology can be regarded as a 4-tuple Onto = (C, P, I, R), where: C, P, I represent the nonempty set of classes, properties, instances respectively; and R represents the relationship of the entities from the ontologies.*

For solving the ontology heterogeneity problem, ontology matching process is the cutting-edge technique for determining the ontology alignment, which is defined as follows:

Definition 3.2 (Ontology Alignment)**.** *Ontology alignment can be seen as a 5-tuple (id, e_1, e_2, confidence, relation), where: id represents the matching element's identifier; e_1, e_2 represent the entities from O_1 and O_2, respectively; confidence represents matched element's confidence, range [0,1]; relation Reflects the relationship of e_1, e_2, such as equivalence, generalization, etc.*

3.2. **Similarity Measure and Aggregation.** Similarity measures are important techniques for calculating the similarity value of mappings. The mapping with higher similarity value, the more it will be regarded as the correct mapping and will be saved. Therefore, the selection of similarity measures will directly affect the matching result. When performing ontology matching tasks, linguistic-based and syntax-based similarity measures are widely selected.

Linguistic-based method utilize synonyms or superlatives of two words to discover correspondences between entities, often with the help of dictionaries or thesauri. Among them, Wu and Palmer method [48] utilizing WordNet [49], an electronic lexical database, is widely used as an effective similarity measure. As for syntax-based similarity measures, the classical technique N-gram [50] is selected. In addition, Bidirectional Encoder Representations from Transformers(BERT) [51] is likewise considered, which has powerful semantic understanding and can better understand longer texts. However, during the processing of ontologies, longer passages of text appear less frequently. For computational efficiency reasons, the more efficient N-gram is selected. Similarity measures based on symbolic regression such as Levenshtein [52], Jaro [53], etc. are also widely used, however, these above methods only consider the parts of the two strings that are the same or different. In contrast, SMOA [54] considers both the same and different characters in two strings, which theoretically provides a better measure of the similarity of two strings. Therefore, we use Wu and Palmer method as linguistic-based similarity measure, and N-gram and SMOA as two syntax-based similarity measures.

Since the complexity of the ontology heterogeneity problem, a single similarity measure is difficult for guaranteeing the high-quality of matching results. It is necessary to integrate various similarity measures and thus improve the confidence of the alignment. The weighted sum method can be used to aggregate similarity measures [55], which can be defined as Equation (1):

$$s_{ag}(e_i, e_j) = \sum_{k=1}^{n} w_k \cdot s_k(e_i, e_j), subject\ to \sum_{k=1}^{n} w_k = 1 \tag{1}$$

where e_i and e_j represent two entities of two ontologies, w_k represents the aggregating weight of the kth similarity measure s_k. Assuming three similarity measures whose values are $s_1 = 0.61$, $s_2 = 0.54$ and $s_3 = 0.85$, respectively. Given the aggregating weight vector $(w_1, w_2, w_3) = (0.25, 0.35, 0.4)^T$, and the final similarity value is $\sum w_i \times s_i = w_1 \times s_1 + w_2 \times s_2 + w_3 \times s_3 \approx 0.68$.

3.3. **Ontology Meta-matching Problem.** Ontology meta-matching aims to obtain satisfactory alignment by automatically finding aggregated weights and thresholds for multiple similarity measures. To better assess the quality of the alignment, two metrics of the classical ontology meta-matching system GOAL [56] are referenced in this work, namely *recall* and *precision*. According to [57], *precision* calculates the percentage of true correct matching sequences and *recall* evaluates the ratio of the correct match sequences found to the all correct matches available, which can be defined as Equation (2) and Equation (3), respectively:

$$recall = \frac{|RA \bigcap A|}{|RA|} \tag{2}$$

$$precision = \frac{|RA \bigcap A|}{|A|} \tag{3}$$

where RA and A are the reference alignment and the alignment, respectively. The quality of ontology alignment is measured by a weighted summed average of recall and precision, i.e. $f - measure$, which can be defined as Equation (4):

$$f - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{4}$$

Based on above mentioned, the definition of ontology meta-matching is presented as follows:

$$\begin{cases} max \quad f - measure(W, T) \\ s.t. \quad W = (w_1, w_2, \cdots, w_{n-1})^T \\ \sum_{i=1}^{n-1} w_i = 1, \quad w_i \in [0, 1], \quad i = 1, 2, \cdots, n-1 \\ T \in [0, 1] \end{cases} \tag{5}$$

where W and T together comprise the n-dimensional decision variables, W represents the integrated weights of multiple similarity measures, and T represents the threshold used to filter low-quality mappings. The function $f - measure(W, T)$ is the f-measure value of the ontology matching result obtained from the parameter set W and T.

4. **Evolutionary Algorithm with Anti-distance Approximation Model.** Classic EA-based OMM needs to compare with reference alignment, which makes the algorithm consume long running time. Therefore, an approximation model is proposed to improve the efficiency of EA. Algorithm 1 presents the framework of EA-ADAM based OMM:

First, in the pre-processing phase, given two ontologies O_1 and O_2. Parse all entities from O_1 and O_2 separately. Then use the given three similarity measures to calculate the similarity of the parsed entities belonging to O_1 and O_2 respectively, and the three similarity matrices get from different similarity measures are obtained, this will be incorporated into the calculation of the fitness function.

During the initialization phase, generate s Sample Points(SPs) using grid sampling and compute their fitness values using the fitness function. These solutions and their corresponding fitness values will be stored in the archive A. The archive A will be set out an extra location to store the Historical Optimal Solution(HOS), and at initialization, HOS will be selected from the SPs to be stored. Then centered on each SP (excluding HOS), the feasible domain is divided into the same number of local sub-regions as the

Algorithm 1 EA-ADAM based OMM framework

Input: The number of maximum generation $MGen$;
Input: Related parameters;
Input: Reference alignment;
Input: Number of neighborhood individuals n;
Input: Number of samples s;
Pre-processing(O_1, O_2, similarity measures)
Sample s solutions using grid sample, and calculate their fitness using the fitness function;
Save the above solutions and their corresponding fitness values in the archive A;
Divide the feasible domain into sub-regions using SPs and mark the relative roughness of FL within each sub-region;
Select the solution from A with the highest fitness as the HOS;
Save the HOS into A(At this point there are two identical historically optimal solutions in A, but the location where the HOS is stored will be constantly updated with the algorithm);
Generate a population P of size M randomly;
Initialize the FOS, whose fitness value is set to 0;
for i=0; i <$MGen$; i++ **do**
  for j=0; j <M; j++ **do**
    Select two solutions closest to j_{th} individual from A;
    if If the selected solution is in a relatively rough region of the FL or the HOS is selected **then**
      calculate the j_{th} individual using the fitness function;
      Update the HOS;
    else
      Construct ADAM to estimate the j_{th} individual;
      Update the FOS;
    end if
  end for
  Calculate the FOS using the fitness function;
  Update the HOS;
  Perform select operations;
  Perform cross operations;
  Perform mutation operations;
end for
Output the HOS;
Decode the HOS and get the appropriate matching pair;
Output: recall, precision, f-measure.

