

Using Artificial Bee Colony Algorithm for Optimizing Ontology Alignment

Yao He, Xingsi Xue* and Shunmiao Zhang

College of Information Science and Engineering
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
No.3 Xueyuan Road; University Town, Minhou, Fuzhou City, 350118, China
*Corresponding Author:jack8375@gmail.com

Received July, 2016; revised January, 2017

ABSTRACT. *Due to the high heterogeneity of ontologies, a combination of many methods is necessary to correctly discover the semantic correspondences between their elements. But how to determine the optimal combination way in order to obtain an ontology alignment with high quality is still a challenge. To this end, in this paper, we propose to use the Artificial Bee Colony Algorithm (ABC) to solve the ontology meta-matching problem. In particular, we construct a new optimal model for the ontology meta-matching problem and design a problem specific ABC to solve it. The benchmark of OAEI 2014 is utilized to test our approach, and the experimental results show that the mean recall and precision of our approach is generally high, and the mean f-measure of the alignments obtained by our approach outperform all other state of the art matching systems.*

Keywords: Ontology meta-matching, Artificial Bee Colony Algorithm, OAEI 2014

1. **Introduction.** In the semantic web, the kernel technology ontology might be designed by various communities, which may result in the ontology heterogeneous problem and hamper the applications on these ontologies from cooperating with each other. To overcome this problem, the most ground approach is to determine the semantically identical entities between heterogeneous ontologies, so called ontology matching. Since manually match the ontologies is highly impractical, a lot of techniques for ontology matching have arisen in recent years. All of the techniques can, in a fully automatic way or in a semi-automatic way, provide a numerical similarity value to determine whether each pair of entities which belonging to different ontology are semantically similar or not. Since no independent similarity measure could provide satisfied result in all scenarios, the common strategy is to take multiple similarity measures into consideration and aggregate them into a single metric through the aggregating parameter set. Therefore, how to select the appropriate similar measures and how to assign the different weight to obtain a satisfactory matching result, so call ontology meta-matching problem, is critical to the success of Semantic Web.

To solve the ontology meta-matching problem, various ontology matching systems have arisen, and since the beginning, heuristic approaches, and in particular, Evolutionary Algorithms (EAs), are appearing as the most suitable methodology. The most notable one is GOAL (Genetics for Ontology ALignments) [1], which does not directly compute the alignment between two ontologies, but it determines, through a Genetic Algorithm (GA), the optimal weight configuration for a weighted average aggregation of several similarity

measures by considering a reference alignment. Also, Naya etc. [2] propose a genetic algorithm based approach to find out how to aggregate different similarity metrics into a single measure, and Ginsca etc. [3] focus on optimizing the whole similarity aggregation step as a single unit, including the threshold for filtering the final alignment. However, the slow converging speed and the premature convergence are two main shortcomings of the classical EAs for ontology meta-matching problems [4], which makes them incapable of effectively searching the optimal solution for large scale and complex problems. To this end, our work investigates a new emerged swarm intelligence algorithm, named Artificial Bee Colony Algorithm (ABC) [5], to efficiently face the problem of ontology meta-matching. In particular, ABC simulates the foraging behavior of honey bees, and comparing with EA, ABC has less control parameters and stronger global search ability. On the basis of current development of ABC [6, 7, 8, 9, 10], in this work, we further design a Directed Artificial Bee Colony algorithm(DABC) [11] to solve the ontology meta-matching problem, which can effectively improve the convergence speed and overcome the slow convergence problem of EA.

The rest of the paper is organized as follows: Section 2 is devoted to the state of the art related to the ontology alignment; Section 3 is give basic concepts and definitions of the ontology alignment; Section 4 describe the ABC; section 5 propose use ABC to solve similarity aggregation in ontology alignment; section 6 present some experiments carried out ; section 7 draws the conclusions.

