Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

Many-Objective Instance Matching in Linked Open Data

Xingsi Xue[1,2], Zhengyi Tang[1,2]

[1]College of Information Science and Engineering
[2]Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
No.3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian, 350118, China
jack8375@gmail.com, tangzy84@126.com

Junfeng Chen

College of IOT Engineering
Hohai University
200 North Jinling Road, Changzhou, Jiangsu, 213022, China
chen-1997@163.com

Jeng-Shyang Pan[1,2,3,*]

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications
[2]College of Information Science and Engineering
Fujian University of Technology, Fujian, China

[3]Department of Information Management
Chaoyang University of Technology
168, Jifeng E. Rd., Wufeng District, Taichung, 41349, Taiwan
*Corresponding author: jspan@cc.kuas.edu.tw

Received November, 2017; revised December, 2017

Abstract. *Instance matching is a key solution to the instance heterogeneity problem in Linked Open Data (LOD), which is able to identify the semantic correspondences between instances. Since different matching algorithms, called matchers, do not necessarily find the same correct correspondences, usually several competing matchers are applied to the same pair of instances in order to increase evidence towards a potential match or mismatch. How to select, combine and tune different instance matchers to obtain the high quality instance alignment is one of the main challenges in LOD domain. Recently, Evolutionary Algorithms (EAs) are appearing as an effective methodology to address this challenge. However, traditional EA based approaches ignore the effects brought about by different instance mapping's preference on different matchers, which could significantly reduce the alignment's quality, and the weights determined in this way could be problem-specific, which might not be reused in other instance matching scenarios. To overcome this drawback, in this paper, we propose a Non-dominated Sorting Genetic Algorithm (NSGA)-III based many-objective instance matching approach, which can tradeoff each mapping's preference on various matchers and determine the optimal instance alignment without tuning different matchers' aggregating weights. The experiment is conducted on the ISLab Instance Matching Benchmark (IIMB) provided by the Ontology Alignment Evaluation Initiative (OAEI), and the comparisons with OAEI's participants show the effectiveness of our proposal.*
Keywords: Many-objective instance matching, Linked open data, NSGA-III

1. **Introduction.** The Linked Open Data (LOD) is a cornerstone in the realization of the Semantic Web. However, since an instance in the LOD is likely to be denoted with many identifiers (e.g., URIs) by different parties, instance heterogeneity problem, which identifies different identifiers for the same instance and eliminates the inconsistency between the datasets, has become critical to the development of LOD [1]. Instance matching is able to identify the semantic correspondences between heterogeneous instances, which is regarded as a key solution to the instance heterogeneity problem in LOD. Since different matching algorithms, called matchers, do not necessarily find the same correct instance correspondences, usually several competing matchers are applied to the same pair of instances in order to increase evidence towards a potential match or mismatch [2]. How to select, combine and tune different instance matchers to determine the high quality instance alignment is one of the main challenges in LOD domain. In particular, among different compositions, the parallel composition of basic matchers, due to its ability of dynamically tuning the basic matchers to obtain the high quality output, becomes the key breakthrough for obtaining first-rate matching performance [3].

Evolutionary Algorithms (EA) is an effective methodology to implement the parallel composition of different matchers [4]. Both single [5, 6, 7] and multi-objective [8, 9, 10, 11] EAs based matching approaches have been proposed, which regard various matchers as the black boxes, and try to determine optimal weights to aggregate their outputs. However, ignoring the effects brought about by different instance mapping's preference on different matchers could significantly reduce the alignment's quality, and weights tuned in this way could be problem specific, which means they might not be reused in other instance matching scenarios. In this paper, we propose a Non-dominated Sorting Genetic Algorithm (NSGA)-III [12] based many-objective instance matching technique, which can tradeoff each instance mapping's preference on various matchers, and determine the optimal instance alignment without tuning the matchers' aggregating weights. In particular, for the first time, we present a problem-specific NSGA-III for the instance heterogeneity problem in LOD, which utilizes the uniform design to generate uniformly distributed reference points and θ-dominance to improve the convergence without sacrificing its diversity.

The rest of the paper is organized as follows: Section 2 describes the related works; Section 3 presents the many-objective instance matching problem and instance matchers; Section 4 presents the NSGA-III based many-objective instance matching technique; Section 5 presents the experimental studies and analysis; finally, Section 6 draws the conclusions and presents the future work.