SPs. And mark the location of the relative rough of FL within each localized region(if it exists).

During the EA optimization phase, first find the first two closest neighboring SPs for the new individual, which will help locate the region of it. Determine if the location of the new individual is marked, and if so, fitness approximation at this time will be considered inaccurate, thus use the fitness function to calculate it. If the selected solutions contain the HOS, it could mean that the new individual has the potential to become the HOS, in which case it will also be calculated using the fitness function. Otherwise, the selected solutions will be utilized to construct the ADAM to approximate the new individual. It is

worth noting that all new individuals that have been computed using the fitness function are used to update the HOS. In addition, since the value being estimated is obtained using a weighted average of the fitness values of the pre-designed SPs, then the estimate will be confined to the range of the fitness values of the SPs, i.e., the estimate will not exceed the maximum fitness value of the SPs at the maximum, which will hinder the algorithm's optimality search. Therefore, we will compute exactly the Forecast Optimal Solution (FOS) for each generation, regardless of whether its estimate value is better than the HOS or not.

4.1. **Encoding Mechanism.** Binary codes are used to store information. The encoding information contains weights of three similarity measures mentioned and one threshold value to filter lower similarity values, both of them will update in the iteration. Specifically, we use the definition of split points in [0, 1] to represent the weights. Assume that m represents the number of weights and the set of split points will be expressed as $sp' = \{sp'_1, sp'_2, \ldots, sp'_{m-1}\}$. When decode the information, the elements in sp' are first arranged in ascending order to obtain $sp = \{sp_1, sp_2, \ldots, sp_{m-1}\}$, and then the weights are calculated by Equation (6):

$$w_j = \begin{cases} sp_1, j = 1 \\ sp_j - sp_{j-1}, 1 < j < m \\ 1 - sp_{m-1} \end{cases} \tag{6}$$

The total length m of the individual code consists of an $m-1$ bit split point and a 1-bit threshold. Figure 1 illustrates an example of weight encoding and decoding, which includes five weights of five different similarity measures.

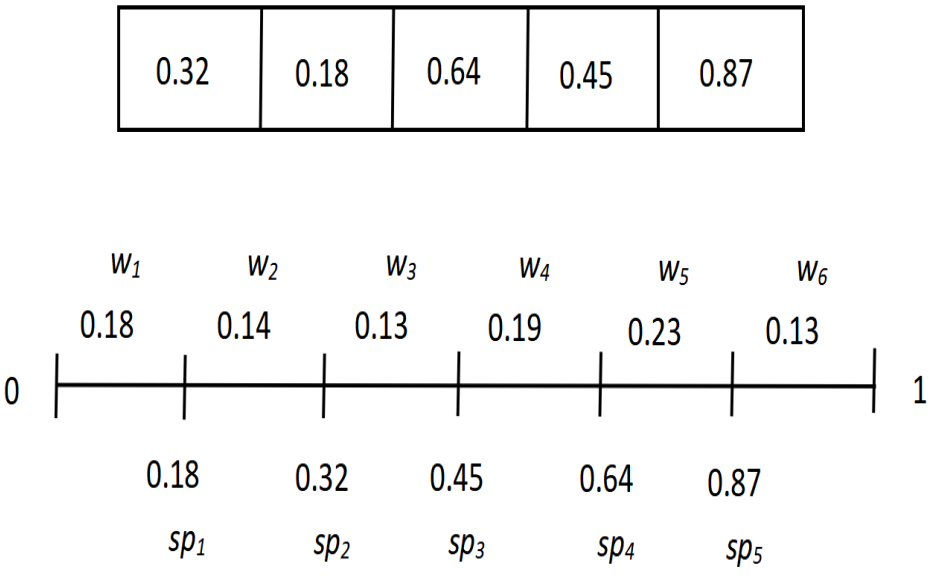

FIGURE 1. An example of weight coding and decoding.

From the given cutting points, the weights can be obtained as $w_1 = s_1 - 0 = 0.18$, $w_2 = sp_2 - sp_1 = 0.14$, $w_3 = sp_3 - sp_2 = 0.13$, $w_4 = sp_4 - sp_3 = 0.19$, $w_5 = sp_5 - sp_4 = 0.23$, $w_6 = 1 - sp_5 = 0.13$. The sum of all weights is 1.

4.2. **Anti-Distance Approximation Model Based on Grid Sampling.** Since the feasible domain is large and complex, it is difficult to forecast the fitness of the newly generated individuals, the classical method is to discretize the feasible domain into sampling points by sampling methods. Then calculate the fitness value of each SP using fitness function, and the fitness value of the new individuals will be estimated using the neighboring SPs. In order to ensure that the SPs can cover the whole feasible region uniformly in the case of low dimension, we use the grid sampling [58], which can theoretically reflect the characteristics of the feasible region well.

For the number of SPs, the algorithm's approximation precision and computational efficiency need to be weighed. In this work, the dimension of the feasible domain is 4. Suppose that n points are collected in each dimension, the final number of samples obtained is n^4. When n=2, the spacing of the samples is large, which will make the approximation precision lower; when n=4, then the final number of SPs can be obtained as 4^4=256, and such a large number of samples will lead to an increase in the amount of computation and deteriorate the computational efficiency. Therefore, we set n=3 to obtain 3^4=81 SPs. In addition, to ensure a uniform distribution of SPs, the three points collected in each dimensional [0,1] interval are 0.25, 0.50, and 0.75, respectively. Figure 2 illustrates an example of using grid sampling in two dimensions, we can get 9 SPs which correspond to a, b, ... , i nine coordinate points.

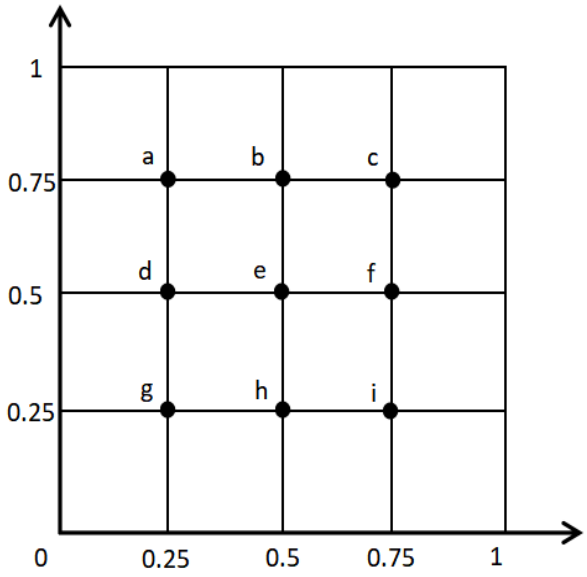

FIGURE 2. Example of sampling in two dimensions using grid sampling.