2. Directed Artificial Bee Colony Algorithm for Ontology Meta-Matching Problem.

2.1. The Single-Objective Optimal Model for Ontology Meta-matching Problem. In this paper, an ontology can be defined as a quintuple $O = (C, P, I)$ [12], where C is the set of classes, i.e. the set of concepts that populate the domain of interest; P is the set of properties, i.e. the set of relations existing between the concepts of domain; I is the set of instances, i.e. the set of objects of the real world representing the instances of a concept. Furthermore, according to [13], the ontology matching process can be defined as a function θ which, from a pair of ontologies O and O' to align, an input alignment A_I , a set of parameters p , a set of resources r , returns a new alignment A_N between these ontologies: $A_N = \theta(O, O', A_I, p, r)$. The output alignment A_N is a set of semantic matchings, each one of them is used for linking an entity belonging to the first ontology with a similar entity belonging to the second ontology. In the matching process, the quality of an alignment is evaluated through recall, precision and f-measure in information retrieval domain [15], which are defined as follows: given an alignment A and a reference alignment R ,

$$recall = \frac{|R \cap A|}{|C|} \quad (1)$$

$$precision = \frac{|R \cap A|}{|A|} \quad (2)$$

$$f - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3)$$

In our work, we take maximizing the value of f-measure as the goal, and given n similarity measures to aggregate, the single-objective optimal model for ontology meta-matching problem can be defined as follows:

$$\begin{cases} \max & (f - \text{measure}(X)) \\ \text{s.t.} & X = (x_1, x_2, \dots, x_n, x_{n+1})^T \\ & \sum_{i=1}^n x_i = 1 \\ & x_i \in [0, 1], i = 1 \dots n + 1 \end{cases} \quad (4)$$

where the decision variable X represents the parameter set, i.e. the weights for aggregating various similarity measures ($x_i, i = 1 \dots n$), and a threshold (x_{n+1}) for filtering the aggregated alignment, used to obtain the final alignment.

2.2. Artificial Bee Colony Algorithm. Swarm Intelligence, as a subfield of artificial intelligence, is used to solve optimization problems, have attracting more and more attention during the past decade. Artificial Bee Colony(ABC) is a popular swarm intelligence algorithm for function optimization that is inspired by the intuitive food foraging behavior of honey bee insects. In ABC, for a search problem in a D-dimensional space, the position of food source represents a potential solution, and the fitness of solution determined by the quality of the food source. The colony of artificial bees consists of three groups of bees: employed bees, onlooker bees and scout bees. Each of them, plays different role in the process, we describe them as follow:

- The employed bees in charge of search food source and gather to share information about the quality of the food source;
- The onlooker bees stay in the hive to get information about the food source and select one of the source to exploit the nectar;
- The employed bee whose food source has been exhausted by the bees become a scout bee, the scout bees search new food source randomly in places of the abandoned foods sources ;

Similar to the other population-based algorithms, ABC solution search process is an iterative process. After the initialization phase, ABC algorithm repeatedly executes the employed bees phase, onlooker bees phase and scout bees phase until meeting an optimal solution. All these phases will be presented in details in the next sections.

2.2.1. Initialization of the population. This phase is to initialize some parameter and population.

There are three Control parameters to be initialized, i.e. the number of food source, marked with symbol SN ; the number that a position of a solution not updated over which considered to be abandoned, marked with symbol $limit$; the maximum cycle number, marked with symbol MCN . At first, the bee colony consists of employed bees and onlooker bees, the number of employed bees is equal to the onlooker bees, so the percentage of onlooker bees and employed bees is 50% of the whole colony respectively.

The initialization of population is to initialize SN food source. Let a D-dimensional vector $x_i(i = 1, 2, \dots, SN)$ be the i th food source (solution) in the population, where SN is the population size and D is the problem dimension size. Each food source is generated obedient to uniformly distribution as follows:

$$x_{ij} = x_{minj} + \text{rand}[0, 1](x_{maxj} - x_{minj}), \quad \text{for } j \in \{1, 2, \dots, D\} \quad (5)$$

where x_{minj} and x_{maxj} are the lower and upper bound of x_i in j^{th} direction, respectively. $\text{rand}[0,1]$ is a a random number uniformly distributed between 0 and 1.