2. **Matcher Selection, Combination and Tuning.** In general, matcher selection, combination and tuning are tackled by setting appropriate weight set through different methods. The most outstanding approaches in this area are COMA [13], COMA++ [14], QuickMig [15] and OntoBuilder [16], but they use weights determined by an expert. Lately, the focus is on the heuristic techniques for combining different matchers. The first method is called harmonic adaptive weighted sum which is presented in the PRIOR+ [17]. The harmony value is calculated through a similarity matrix and further assigned as the weight to the matcher associated with that matrix. The second method is called local confidence weighted sum, which is the core method for combining individual matchers in the AgreementMaker [18]. This measure is defined for an entity by considering the average of similarity values of entities that are associated with that entity, and the average of similarity values of entities that are not associated with it. Finally, the selection of the final candidates from the set of candidates is performed by a greedy selection strategy. The third approach is dynamic weighted sum method, which is used to combine

terminological and structural matchers in YAM++ [19]. For a given matching scenario, this method evaluates the degree of reliability of these matchers, and assigns appropriate weight values to them. More recently, Benaissa et al. propose a heuristic strategy to estimate the weights for different matchers [20], which is of a statistical nature and estimates the weights by an estimation of the precision standard metric.

Recently, EAs are appearing as an effective methodology to determine the optimal aggregating weights for the matchers. GOAL [21] is the first matching system that utilizes EA to determine the weight configuration for a weighted average aggregation of several matchers by considering a reference alignment. Similar idea of combining multiple matchers is also developed in [22] and [23]. More recently, Xue et al. present an approach based on a Multi-Objective EA (MOEA) to determine the optimal weights being assigned to different matchers used [24]. All these methods dedicate to tune the weights for aggregating different matchers, which ignores the effects brought about by different entity mappings' preferences on different matchers and decreases the quality of alignment. In our work, we propose a many-objective EA based matching technique to take into consideration each mapping's preference on various matchers, and determine the optimal alignment without tuning the aggregating weights.

3. **Preliminaries.**

3.1. **Many Objective Instance Matching.** Given an instance candidate set $S_o = o_1, o_2, \ldots, o_n$, the instance matching on set S_o can be defined as looking for the bigest sub-set of set S_o such that the similarity between o and each element of the sub-set is larger than a threshold. Based on the observations that the more correspondences found and the higher mean similarity values of the correspondences are, the better the alignment quality is [25], we utilize the following metric to measure the quality of an instance alignment:

$$f(A) = \frac{\phi(A) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|}}{\alpha \times \phi(A) + (1-\alpha) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|}} \tag{1}$$

where $|A|$ is the number of correspondences in A, ϕ is a function of normalization in [0,1], δ_i is the similarity value of the ith correspondence in A, and α is a parameter used to tradeoff the instance alignments characterized by high recall (with the decreasing of α) or high precision (with the increase of α). In general, the value of α is set to 0.5 to prefer neither recall nor precision.

On this basis, the many-objective optimal model of instance matching can be defined as follows:

$$\begin{cases} min \quad F(A) = (1 - f_1(A), 1 - f_2(A), \cdots, 1 - f_m(A)) \\ s.t. \quad A = (a_1, a_2, \cdots, a_{|I_{src}|})^T \\ \quad a_i \in \{1, 2, \cdots, |O_{tgt}|\}, i = 1, 2, \cdots, |I_{src}| \end{cases} \tag{2}$$

where m is the number of matchers, $f_i(A), i = 1, 2, \cdots, m$ calculates the alignment A's quality with respect to the ith matcher, $|I_{src}|$ and $|I_{tgt}|$ respectively represent the cardinalities of source instance set I_{src} and target instance set I_{tgt}, and $x_i, i = 1, 2, \cdots, |I_{src}|$ represents the ith pair of correspondence.

3.2. **Instance Matcher.** Instance matcher takes as input two instance sets I_{src} and I_{tgt} and output an $|I_{src}| \times |I_{tgt}|$ similarity matrix S, whose element s_{ij} is the similarity score between ith instance in $|I_{src}|$ and jth instance in $|I_{tgt}|$. In general, the basic matchers

can be divided into four categories, i.e. syntactic-based matcher, linguistic-based matcher and structure-based matcher [1].