After SPs are collected, it is necessary to find the neighborhood individuals of the new individual whose fitness is to approximate. This approximation is based on the assumption that if two individuals are close enough in the decision space, they are equally close in the objective space [59]. However, when the FL is relatively rough, the accuracy of the approximation will be difficult to guarantee. FL is proposed by [60] and is used to describe a visual model of the relationship between fitness functions and decision variables for optimization problems in EA. The roughness of the FL is used to describe the degree of undulation or irregularity of the surface of the fitness function in an optimization problem. A rough local FL implies that the current local region of the fitness function varies dramatically and is not conducive to the prediction of fitness values.

Figure 3 presents the effect of different rough of the FL on the fitness approximation in a one-dimensional Michalewicz function. Where x_1 and x_3 represent two solutions calculated with fitness function, and x_2 represents a new solution with the fitness to be approximated. The vertical coordinate represents their real fitness. It can be seen that the FL between x_2 and x_3 is rougher than between x_1 and x_2. Even though the distance between x_2 and x_3 is also close, it is obvious that using the fitness of x_3 to estimate x_2 has a large error. In contrast, using x_1 to estimate x_2 yields satisfactory results.

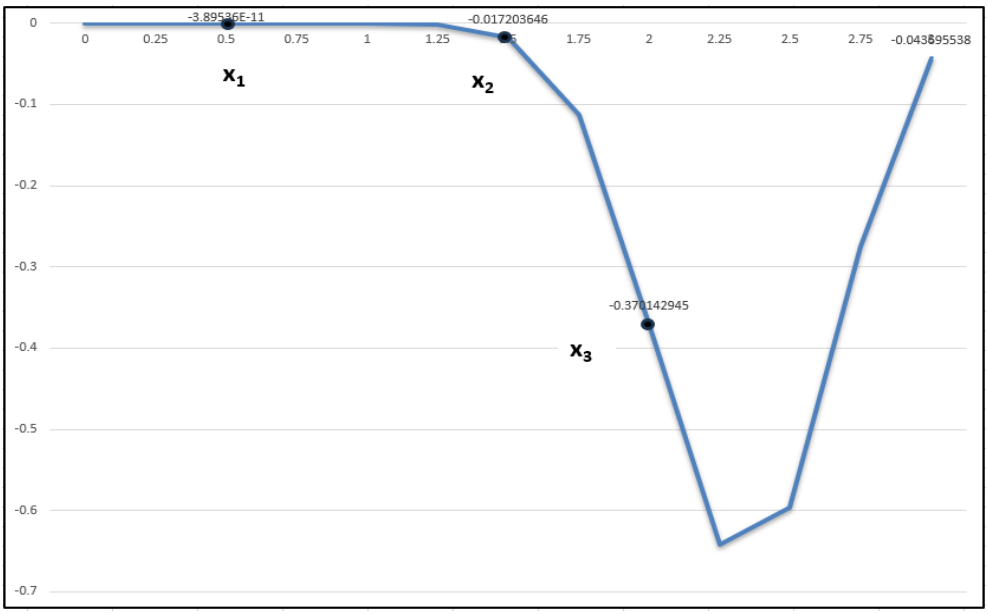

FIGURE 3. An example of three solutions.

Therefore, it is necessary to analyze the roughness of the FL in which x_i and its neighborhood individuals are located to determine whether to perform fitness approximation for x_i. Since the construction of the fitness approximation is only in a local region, it is more meaningful to analyze the roughness of the local FL compared to the global FL.

Before analyzing the FL, we need to determine the sub-region centered on each SP. In this work, each sub-region is formed with each SP as the center and its neighboring SPs as the boundary. All SPs in each sub-region consist of the central SP of that sub-region and the neighboring SPs of the central SP. Suppose the i_{th} SP is: $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$, its neighboring SPs can be seen as $(x_{i1} \pm st, x_{i2}, x_{i3}, x_{i4})$, $(x_{i1}, x_{i2} \pm st, x_{i3}, x_{i4})$, $(x_{i1}, x_{i2}, x_{i3} \pm st, x_{i4})$, $(x_{i1}, x_{i2}, x_{i3} \pm st, x_{i4})$, $(x_{i1}, x_{i2}, x_{i3}, x_{i4} \pm st)$ respectively, totaling 8 points, where st represents the sample interval, which in this work is 0.25. Note that when a dimension of x_{ij} is sampled with points on the boundary, there will be fewer than 8 points in its neighborhood. For example, if $x_{i1} = 0.25$, and $0.25\text{-}st = 0$, but 0 is not in the collected points.

Next, the FL in each sub-region needs to be analyzed. In each sub-region, most of the SPs are theoretically close to each other in terms of fitness value. If there is a SP with a large difference in fitness value from the rest of the SPs, the FL near that SP will be considered to be relatively rough, and estimation using that SP will be considered as inaccurate. Such SPs' fitness values will be called outliers.

In order to find the outliers, the Boxplot method that can effectively analyze data is used in this work. The specific process is as follows. First find the sub-region to be analyzed, suppose that the fitness values of all SPs it contains are in ascending order $\{f(sp_1), f(sp_2), ..., f(sp_n)\}$, where n is the number of SPs within that sub-region. Find

the lower quartile Q_1 of the whole set of data (i.e., the value corresponding to the quarter position in the whole set of data. If the total number of data is even, the quarter position is n/4; if the total number is odd, the quarter position is $(n+1)/4$) and the upper quartile Q_3 (i.e., the value that corresponds to the three-quarter position in the whole set of data. If the total number of data is even, the quarter position is $3n/4$; if the total number is odd, the quarter position is $(3n + 1)/4$). Then the non-anomalous range can be seen as: $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$, where IQR represents the interquartile range and $IQR = Q_3$ - Q_1. Fitness values that do not fall into this range can be considered outliers.

After that, given a new individual X, first find the SP closest to X and determine the sub-region where X locates. Then find the second closest SP to X within that sub-region, and if either of these two SPs' fitness value is an outlier, the FL in which X is located will be seen as relatively rough, then evaluate X exactly. If there are no outliers or HOS in the fitness values of these two SP's, this means that X can be estimated.

Suppose the two SPs found are SP_{first} and SP_{second}, their corresponding fitness values are $f(SP_{first})$ and $f(SP_{second})$ respectively. Then the f-measure for individual X can be calculated as follows:

$$f(X) = w_1 f(SP_{first}) + w_2 f(SP_{second}) \tag{7}$$

where w_1 and w_2 represent the weights of $f(SP_{first})$ and $f(SP_{second})$, respectively. Which can be calculated as Equation (8):

$$\begin{cases} w_1 = 1 - \dfrac{d(SP_{first}, X)}{d(SP_{first}, X) + d(SP_{second}, X)} \\ w_2 = 1 - \dfrac{d(SP_{second}, X)}{d(SP_{first}, X) + d(SP_{second}, X)} \end{cases} \tag{8}$$

where $d(SP_{first}, X)$ and $d(SP_{second}, X)$ are represent the distance in the decision space between X and SP_{first} and SP_{second}, respectively.