2.2.2. *Employed bees phase.* In employed bees phase, each employed bee modifies a current solution into a new solution according to Eq.(6). If the fitness value of new solution is higher than the parent solution, the employed bee chooses the new solution instead of the old one to update its position. The new solution generated based on the following formula:

$$x'_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}) \quad (6)$$

where x_k is a randomly selected solution in the current population, $k \neq i, k \in \{1, 2, \dots, SN\}$ and $j \in (1, 2, \dots, D)$ are randomly chosen indices. ϕ_{ij} is a uniformly distributed random number in the range[-1,1].

2.2.3. *Onlooker bees phase.* When all employed bees complete their search, they come into the hive and share the nectar information of the sources with onlooker bees which waiting on the dance area. An onlooker bee chooses a food source according to a probability, p_i associated with the fitness of the food source, calculated by the following expression:

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \quad (7)$$

where f_i is the fitness value of the i th solution in the population. Like the employed bee did, the onlooker bee produces a modification on the position according Eq.(6), and compares the fitness of the candidate solution with the parent solution, if the candidate solution's fitness is higher than the parent solution's fitness, then the bee will choose the candidate one over the parent one.

2.2.4. *Scout bees phase.* If the position of a solution cannot be improved further over a predefined number of cycles, limit, then that food source will be considered to be abandoned and replaced by a new food source, which randomly searched by a scout bee within the search space. This operation is defined as follows:

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}), \quad for \ j \in \{1, 2, \dots, D\} \quad (8)$$

Where x_{minj} and x_{maxj} are lower and upper bound of x_i in j^{th} direction, respectively. $rand[0,1]$ is a a random number uniformly distributed between 0 and 1.

2.2.5. *Main steps of the ABC algorithm.* Base on the description above, the pseudo-code of ABC as show as follows:

Algorithm 1 Artificial Bee Colony Algorithm:

initialize population:

define the control parameters (SN , $limit$ and MCN) ;
randomly generate SN solutions;

repeat

employed bees phase:

modify the current solutions;
choose the solutions with higher fitness;

onlooker bees phase:

select some solutions according to a probability ;
update the solutions depend on their fitness;

scout bees phase:

send the scout bees to randomly search new solutions ;
use the new solutions instead of the abandoned solutions;

memorize the best food source found so far

until termination criteria is satisfied;

2.2.6. *Directed Artificial Bee Colony Algorithm.* ABC use only one design parameter to generate the candidate solution at each iteration, this causes the convergence rate of the algorithm is poorer. In order to accelerate convergence of the algorithm, [16] introduces a control parameter (modification rate-MR) in ABC to determined not one but multiple parameters to be modified, however this approach is based on updating more design parameters than one. Considering ϕ_{ij} is a random number in the range[-1,1] in Eq.(6), this cause fully random direction in the searching around the food source in the standard ABC and cause the slow convergence of the algorithm, Directed Artificial Bee Colony Algorithm(DABC)[11] add directional information for each dimensions of each food source position, and use this directional information to modify Eq.(6) as follows:

$$x'_{ij} = \begin{cases} x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj}) & \text{if } d_{ij} = 0 \\ x_{ij} + r_{ij} \times \text{abs}(x_{ij} - x_{kj}) & \text{if } d_{ij} = 1 \\ x_{ij} - r_{ij} \times \text{abs}(x_{ij} - x_{kj}) & \text{if } d_{ij} = -1 \end{cases} \quad (9)$$

while ϕ_{ij} is a random number in range of [-1,1], r_{ij} is a random number in range of [0,1], and abs is absolute function, d_{ij} is the direction information for j^{th} dimension of the i^{th} food source position.