3.2.1. *Syntactic-based Matcher.* Syntactic-based matcher calculates the edit distance between two instances, and in this work, we choose the SMOA, which is the most performing syntactic similarity measure in the ontology matching domain [26]. Formally, given two words w_1 and w_2, the SMOA value between them can be defined by the following equation:

$$SMOA(w_1, w_2) = c(w_1, w_2) - d(w_1, w_2) + winklerImprove(w_1, w_2) \tag{3}$$

where $c(w_1, w_2)$ stands for the commonality between w_1 and w_2, $d(w_1, w_2)$ for the difference and $winklerImprove(w_1, w_2)$ for the improvement of the result proposed in [27].

3.2.2. *Linguistic-based Matcher.* Linguistic-based matcher utilizes synonymy, hypernymy and other linguistic relations to calculate the similarity score between instances. To this end, a lexicon and thesauri are needed, and the most popular one is WordNet [28] which is an electronic lexical database where various senses of words are put together into sets of synonyms. Given two words w_1 and w_2,

$$LinguisticDistance(w_1, w_2) = \begin{cases} 1, & \text{if } w_1 \text{ and } w_2 \text{ are synonymous,} \\ 0.5, & \text{if } w_1 \text{ is the hypernym of } w_2 \text{ or vice versa,} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

3.2.3. *Structure-based Matcher.* Structure-based matcher computes a similarity score between two instances based on their neighbors in the LOD, and the common intuition is that two distinct instances are similar when their neighbor instances are similar. In this work, we first construct a neighbor profile for each instance, and finally, the similarity between two instances I_1 and I_2 is calculated as follows [1]:

$$InstanceSim(I_1, I_2) = \frac{\sum_{i=1}^{h} \max_{j=1\cdots k}(sim(I_{1i}, I_{2j})) + \sum_{j=1}^{k} \max_{i=1\cdots h}(sim(I_{1i}, I_{2j}))}{h + k} \tag{5}$$

where:

- h is the number of neighbors of I_1 and k is the number of neighbors of I_2,
- I_{1i} is the ith neighbor of I_1 and I_{2j} is the jth neighbor of I_2,
- $sim()$ calculates the SMOA distance between the labels of I_{1i} and I_{2j}.

In particular, we set the maximum distance between an instance and its neighbor as 2 to reduce the time of building neighbor instance profile.

4. **NSGA-III based Many-objective Instance Matching.** In this work, we propose a NSGA-III based many-objective instance matching technique to automatically select and combine various matchers and determine the optimal instance alignment. In particular, NSGA-III is a many-objective algorithm proposed by Deb et al. [12], which introduces a well distributed reference points based clustering operator to replace the crowding distance operator in NSGA-II [29]. Particularly, original NSGA-III emphasizes that the solutions should be pareto non-dominated and closed to the reference line of each reference point. However, with the growing number of the objectives, selection pressure based on pareto dominance would be too small to pull the population towards pareto front, and in this case, NSGA-III indeed emphasizes diversity more than convergence. To this end, we present a problem-specific NSGA-III to improve the convergence as well as maintain the diversity.

In the next, three key components of NSGA-III are presented in details, i.e. encoding mechanism, uniform design based reference points generation and θ-dominance. Finally, we give the outline of NSGA-III.

4.0.1. *Encoding Mechanism.* Let $|I_{src}|$ and $|I_{tgt}|$ be the number of instance in the source instance set I_{src} and target instance set I_{tgt}, respectively. Each chromosome in the population would be a one-dimensional array with $|I_{src}|$ integer elements, and the elements are denoted as: $N_1N_2\cdots N_{|I_{src}|}$, where $N_i \in \{0, 1, \cdots, |I_{tgt}|\}$, $i \in \{1, \cdots, |I_{src}|\}$, which means the ith instance in I_{src} is mapped to the N_ith instance in I_{tgt}. In particular, when $N_i = 0$, the ith instance is not mapped to any instance in I_{tgt}.