The Euclidean distance in Equation (9) which is a common method for calculating the distance between two points in geometric space, is selected to calculate the distance between two individuals:

$$d(I_1, I_2) = \sqrt{\sum_{i=1}^{D} (I_{1i} - I_{2i})^2} \tag{9}$$

where I_1 and I_2 represent two individuals, and D represents the dimensions of decision variables.

4.3. **Selection, Crossover and Mutation.** Selection is for screening out some better individuals from the current population as the next generation population using the fitness function, which commonly include roulette-wheel selection, tournament selection, truncation selection, Monte Carlo selection, etc. To ensure population diversity and the better individuals will be selected with higher probability. We used roulette selection, where the fitness of the individual is proportional to its probability of being selected.

Crossover is an important operation for ensuring populations' diversity. Two individuals in the are selected as parents for mating, i.e., their chromosomes are exchanged in a certain way for some of their genes, thus forming new individuals. In this work, we selected single-point crossover operation which is commonly used. A position on both parents' chromosomes is first randomly selected as a crossover point, and then gene translocation was performed on the paired chromosomes at that crossover position.

Mutation aims to help the algorithm to jump out of the local optimum solution during the optimization process. In this work, the common single point mutation was selected, which means that a single bit from the gene sequence needs to be mutated. In binary coding, changing 0 to 1 and 1 to 0.

5. **Experiment.**

5.1. **Experimental Configuration.**

In this paper, the classic Benchmark and Anatomy from OAEI [61] are selected to test the performance of EA-ADAM. OAEI is an internationally recognized platform for ontology alignment test and evaluation, aiming to promote the progress of ontology alignment by providing test datasets and giving evaluation metrics to compare the performance of various matching systems and techniques.

Benchmark developed by OAEI is able to distinguish between different matching methods to a certain extent, and discover the shortcomings of the system under test in an incremental way.The ontologies of Benchmark can be categorized into 1XX, 2XX, and 3XX according to the different heterogeneous types.1XX has the same ontology composition, and is usually used for conceptual testing; 2XX has ontologies with different lexical, linguistic, or structural features that are usually used to compare different modifications; and 3XX has ontologies developed by different organizations from the same domain in the real world.

Unlike Benchmark, Anatomy has two parts: mouse anatomy ontology and human anatomy ontology. The main characteristics of Anatomy are the larger size of the ontologies (the number of entities in both ontologies is around 3,000) and the higher difficulty of the matching (these ontologies are described using techniques that use conceptualized representations of natural language to a only limited extent), which will be challenges for the matching quality and efficiency.

For the fair comparison, we set EA-ADAM and EA to the same parameters, i.e.:

- $PopSize$: 20,
- $CrossoverP$: 0.6,
- $MutationP$: 0.01,
- $MGen$: 250,

where $PopSize$ represents population size which is related to of the decision variables' dimension. In this work, the dimension n of the decision variable is 4. According to existing studies [62], the range of $PopSize$ should be in $[4 \times n, 6 \times n]$, i.e. $[16, 24]$. If $PopSize$ is too large, the algorithm will be difficult to converge; if $PopSize$ is too small, the probability of early convergence will become high [63]. Combining the OMM is a small-scale problem, $PopSize$ is set to 20.

$CrossoverP$ and $MutationP$ represent crossover probability and mutation probability, respectively. The suggested ranges are $[0.6, 0.8]$ and $[0.01, 0.05]$, respectively. Too small a probability will reduce population diversity, and too high a probability will miss the optimal individuals [64]. Considering the low dimension of the OMM problem, we set $CrossoverP$ and $MutationP$ to 0.6 and 0.01.

$MGen$ represents Maximum generation. If $MGen$ is too small, it is difficult for the population to converge. At later stages of evolution, when the results of the algorithm hardly change, a larger $MGen$ will lead to a waste of time and storage space. Combining dimension n is set to 4, a relatively small maximum number of generations can be set, and we set $MGen$ to 120 which is robust enough in the experiments.

In this work, EA-ADAM and EA are compared from Table 1 in terms of recall, precision and f-measure mentioned in Section 3.3 correspond to R, P and F in the table, respectively. And the corresponding box-and-whisker plots in Figures 5-7. Then the running

time in Table 2 are compared to demonstrate the efficiency of our method. Table 1 and Table 2 show the average results of 30 independent runs of the algorithm. After that, EA-ADAM and cutting-edge techniques from OAEI are compared in Table 3 and Table 4.

5.2. **Experimental Results.** As shown in Table 1, EA-ADAM and EA's mean f-measure are 0.865 and 0.866, respectively. In particular, for further measuring the closeness of the results of the two algorithms, we calculated their mean difference, which is the mean of the absolute values of the differences between the two algorithms on each testing case. The results show that the mean difference of both methods is about 0.001, which indicates that the results of them are very close. *stDev* represents the standard deviation of 30 independent results, which can show the stability of the matching system. For 1XX, the f-measure of EA-ADAM can reach to 1.000, which shows the ability to perform the matching task excellently under simple heterogeneous context. For the more complex heterogeneous ontologies such as testing cases 2XX and 3XX and Anatomy, EA-ADAM can also achieve great alignments using a linguistic-based similarity and two syntax-based similarity measures. We need to point out that on testing cases 202, 265 and 266, the f-measure of EA-ADAM is relatively low. This is due to the fact that in these matching tasks, ontologies require the techniques using contextual information for finding high-quality correspondences, yet EA-ADAM uses no context-based similarity measures. In general, EA-ADAM demonstrated a acceptable degree of closeness to the EA results. Additionally, the relatively low mean standard deviation indicates that ADAM is not only effective in estimating the fitness of individuals, but also helps to enhance the stability of the algorithm. Figure 4 illustrates the convergence of EA-ADAM and EA with increasing number of iterations on Anatomy, where the red curve represents EA and the blue curve represents EA-ADAM. As can be seen, the blue curve is close to the red curve, which demonstrates the optimization ability of EA-ADAM and further illustrating the accuracy of ADAM forecast. Although the optimized results are still slightly inferior to EA, the running time of the algorithm can be reduced significantly (as will be explained later).