The directional information for all dimensions are initialized to zero during the initialization phase of the algorithm and changed during the employed bees phase and the onlooker bees phase. If the fitness value of a new solution which obtained by Eq.(9) is higher than the old one, the direction of the dimension is set to 1 or -1. If previous value of dimension is less than the current value, the direction of the dimension is set to 1, otherwise is set to -1. If the fitness value of a new solution is less than old one, the direction of the dimension is set to 0. The formulas is as follows:

$$d_{ij} = \begin{cases} 1 & \text{if } \text{fitness}_{i'} > \text{fitness}_i \text{ and } x'_{ij} > x_{ij} \\ -1 & \text{if } \text{fitness}_{i'} > \text{fitness}_i \text{ and } x'_{ij} < x_{ij} \\ 0 & \text{if } \text{fitness}_{i'} < \text{fitness}_i \end{cases} \quad (10)$$

where $\text{fitness}_{i'}$ is the i^{th} new solution and fitness_i is the i^{th} old solution, x'_{ij} is the value of the i^{th} new solution in j^{th} dimension and x_{ij} is the value of the i^{th} old solution in j^{th} dimension.

3. Experimental Result and Analysis.

3.1. **Experiment Configuration.** The similarity measures used are as follows:

- Levenstein Distance (Syntactic Measure) [17],
- Jaro Diatance (Syntactic Measure) [13],
- WordNet based similarity measure (Linguistic Measure) [18],
- Similarity Flooding based similarity measure (Taxonomy-based Measure) [19].

The parameters used by DABC algorithm are listed as follows:

- MCN(maximum cycle number): 300,
- NP(the Size of colony): 20,
- SN(the number of food source): 10(NP/2),
- the number of employee bees: 10(NP/2),
- the number of onlooker bees: 10(NP/2),
- D(the number of dimension of food source): 4,
- limit: 5,
- upper bound: 1,
- lower bound: 0,

TABLE 1. Brief Description of OAEI 2014 Benchmark

ID	Brief description
101-104	The ontologies under alignment are the same with each other
201-210	The ontologies under alignment have the same structure, but different lexical and linguistic features
221-247	The ontologies under alignment have the same lexical and linguistic features, but different structure
248-266	The ontologies under alignment have different lexical, linguistic and structure features
301-304	The ontologies under alignment are real world cases

- direction: a $SN \times D$ matrix, The initial value of all elements is 0,
- terminate condition: the number of cycles reach MCN, or f-measure reach 1, or Global optimal solution is not updated up to 30.

The hardware configurations used to run the algorithms are as follows:

- Processor: Intel Core i7-4600U CPU,
- CPU speed: 2.10GHz,
- RAM capacity: 15G.

3.2. Results and analysis. In order to study the effectiveness of our proposal, we have exploited a well-known dataset, named benchmark track, provided by the Ontology Alignment Evaluation Initiative (OAEI) 2014 [14] and commonly used for experimentation about ontology alignment problem. In detail, each test case, see Table I, consists of a set of small scale ontologies which are built around a seed ontology that contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals, and many variations of it. Variations are artificially generated, and focus on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups: Simple tests (1xx) compares the reference ontology with itself; Systematic tests (2xx) are obtained by discarding/modifying features, which include names of entities, comments, the specialization hierarchy, instances, properties and classes, from the reference ontology; Real-life ontologies (3xx) are found on the web. In this experiment, we utilize the downloadable datasets from the OAEI 2014 official website for testing purposes.

In order to compare the quality of our proposal with the participants of OAEI 2014 and GOAL, the famous ontology matching system based on GA which is introduced in the Section 1, we evaluate the obtained alignments with traditional recall, precision and f-measure, and the results in Table II are the mean values of all the test cases, and the symbols R , P and F stand for recall, precision and f-measure, respectively.

As can be seen from Table II that the mean recall and precision of our approach are generally high, and the mean f-measure of the alignments obtained by our approach outperforms all other matching systems. Therefore, through the comparison with the state-of-the-art ontology matching systems, our proposal is effective.