4.0.2. *Uniform Design based Reference Points Generation.* In the original NSGA-III, the Das and Dennis's systematic approach [30] is used to generate reference points. However, when the number of objectives is high, the number of reference points generated by this approach would become very large [31]. In our work, we propose to use a uniform design [32], which aims at determining a set of points that are uniformly distributed over the design space, to produce uniformly distributed reference points in a unit sphere $S = \{(s_1, s_2, \cdots, s_m) | \sum_{i=1}^{m} s_i^2 = 1, s_i \geq 0, i = 1, 2, \cdots, m\}$. Firstly, we need to generate a set of Q uniformly distributed points on $C = \{(c_1, c_2, \cdots, c_m) | 0 \leq c_1, c_2, \cdots, c_m \leq 1\}$. Let Q be the number of uniform distributed points in C, and m be the dimension of the problem that is equal to the number of basic matchers in this work, δ be the number that yields the smallest discrepancy of generated point set (see also [33]), an integer matrix so called uniform array $[M_{ij}]_{Q\times m}$ can be calculated with $M_{ij} = i\delta^{j-1} \bmod Q+1, i = 1, 2, \cdots, Q, j = 1, 2, \cdots, m$, where ith row of it can define a point $C_i = (c_{i,1}, c_{i,2}, \cdots, c_{i,m})$ with $c_{ij} = \frac{2M_{ij}-1}{2Q}, i = 1, 2, \cdots, Q, j = 1, 2, \cdots, m$. Next, a set of Q reference points uniformly distributed on S, denote by $P(Q, m) = P_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,m})$, can be calculated as follows:

$$p_{i,j} = \begin{cases} \prod_{s=1}^{m-1} \cos(0.5c_{i,s}\pi) & j = 1 \\ \sin(0.5c_{i,m-j+1}\pi) \prod_{s=1}^{m-j} \cos(0.5c_{i,s}\pi) & 1 < j < m \\ \sin(0.5c_{i,1}\pi) & j = m \end{cases} \tag{6}$$

Finally, we also added m apexes of the normalized hyper-plane $C_{norm} = \{(c_1, c_2, \cdots, c_m) | 0 \leq c_1, c_2, \cdots, c_m \leq 1, \sum_{i=1}^{m} c_i = 1\}$ as the reference points.

4.0.3. *θ-dominance.* Given reference points $P(Q, m) = \{P_i, P_2, \cdots, P_Q\}$, a reference line is defined by joining a reference point with the origin. After that, each individual is associated with a reference point by calculating the perpendicular distance of it from each of the reference line. The reference point whose reference line is closest to a solution is considered to be associated with this solution. In this way, the population can be split into Q clusters $C = \{C_1, C_2, \cdots, c_Q\}$ where cluster C_j is represented by the reference point P_j, $j = 1, 2, \cdots, Q$.

Given a solution x and its objective vector $f(x) = (f_1(x), f_2(x), \cdots, f_m(x))$, reference line L_j passing through the origin and P_i, a penalty function [34] can be defined as $D_j(x) = \|f(x)\| + \theta d_{j,perpendicular}(x), j = 1, 2, \cdots, Q$, where $d_{j,perpendicular}(x)$ calculates the perpendicular distance between $f(x)$ and L_j:

$$d_{j,perpendicular}(x) = \|f(x) - \frac{\|f(x)^T P_j\|}{\|P_j\|}(\frac{P_j}{\|P_j\|})\| \tag{7}$$

In this work, $\theta > 0$ is a predefined penalty parameter, which is set as 2 to achieve best mean quality of alignment on all test cases. It's obvious that the smaller $\|f(x)\|$ and $d_{j,perpendicular}(x)$ respectively leads to better convergence and better diversity. Given

two solutions $x, y \in \Omega$, x is said to θ-dominate y, denoted by $x \prec_\theta y$, if $x, y \in C_j$ and $D_j(x) < D_j(y)$, $j \in \{1, 2, \cdots, Q\}$ [31]. In this work, we utilize the θ-dominance to implement the fast non-dominated sorting [29] on the population to partition it into different θ-non-domination levels.