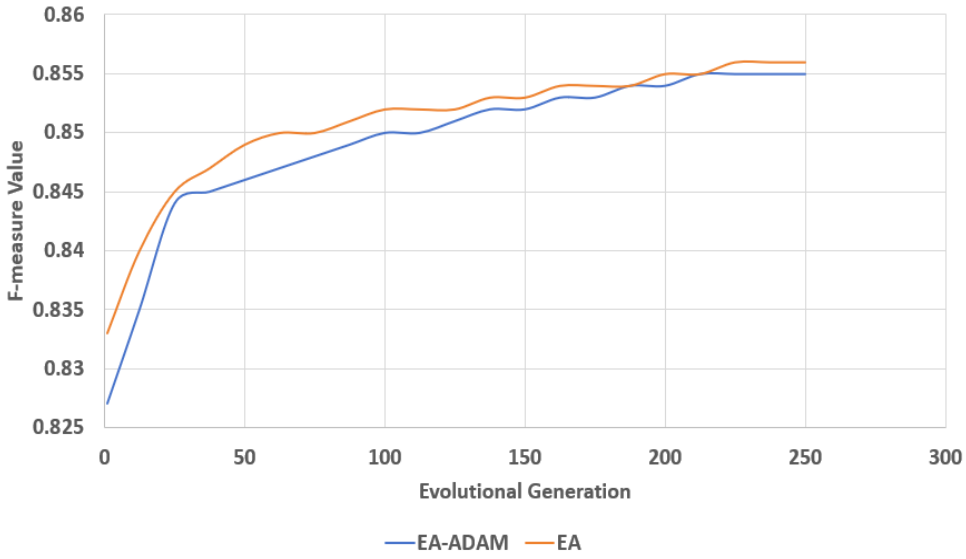

FIGURE 4. The convergence of EA-ADAM and EA.

Figures 5, 6, and 7 show three box plots for the two sets of data, corresponding to the results of precision, recall, and f-measure, respectively. The three sets of plots show that

TABLE 1. Comparison of EA-ADAM and EA on OAEI's Benchmark and Anatomy.

Testing Case	EA-ADAM	EA-ADAM	EA-ADAM	EA	EA	EA
	$P(stDev)$	$R(stDev)$	$F(stDev)$	$P(stDev)$	$R(stDev)$	$F(stDev)$
101	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
201	0.989 (0.000)	0.928 (0.000)	0.957 (0.000)	0.989 (0.000)	0.928 (0.000)	0.957 (0.000)
202	0.500 (0.000)	0.021 (0.000)	0.040 (0.000)	0.500 (0.000)	0.021 (0.000)	0.040 (0.000)
203	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
204	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
205	0.989 (0.000)	0.918 (0.000)	0.952 (0.000)	0.989 (0.000)	0.915 (0.005)	0.950 (0.003)
206	1.000 (0.000)	0.928 (0.000)	0.963 (0.000)	1.000 (0.000)	0.928 (0.000)	0.963 (0.000)
207	1.000 (0.000)	0.938 (0.000)	0.968 (0.000)	1.000 (0.000)	0.938 (0.000)	0.968 (0.000)
209	0.683 (0.013)	0.275 (0.005)	0.392 (0.003)	0.729 (0.008)	0.271 (0.005)	0.395 (0.005)
210	0.957 (0.027)	0.490 (0.007)	0.647 (0.003)	0.972 (0.017)	0.487 (0.004)	0.648 (0.002)
221	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
222	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
223	1.000 (0.000)	0.990 (0.000)	0.995 (0.005)	1.000 (0.000)	0.990 (0.000)	0.995 (0.000)
224	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
225	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
228	1.000 (0.000)	0.970 (0.000)	0.985 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
230	0.966 (0.025)	0.992 (0.007)	0.978 (0.010)	0.986 (0.000)	0.986 (0.000)	0.986 (0.000)
231	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
232	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
233	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
236	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
237	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.000)	1.000 (0.001)
238	0.990 (0.000)	0.990 (0.000)	0.990 (0.000)	0.990 (0.000)	0.990 (0.000)	0.990 (0.000)
239	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
240	1.000 (0.000)	0.970 (0.000)	0.985 (0.000)	1.000 (0.000)	0.970 (0.000)	0.985 (0.005)
241	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
246	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
247	1.000 (0.000)	0.970 (0.000)	0.985 (0.000)	1.000 (0.009)	0.970 (0.000)	0.985 (0.005)
248-2	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)
249-2	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)
250-2	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)
251-2	1.000 (0.000)	0.796 (0.000)	0.886 (0.000)	1.000 (0.000)	0.796 (0.000)	0.886 (0.000)
252-2	1.000 (0.005)	0.794 (0.000)	0.885 (0.002)	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)
253-2	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)
254-2	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)
257-2	1.000 (0.000)	0.788 (0.000)	0.881 (0.005)	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)
258-2	1.000 (0.005)	0.796 (0.000)	0.886 (0.000)	1.000 (0.000)	0.796 (0.000)	0.886 (0.000)
259-2	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)	1.000 (0.000)	0.794 (0.000)	0.885 (0.000)
260-2	1.000 (0.000)	0.793 (0.000)	0.885 (0.000)	1.000 (0.000)	0.793 (0.000)	0.885 (0.000)
261-2	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)
262-2	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)	1.000 (0.000)	0.788 (0.000)	0.881 (0.000)
265	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
266	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
301	0.980 (0.000)	0.814 (0.000)	0.889 (0.000)	0.980 (0.001)	0.814 (0.000)	0.889 (0.001)
302	1.000 (0.000)	0.604 (0.000)	0.753 (0.000)	1.000 (0.000)	0.604 (0.000)	0.753 (0.000)
303	0.889 (0.025)	0.821 (0.019)	0.853 (0.003)	0.895 (0.023)	0.821 (0.019)	0.856 (0.004)
Anatomy	0.941 (0.010)	0.784 (0.008)	0.855 (0.001)	0.944 (0.010)	0.783 (0.006)	0.856 (0.005)
Average	0.934	0.823	0.865	0.936	0.824	0.866

the results of EA-ADAM and EA are relatively similar, as evidenced by the fact that the upper and lower edges and the median of the two data sets in each plot are the same. The mean values of EA-ADAM in all three figures are lower than EA(corresponds to the fork in each plot), which indicates that there is still a difference between the results forecast by the model and the real calculated results.