4. Conclusion. Due to the high heterogeneity of ontologies, a combination of many methods is necessary to correctly discover the semantic correspondences between their elements. But how to determine the optimal combination way in order to obtain an ontology alignment with high quality is still a challenge. To this end, in this paper, we propose to use ABC to solve the ontology meta-matching problem. In particular, we construct a new optimal model for the ontology meta-matching problem and design a

TABLE 2. Comparison of our system with the participants in OAEI 2014

Systems	R	P	F
AML	0.39	0.92	0.55
AOT	0.53	0.80	0.64
AOTL	0.53	0.85	0.65
LogMap	0.40	0.40	0.40
LogMap-C	0.40	0.42	0.41
LogMapLite	0.50	0.43	0.46
MaasMatch	0.39	0.97	0.56
OMReasoner	0.50	0.73	0.59
RSDLWB	0.50	0.99	0.66
XMap2	0.40	1.00	0.57
GOAL	0.76	0.68	0.72
Our System	0.79	0.92	0.85

problem specific ABC to solve it. We utilize the benchmark of OAEI 2014 to test our approach, and the experimental results show that the mean recall and precision of our approach is generally high, and the mean f-measure of the alignments obtained by our approach outperform all other state of the art matching systems.

5. Acknowledgment. This work is supported by the science and technology projects in Fujian Province Department of Education(No. JA15356), National Natural Science Foundation of China (No. 61503082), Natural Science Foundation of Fujian Province (No. 2016J05145), Fujian Provincial key project(No. 2015H0009) and Fujian young teachers Educational Research Foundation-funded project(No. JA14217), and China Scholarship Council.

REFERENCES

- [1] Martinez-Gil, E. Alba and Aldana-Montes J. F., Optimizing ontology alignments by using genetic algorithms, *Nature Inspired Reasoning for the Semantic Web (NatuReS2008)*, vol.419, pp.31-45, 2008.
- [2] J.M.V. Naya, M.M. Romero and J.P. Loureiro, Improving ontology alignment through genetic algorithms, *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*, pp.240-259, 2010.
- [3] A-L. Ginsca and A. Iftene, Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment, *9th Roedunet Int Conference (RoEduNet)*, pp.118-122, 2010.
- [4] 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.
- [5] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, *Technical Report C- TR06*, vol.200, Erciyes university, engineering faculty, computer engineering department, 2005.
- [6] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Yong Liu and J. Pan, Multi-Strategy Ensemble Artificial Bee Colony Algorithm, *Information Sciences*, Vol. 279, pp. 587-603, 2014.
- [7] P. Tsai, M. K. Khan, J. Pan, and Bin-Yih Liao, Interactive Artificial Bee Colony Supported Passive Continuous Authentication System, *IEEE Systems Journal*, Vol. 8, No. 2, pp. 395-405, 2014.
- [8] P. Tsai, J. Pan, B. Liao and S. Chu, Enhanced Artificial Bee Colony Optimization, *International Journal of Innovative Computing, Information and Control*, Vol. 5, No. 12 (B), pp. 5081-5092, 2009.
- [9] J. Zhao, X. Fu, L. Lv, R. Wu, H. Wang, X. Yu and T. Fan, Opposition-Based Artificial Bee Colony Using Different Learning Models, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 7, No. 6, pp. 1206-1214, 2016.
- [10] T. K. Dao, T.S. Pan, T. T. Nguyen and S. S. Chu, A Compact Artificial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 6, No. 2, pp. 297-310, 2015.

- [11] M.S. Kiran and O. Fındık, A directed artificial bee colony algorithm, *Applied Soft Computing*, vol.26, pp.454-462, 2015.
- [12] G. Acampora, V. Loia, S. Salerno S and A. Vitiello, A hybrid evolutionary approach for solving the ontology alignment problem, *International Journal of Intelligent Systems*, vol.27, no.3, pp.189-216, 2012.
- [13] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer Publishing Company, 2007.
- [14] Ontology Alignment Evaluation Initiative (OAEI), Available at <http://oaei.ontologymatching.org/2014/>, Accessed on 15 June, 2016.
- [15] C. J. Van Rijsbergen, *Information Retrieval*, London: Butterworth, 1975.
- [16] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, *Information Sciences*, vol.192, pp.120-142, 2012.
- [17] A. Maedche and S. Staab, Measuring Similarity between Ontologies, *Proceeding of the International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pp.251-263, 2002.
- [18] G.A. Miller, WordNet: A lexical database for English, *Communications of the ACM*, vol.38, pp.39-41, 1995.
- [19] S. Melnik, H. Garcia-Molina and E. Rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, *18th International Conference on Data Engineering*, pp.117-128, 2002.