4.0.4. *The Flowchart of NSGA-III.* The flowchart of NSGA-III is presented in figure 2. First, we apply a uniform design based method that could generate any number of reference points. In this work, we use the common one point crossover operator, and the bit mutation operator, i.e. for each bit in the chromosome we check if the mutation could be applied according to the mutation probability and if it is, the value of that bit is then flipped. After offspring population is generated, before calculating the perpendicular distance between a population and each of the reference lines, NSGA-III need to normalize objective lvalues and supplied reference points to ensure they have an identical range and the ideal point is the zero vector. In this work, since all the objective's values are in the same range [0,1] and the ideal point is the zero vector, we don't need to carry out the normalization in each generation. In addition, replace the Pareto dominance in NSGA-III with θ-dominance to tradeoff the convergence and diversity in many-objective optimization, and utilize the θ-dominance based fast non-dominated sorting is employed on the population clusters to divide them into different θ-non-domination levels. Finally, we determine the next generation's population by including one θ-non-domination at a time, starting from the first level. With respect to the solutions in last accepted level, we first sort them in ascending order according to their mean $f()$ values [4], and then select the solutions sequentially. In this work, in order to compare with other instance matching systems whose results are measured with f-measure [35], we pick up the solution in the Pareto front with the highest $\frac{\sum_{i=1}^{m} f_i}{m}$ as the representative solution.

5. **Experimental Studies and Analysis.** In order to study the effectiveness of our proposal, we exploit the ISLab Instance Matching Benchmark (IIMB) provided by the Ontology Alignment Evaluation Initiative (OAEI). IIMB is a collection of OWL ontologies consisting of 29 concepts, 20 object properties, 12 data properties and thousands of individuals divided into 80 test cases. This benchmark defines 80 test cases, divided into 4 sets of 20 test cases each. The first three sets are different implementations of data value, data structure and data semantic transformations, respectively, while the fourth set is obtained by combining together the three kinds of transformations.

The parameters used by NSGA-III are as follows: numerical accuracy=0.01, number of reference points=20, population size=25, crossover probability=0.8, mutation probability=0.02 and maximum number of generation=300. These parameters represent a tradeoff setting obtained in an empirical way to achieve the highest average alignment quality on all test cases of exploited dataset, which is robust against the heterogeneous situations in our experiment.

Table 5 shows the mean value of f-measure on reference alignment of the alignments obtained by our approach in thirty independent runs and the results obtained by the participants of OAEI. As can be seen from Table 5, although all four systems have good values of f-measure on datasets with data value transformation, RiMOM and our approach have the best values for f-measure. Moreover, our approach has the best operation in test cases with structural value transformation which means that our approach is more stable in the modifications such as removing, adding and hierarchal changing of properties. Our approach also has the best results when confronting with semantic value transformation. Although all the instance matchers do not have desirable results which combine all kinds