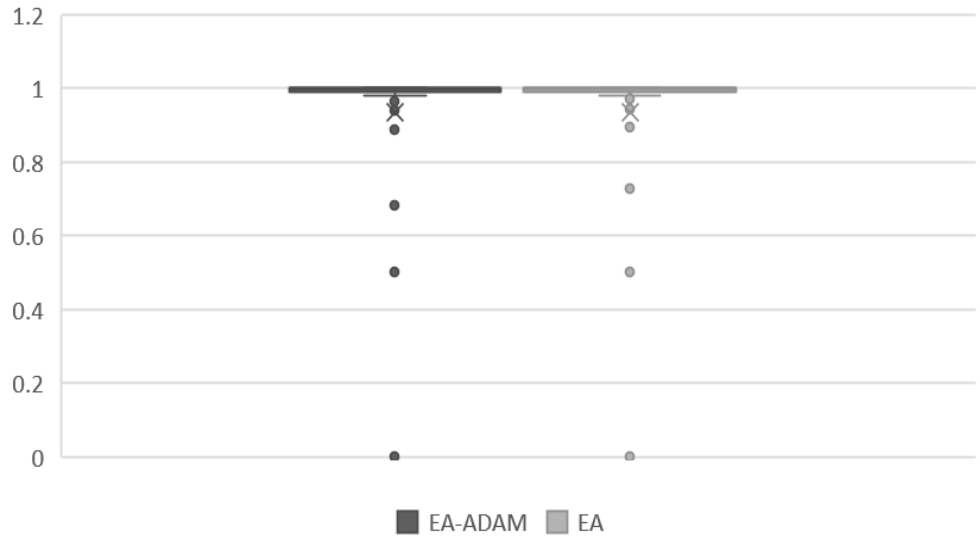

FIGURE 5. The Box-and-whisker Plot of precision.

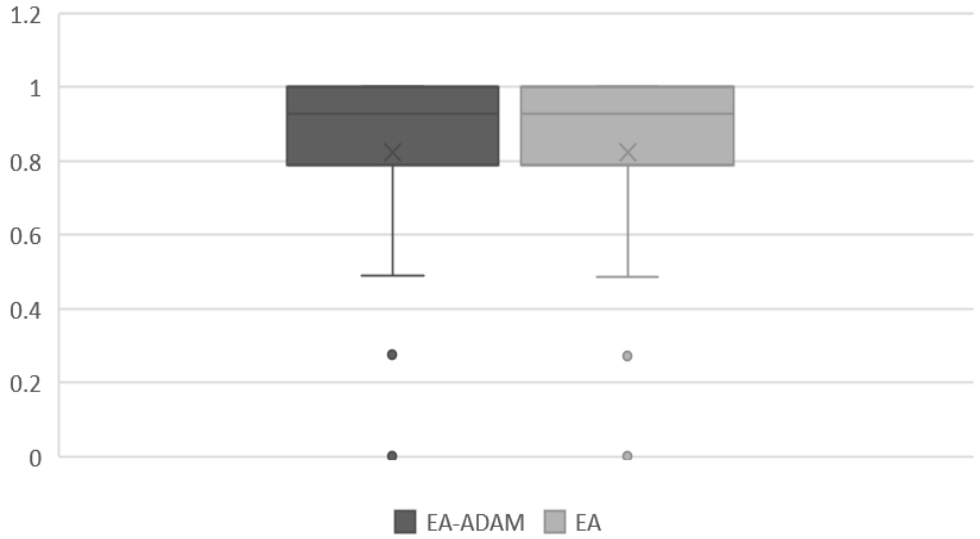

FIGURE 6. The Box-and-whisker Plot of recall.

In Table 2, the average running time of EA-ADAM and EA are 1652 and 6556 milliseconds on Benchmark, respectively. And the improvement degree is 74.8%. For Anatomy, EA-ADAM takes 77,286 milliseconds, while EA takes 504,156 milliseconds. The improvement degree is 84.7%. In the classical EA-based OMM technique, the evaluation of individuals needs to be compared with reference alignment, which consumes much running time. ADAM will reduce the number of fitness evaluations by estimating the individual's fitness and thus reduce EA's running time.

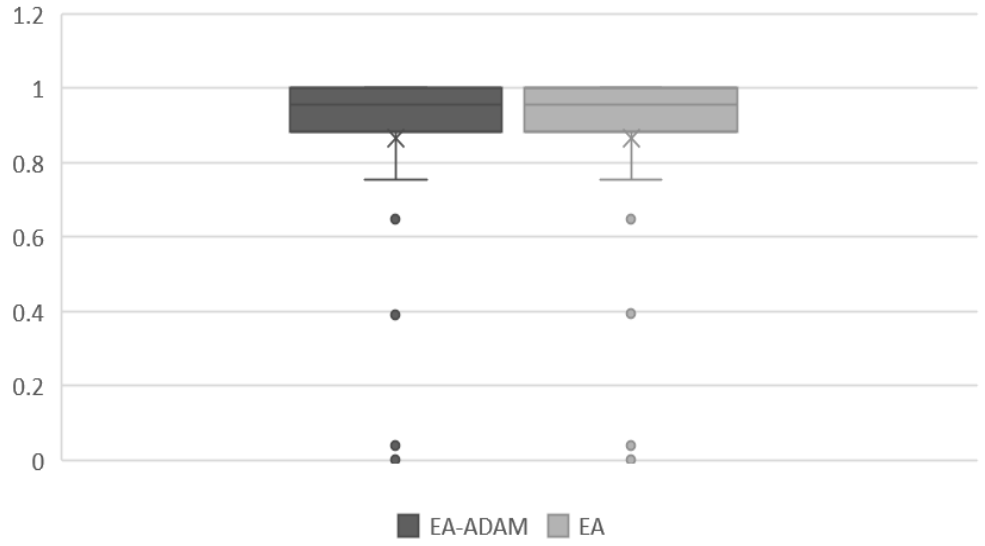

FIGURE 7. The Box-and-whisker Plot of f-measure.

Table 3 shows the comparison of EA-ADAM and other matching systems in terms of f-measure on Benchmark. By comparison, it can be found that EA-ADAM's results are second only to CroMatch and Lily. Further analysis of matching process is necessary. CroMatch proposed autoweight++ to aggregate multiple matchers, which enables a more comprehensive analysis of the degree of similarity between entities. However, as the number of matchers increases, the efficiency and memory of the system will be a concern. Lily needs to do much processing on the data to achieve alignment. Before matching, Lily reduces the impact of special test cases on the overall matching quality by means of preprocessing. In the post-processing stage, Lily utilizes ontology mapping debugging techniques to analyze and diagnose the mappings. In summary, CroMatch and Lily, need to integrate additional matchers or increase the processing procedures to improve the quality of the matching results, which increases its own size. EA-ADAM, although there is a gap between the matching quality and that of CroMatch and Lily, however, there is a greater advantage in terms of algorithmic lightweight. Compared to other matching systems, EA-ADAM shows the advantage of matching quality. In summary, the above results can illustrate the ability of EA-ADAM for obtaining satisfactory alignment.

Table 4 shows the comparison of EA-ADAM and other matching systems on Anatomy. In particular, we chose matching system which is also based on surrogate-model for comparison, i.e., IM-HEA [65]. The results show that EA-ADAM achieves higher matching quality. It has been found that IM-HEA does not add enough evaluation of the fitness function in the matching process and also does not analyze the effect of FL on the model forecast, which makes EA-ADAM perform better in the matching results. Compared with 11 other state-of-the-art matching systems, EA-ADAM ranks 5th, which indicates that our approach can also achieve acceptable results when dealing with large-scale ontologies. The above test results reflect the high matching quality of EA-ADAM when facing different sizes of ontologies.

6. **Conclusions.** Ontology is an advanced technology for describing domain knowledge. However, there is no uniform standard for ontology construction, and the resulting heterogeneity problem greatly hinders the sharing and interaction of knowledge. OM has become an advanced method for solving ontology heterogeneity. For EA-based OMM, the fitness evaluation of individuals requires the comparison of the reference alignment,

TABLE 2. Comparison of EA-ADAM and EA in terms of Running Time (millisecond).

Testing Case	**EA-ADAM**	**EA**
101	1299	7361
201	2719	7652
202	1524	7556
203	2210	7542
204	1670	7555
205	2139	7504
206	2337	7689
207	2910	7518
208	2303	7512
209	2224	7824
210	2232	7641
221	1440	7405
222	1451	7356
223	1598	7897
224	1342	7351
225	1446	7333
228	977	4793
230	1435	6284
231	1491	7425
232	1428	7468
233	961	7468
236	937	4914
237	1354	7461
238	1570	7450
239	907	4720
240	1047	4928
241	982	4852
246	1168	4660
247	1156	5398
248-2	2210	7776
249-2	2199	7630
250-2	1639	4805
251-2	2134	7422
252-2	2293	7600
253-2	2158	7696
254-2	1513	4812
257-2	1499	4982
258-2	2271	7830
259-2	2088	7696
260-2	1401	4753
261-2	1516	4983
262-2	1476	4869
265	848	4678
266	938	4556
301	1552	5894
302	2010	5311
303	1630	5515
Average	1652	6556
Anatomy	77,286	504,156

thus makes the algorithm computationally costly. For reasons of algorithmic efficiency, we propose ADAM for approximating individuals' fitness. We extract feasible solutions and evaluate them with fitness function in the solution space using a grid design. Based on this, ADAM is constructed for each individual with the approximation of its fitness value using neighborhood. In addition, we analyze the local FL where the new individual locates to determine whether to use ADAM. Experimental results illustrate the

Table 3. Comparison among EA-IM and OAEI's participants in terms of f-measure on Benchmark.

Matcher	F-Measure
AML	0.38
CroMatcher	0.89
Lily	0.89
LogMap	0.55
PhenoMF	0.01
PhenoMM	0.01
PhenoMP	0.01
XMap	0.56
EA-ADAM	0.87

Table 4. Comparison among EA-IM and OAEI's participants in terms of f-measure on Anatomy.

Matcher	F-Measure
ALIN	0.85
LSMatch	0.76
AMD	0.86
SORBETMtch	0.91
Matcha	0.94
OLaLa	0.91
LogMap	0.88
LogMapBio	0.90
LogMapLite	0.83
StringEquiv	0.77
IM-HEA	0.84
EA-ADAM	0.86

ability of ADAM for helping EA to explore the feasible domain efficiently, thus determine high-quality alignments.

For further improving EA-ADAM's performance, we will attempt to adjust SPs' number according to the heterogeneous characteristics of the matching task. Additionally, we will train different similarity measures for specific problems for further distinguishing heterogeneous entities.

Acknowledgment. Research project supported by Shanxi Scholarship Council of China 2023061. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[2] X. Xue and J. Lu, "A compact brain storm algorithm for matching ontologies," *IEEE Access*, vol. 8, pp. 43 898–43 907, 2020.

[3] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities," in *The Semantic Web — ISWC 2002*, I. Horrocks and J. Hendler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 333–347.

[4] V. Tamma, S. Phelps, I. Dickinson, and M. Wooldridge, "Ontologies for supporting negotiation in e-commerce," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 2, pp. 223–236, 2005, agent-oriented Software Development. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197604001836

[5] X. Xue, Q. Wu, M. Ye, and J. Lv, "Efficient ontology meta-matching based on interpolation model assisted evolutionary algorithm," *Mathematics*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252145909

[6] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: A literature review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414005144

[7] X. Xue and J. Chen, "Matching biomedical ontologies through compact differential evolution algorithm with compact adaption schemes on control parameters," *Neurocomputing*, vol. 458, pp. 526–534, 2020.

[8] A. Maedche and S. Staab, "Measuring similarity between ontologies," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, A. Gómez-Pérez and V. R. Benjamins, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 251–263.

[9] J. Martinez-Gil, E. Alba, and J. F. A. Montes, "Optimizing ontology alignments by using genetic algorithms," in *Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web. Karlsruhe, Germany.* Aachen, DEU: CEUR-WS.org, 2008.

[10] X. Xue and Q. Huang, "Generative adversarial learning for optimizing ontology alignment," *Expert Systems*, vol. 40, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:258112729

[11] Q. Lv, 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, 2021.

[12] 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, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2073-8994/15/1/124

[13] T.-Y. Wu, H. Li, and S.-C. Chu, "Cppe: An improved phasmatodea population evolution algorithm with chaotic maps," *Mathematics*, vol. 11, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/9/1977

[14] X. Xue and X. Yao, "Interactive ontology matching based on partial reference alignment," *Applied Soft Computing*, vol. 72, pp. 355–370, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494618304514

[15] J. Martinez-Gil, E. Alba, and J. F. A. Montes, "Optimizing ontology alignments by using genetic algorithms," in *Proceedings of the First International Conference on Nature Inspired Reasoning for the Semantic Web - Volume 419*, ser. NatuReS'08. Aachen, DEU: CEUR-WS.org, 2008, p. 1–15.

[16] X. Lu, K. Tang, and X. Yao, "Classification-assisted differential evolution for computationally expensive problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 1986–1993.

[17] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, "Selecting hyper-parameters of gaussian process regression based on non-inertial particle swarm optimization in internet of things," *IEEE Access*, vol. 7, pp. 59 504–59 513, 2019.

[18] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[19] Y. Sun, S. K. Halgamuge, M. Kirley, and M. A. Munoz, "On the selection of fitness landscape analysis metrics for continuous optimization problems," in *7th International Conference on Information and Automation for Sustainability*, 2014, pp. 1–6.

[20] X. Xue and J. Chen, "A preference-based multi-objective evolutionary algorithm for semiautomatic sensor ontology matching," *International Journal of Swarm Intelligence Research*, vol. 9, no. 2, pp. 1–14, 2018.

[21] G. Alexandru-Lucian and A. Iftene, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," in *9th RoEduNet IEEE International Conference*, 2010, pp. 118–122.

[22] X. Xue, J. Liu, P.-W. Tsai, X. Zhan, and A. Ren, "Optimizing ontology alignment by using compact genetic algorithm," in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, 2015, pp. 231–234.