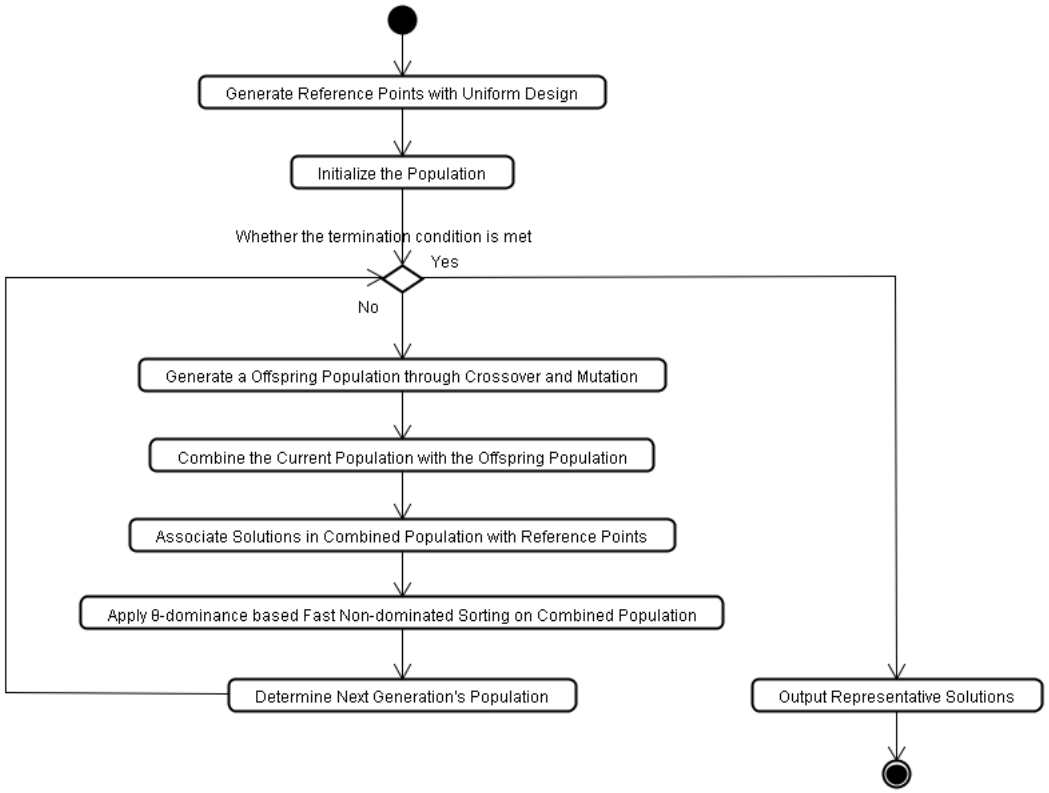

FIGURE 1. The Flowchart of NSGA-III

TABLE 1. Comparison between OAEI 2016's participants and our approach on instance matching track in terms of f-measure.

benchmark ID	ASMOV	CODI	RiMOM	Our approach
001-020	0.97	0.96	**0.98**	**0.98**
021-040	0.86	0.88	0.85	**0.91**
041-060	0.89	0.96	0.93	**0.97**
061-080	0.46	0.55	0.55	**0.63**

of transformations, our approach still has the best result. Therefore, our approach is able to effectively combine different matchers to determine the high quality instance alignment.

6. **Conclusion and Future Work.** One of the main challenges in instance matching domain is how to select, combine and tune different matchers to obtain the high quality instance alignment. Since among the existing instance matching technologies, those based on EA are appearing as the most suitable methodology to solve the instance matching problem, in this paper, we present a many-objective instance matching technology based on NSGA-III, which can optimize each matcher's alignment simultaneously, and efficiently combine various matchers without tuning the aggregating weights. The experimental results show that our proposal is effective to determine the high quality instance alignment in LOD.

In continuation of our research, we are trying to select and combine more matchers, e.g. more than 20 different matchers. Moreover, some strategies which could remove the mappings that lead to logical conflicts can be introduced to further improve the

alignment's quality. Last but not least, we are also interested in getting the user involved in our approach to guide the search direction, so that the alignment quality could be further improved.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Nos. 61503082 and 61403121), Natural Science Foundation of Fujian Province (Nos. 2016J05145 and 2016J05146), Fundamental Research Funds for the Central Universities (No. 2015B20214), Scientific Research Startup Foundation of Fujian University of Technology (No. GY-Z15007), Fujian Province Outstanding Young Scientific Researcher Training Project (No. GY-Z160149) and Scientific Research Development Foundation of Fujian University of Technology (No. GY-Z15087).

REFERENCES

[1] X. Xue, Y. Wang, Using Memetic Algorithm for Instance Coreference Resolution, *IEEE Transactions on Knowledge and Data Engineering*, vol.28, no.3, pp.580–591, 2016.

[2] T. T. A. Nguyen, S. Conrad, Ontology Matching using multiple similarity measures, *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol.1, pp.603–611, 2015.

[3] M. Gulic, B. Vrdoljak, M. Banek, CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.41, pp.50–71, 2016.

[4] X. Xue, Y. Wang, Optimizing Ontology Alignments through a Memetic Algorithm Using both MatchFmeasure and Unanimous Improvement Ratio, *Artificial Intelligence*, vol.223, pp.65–81, 2015.

[5] X. Xue, Y. Wang, A. Ren, Optimizing ontology alignment through memetic algorithm based on partial reference alignment, *Expert Systems with Applications*, vol.41, no.7, pp.3213-3222, 2014.

[6] X. Xingsi, R. Aihong, An Evolutionary Algorithm based Ontology Alignment Extracting Technology, *Journal of Network Intelligence*, vol.2, no.2, pp.205–212, 2017.

[7] G. Acampora, V. Loia, A. Vitiello, Enhancing ontology alignment through a memetic aggregation of similarity measures, *Information Sciences*, vol.250, pp.1-20, 2013.

[8] X. Xue and Y. Wang and W. Hao, Optimizing Ontology Alignments by using NSGA-II, *International Arab Journal of Information Technology*, vol.12, no.2, pp.176-182, 2015.

[9] X. Xue, Y. Wang, W. Hao, J. Hou, Optimizing Ontology Alignments through NSGA-II without Using Reference Alignment, *Computing and Informatics*, vol.33, no.4, pp.857-876, 2014.

[10] X. Xue, Y. Wang, W. Hao, Using MOEA/D for optimizing ontology alignments, Soft Computing, vol.8, no.18, pp.1589–1601, 2014.

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

[12] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, vol.18, no.4, pp.577–601, 2014.

[13] H. H. Do, E. Rahm, COMA: a system for flexible combination of schema matching approaches, *Proceedings of the 28th international conference on Very Large Data Bases*, pp.610–621, 2002.

[14] D. Aumueller, H. H. Do, S. Massmann, E. Rahm, Schema and ontology matching with COMA++, *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp.906–908, 2005.

[15] C. Drumm, M. Schmitt, H. H. Do, E. Rahm, Quickmig: automatic schema matching for data migration projects, *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp.107–116, 2007.

[16] A. Gal, A. Anaby-Tavor, A. Trombetta, D. Montesi, Danilo, A framework for modeling and evaluating automatic semantic reconciliation, *The VLDB JournalThe International Journal on Very Large Data Bases*, vol.14 no.1, pp.50–67, 2005.

[17] M. Mao, Y. Peng, M. Spring, An adaptive ontology mapping approach with neural network based constraint satisfaction, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.8, no.1, pp.14–25, 2010.

[18] F. I. Cruz, F. P. Antonelli, C. Stroe, Efficient selection of mappings and automatic quality-driven combination of matching methods, *Proceedings of the 4th International Conference*, pp.49–60, 2009.
[19] D. Ngo, Z. Bellahsene, Overview of YAM++(not) Yet Another Matcher for ontology alignment task, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.41, pp.30–49, 2016.
[20] M. Benaissa, A. Khiat, A New Approach for Combining the Similarity Values in Ontology Alignment, *IFIP International Conference on Computer Science and its Applications*, pp.343–354, 2015.
[21] J. Martinez-Gil, E. Alba, J. F. A. Montes, Optimizing ontology alignments by using genetic algorithms, *Proceedings of the First International Conference on Nature Inspired Reasoning for the Semantic Web*, pp.1–15, 2008.
[22] J. Naya, M. M. Romero, J. P. Loureiro, C. R. Munteanu, A. P. Sierra, Improving ontology alignment through genetic algorithms, *Soft computing methods for practical environment solutions: Techniques and studies*, pp.240–259, 2010.
[23] G. Alexandru-Lucian, A. Iftene, Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment, *9th Roedunet International Conference*, pp.118–122, 2010.
[24] X. Xue, J. Liu, Optimizing Ontology Alignment Through Compact MOEA/D, *International Journal of Pattern Recognition and Artificial Intelligence*, pp.1–19, 2017.
[25] J. Bock, J. Hettenhausen, Discrete particle swarm optimisation for ontology alignment, *Information Sciences*, vol.192, pp.152–173, 2012.
[26] G. Stoilos, G. Stamou, S. Kollias, A string metric for ontology alignment, *International Semantic Web Conference*, pp.624–637, 2005.
[27] W. E. Winkler, The state of record linkage and current research problems, *Statistical Research Division, US Census Bureau*, 1999.
[28] G. A. Miller, WordNet: a lexical database for English, *Communications of the ACM*, vol.38, no.11, pp.39–41, 1995.
[29] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE transactions on evolutionary computation*, vol.6, no.2, 182–197, 2002.
[30] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM Journal on Optimization*, vol.8, no.3, pp.631–657, 1998.
[31] Y. Yuan and H. Xu, B. Wang, An improved NSGA-III procedure for evolutionary many-objective optimization, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp.661–668, 2014.
[32] K. T. Fang, Y. Wang, Number-theoretic methods in statistics, vol.51, 1993.
[33] C. Dai, Y. Wang, A new uniform evolutionary algorithm based on decomposition and CDAS for many-objective optimization, *Knowledge-Based Systems*, vol.85, pp.131–142, 2015.
[34] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on evolutionary computation*, vol.11, no.6, pp.712–731, 2007.
[35] C. J. Van Rijsberge, Information Retrieval, Butterworth, London, 1975.