[23] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[24] T.-Y. Wu, A. Shao, and J.-S. Pan, "Ctoa: Toward a chaotic-based tumbleweed optimization algorithm," *Mathematics*, vol. 11, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/10/2339

[25] H. Zhu, X. Xue, A. Geng, and H. Ren, "Matching sensor ontologies with simulated annealing particle swarm optimization," *Mobile Information Systems*, vol. 2021, pp. 5 510 055:1–5 510 055:11, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233313414

[26] X. Xue, Y. Wang, and W. Hao, "Optimizing ontology alignments by using nsga-ii," *The International Arab Journal of Information Technology*, vol. 12, pp. 175–181, 2015.

[27] X. Xue and Y. Wang, "Using moea/d for optimizing ontology alignments," *Soft Computing*, vol. 18, pp. 1589–1601, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:13370340

[28] U. Marjit and M. Mandal, "Multiobjective particle swarm optimization based ontology alignment," in *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, 2012, pp. 368–373.

[29] Z. Lv and R. Peng, "A novel meta-matching approach for ontology alignment using grasshopper optimization," *Knowledge-Based Systems*, vol. 201-202, p. 106050, 2020.

[30] X. Zhou, Q. Lv, and A. Geng, "Matching heterogeneous ontologies based on multi-strategy adaptive co-firefly algorithm," *Knowledge and Information Systems*, vol. 65, pp. 2619 – 2644, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257324368

[31] I. El Naqa and M. J. Murphy, *What Is Machine Learning?* Cham: Springer International Publishing, 2015, pp. 3–11. [Online]. Available: https://doi.org/10.1007/978-3-319-18305-3_1

[32] X. XUE, Y. HUANG, and Z. ZHANG, "Deep reinforcement learning based ontology meta-matching technique," *IEICE Transactions on Information and Systems*, vol. E106.D, no. 5, pp. 635–643, 2023.

[33] Z. Ding, Y. Huang, H. Yuan, and H. Dong, *Introduction to Reinforcement Learning.* Singapore: Springer Singapore, 2020, pp. 47–123. [Online]. Available: https://doi.org/10.1007/978-981-15-4095-0_2

[34] A. Dean, D. Voss, and D. Draguljić, *Polynomial Regression.* Cham: Springer International Publishing, 2017, pp. 249–284. [Online]. Available:https://doi.org/10.1007/978-3-319-52250-0_8

[35] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[36] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, and C.-M. Chen, "Gaussian mutation-ndash;spider monkey optimization (gm-smo) model for remote sensing scene classification," *Remote Sensing*, vol. 14, no. 24, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/24/6279

[37] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104 555–104 564, 2020.

[38] H. S. Bernardino, H. J. C. Barbosa, and L. G. da Fonseca, "Surrogate-assisted clonal selection algorithms for expensive optimization problems," *Evolutionary Intelligence*, vol. 4, pp. 81–97, 2011. [Online]. https://api.semanticscholar.org/CorpusID:41536839

[39] A. Ciccazzo, G. D. Pillo, and V. Latorre, "Support vector machines for surrogate modeling of electronic circuits," *Neural Computing and Applications*, vol. 24, pp. 69–76, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:7713208

[40] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[41] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[42] L. Gräning, Y. Jin, and B. Sendhoff, "Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study," in *The European Symposium on Artificial Neural Networks*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:1654446

[43] T. Østergård, R. L. Jensen, and S. E. Maagaard, "A comparison of six metamodeling techniques applied to building performance simulations," *Applied Energy*, vol. 211, pp. 89–103, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261917315489

[44] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Transactions on Cybernetics*, vol. 49, pp. 1642–1656, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:51615934

[45] R. E. Smith, B. A. Dike, and S. A. Stegmann, "Fitness inheritance in genetic algorithms," in *ACM Symposium on Applied Computing*, 1995. [Online]. Available: https://api.semanticscholar.org/CorpusID:18329843
[46] Z. Y. Jin, "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems," *Memetic Computing*, vol. 10, no. 2, 2018.
[47] H. S. Bernardino, L. G. da Fonseca, H. J. C. Barbosa, and L. N. de Computação, "Surrogate-assisted artificial immune systems for expensive optimization problems," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:14237374
[48] *ACL '94: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. USA: Association for Computational Linguistics, 1994.
[49] G. A. Miller, "Wordnet: A lexical database for english," *Computer Science*, vol. 38, pp. 39–41, 1995.
[50] V. Mascardi, A. Locoro, and P. Rosso, "Automatic ontology matching via upper ontologies: A systematic evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 609–623, 2009.
[51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399
[52] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1965. [Online]. Available: https://api.semanticscholar.org/CorpusID:60827152
[53] M. A. Jaro, "Probabilistic linkage of large public health data files." *Statistics in Medicine*, vol. 14 5-7, pp. 491–8, 1995. [Online]. Available: https://api.semanticscholar.org/CorpusID:22858045
[54] W. E. Winkler, "The state of record linkage and current research problems," in *Statistical Research Division, US Census Bureau*. Citeseer, 1999.
[55] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addison-Wesley Pub. Co.,*, 1989.
[56] J. M. Gil and J. F. A. Montes, "Evaluation of two heuristic approaches to solve the ontology meta-matching problem," *Knowledge and Information Systems*, vol. 26, pp. 225–247, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:422693
[57] M. Ehrig and J. Euzenat, "Relaxed precision and recall for ontology matching," in *Integrating Ontologies*, 2005.
[58] J. O. Gyapong and J. H. Remme, "The use of grid sampling methodology for rapid assessment of the distribution of bancroftian filariasis," *Transactions of the Royal Society of Tropical Medicine and Hygiene*, vol. 95, no. 6, pp. 681–686, 2001.
[59] L. G. da Fonseca, H. J. C. Barbosa, and A. C. C. Lemonge, "A similarity-based surrogate model for enhanced performance in genetic algorithms," *OPSEARCH*, vol. 46, pp. 89–107, 2009.
[60] S. Wright, "The roles of mutation, inbreeding, crossbreeding, and selection in evolution," *Proceedings of the Sixth International Congress on Genetics*, pp. 356–366, 1932.
[61] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, E. Jiménez-Ruiz, E. Kuss, P. Lambrix, H. Leopold, H. Li, C. Meilicke, S. Montanelli, C. Pesquita, T. Saveta, and O. Šváb Zamazal, "Results of the ontology alignment evaluation initiative 2016," 2016, pp. 73–129.
[62] X. Liu, "A research on population size impaction on the performance of genetic algorithm." Ph.D. dissertation, North China Electric Power University;North China Electric Power University(Baoding), 2010.
[63] S. Mirjalili, *Genetic Algorithm*. Cham: Springer International Publishing, 2019, pp. 43–55. [Online]. Available: https://doi.org/10.1007/978-3-319-93025-1
[64] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.
[65] X. Xue, Q. Wu, M. Ye, H. Zhu, and Y. Huang, "Efficient biomedical ontology meta-matching based on interpolation model based hybrid evolutionary algorithm," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 2490–2497.