Journal of Network Intelligence ©2020 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **5**, Number **4**, November **2020**

Matching Ontologies through Compact Monarch Butterfly Algorithm

Xueqin Kou

School of Mechanical and Electronic Engineering
Xi'an University of Architecture and Technology
No.13 Yanta Road, Xi'an, Shaanxi, 710311, China
snow_kou78@163.com

Junhong Feng*

School of Computer Science and Engineering
Yulin Normal University
No.299 Education Middle Road, Yulin, Guanxi, 537000, China
*Corresponding Author: jgxyfjh@126.com

Received September 2020; revised October 2020

Abstract. *As a kernel technique of Semantic Web (SW), ontology is a state-of-the-art knowledge modeling technique that formally defines the entities and their relationships in a particular domain, which aims at overcoming the data heterogeneity problem. To bridge the semantic gap between two heterogeneous ontologies, it is necessary to execute the ontology matching process and find the semantic correspondences between ontology entities. Being inspired by the success of Swarm Intelligence Algorithm (SIA) based ontology matching technique in the ontology matching domain, this work further proposes a Compact Monarch Butterfly Algorithm (CMBA) based ontology matching technique to improve the quality of the ontology alignment. In particular, we propose two compact evolutionary operators, i.e. compact migration operator and compact adjusting operator, to approximate the behaviour of population-based Monarch Butterfly Algorithm (MBA), and a linguistic-based similarity measure to distinguish the heterogeneous entity mappings. The experimental results show that our approach outperforms the state-of-the-art ontology matching systems and SIA-based ontology matching techniques.*

Keywords: Ontology Matching; Compact Monarch Butterfly Algorithm; Compact Migration Operator; Compact Adjusting Operator

1. **Introduction.** As a kernel technique of Semantic Web (SW) [1], ontology is a state-of-the-art knowledge modeling technique that formally defines the entities and their relationships in a particular domain, which aims at overcoming the data heterogeneity problem. However, due to different requirements of the application, ontologies in the same domain could define an entity with different terminologies and in different context, yielding the ontology heterogeneity problem. To integrate the knowledge inside these overlapped ontologies and enhance the co-operation between knowledge-based intelligent systems, it is necessary to determine the heterogeneous entities in two ontologies to bridge their semantic gap, which is called ontology matching. The results obtained by an ontology matching technique consists of dozens of entity correspondences, which is the so-called ontology alignment.

With the development of SW, the scale of an ontology becomes larger and their semantic relationships become more and more complex, and thus, the determination of a high-quality ontology alignment is a non-trivial task [2]. In general, matching two ontologies has three steps: 1) use the similarity measure to distinguish the entity correspondences; 2) find all the correct entity correspondences from large-scale candidate mappings; 3) filter the error entity correspondence. In recent years, Swarm Intelligence Algorithm (SIA) has become the state-of-the-art method to address the various network optimization problems [3, 4, 5] and ontology heterogeneity problem [6, 7]. Genetics Ontology ALignment (GOAL) [8] first use Genetic Algorithm to optimize the aggregating weights of similarity matrices, which indirectly optimize the final alignment. After that, various SIAs [9, 10] are proposed one by one to improve the efficiency of ontology matching process, and among them, a newly emerging category of SIAs that utilize the compact encoding mechanism to reduce the memory consumption attracts researchers' interests. Xue et al. [11] first propose to use the compact GA (CGA) to optimize the alignment's quality, which takes the idea from GOAL but significantly improving its performance. CGA only uses one Probability Vector (PV) to approximately present the population, and to further improve its search ability, Xue et al. [12] propose a Compact Co-Evolutionary Algorithm (CCEA) that uses two PVs to respectively focus on the exploitation and exploration. CCEA is able to find more correct correspondences and therefore its results are better than CGA. Being inspired by the success of SIA-based ontology matching technique, this work further proposes a Compact Monarch Butterfly Algorithm (CMBA) [13] to determine and filter the alignment. To ensure the quality of the alignment, we propose a linguistic-based entity similarity measure that combines the syntax-based and Wordnet [14] based similarity measures to distinguish the heterogeneous entities.

In the rest, this paper is organized as follows: Section 2 defines the ontology heterogeneity problem and linguistic-based similarity measure; Section 3 presents the CMBA-based ontology matching technique in detail; Section 4 shows the experimental results; and Section 5 draws the conclusion.

2. Ontology Heterogeneity Problem and Linguistic-based Similarity Measure.

2.1. Ontology Heterogeneity Problem.

In general, an ontology consists of the concepts, their datatype properties and object properties, instances and axioms. In this work, we mainly focus on matching the entities such as concept, datatype property and object property, and in this context, an ontology O is defined as a 3-tuple (C, DP, OP) [15], where C, DP and OP are respectively referred to these entities. An entity correspondence *corr* is a 4-tuple $(e_{O_1,i}, e_{O_2,j}, simValue, \equiv)$, where $e_{O_1,i}$ and $e_{O_2,j}$ are two entities from source ontology O_1 and target ontology O_2, and $simValue \in [0,1]$ is their similarity value and $\equiv$ is the equivalence relationship. An ontology alignment A is the set of the entity correspondences. Ontology matching process takes two ontologies to be aligned as the input, uses some external resources, such as the digital dictionary and background knowledge base, to find the ontology alignment. How to efficiently determine a high-quality alignment in different application background is still a challenge. In this work, we regard the ontology matching process as a discrete optimization problem, which can be defined as follows [16, 17]:

$$\begin{cases} max \quad f(X) \\ s.t. \quad X = (x_1, x_2, ..., x_{|C_1|})^T \\ \qquad\quad x_i \in 0, 1, 2, \cdots, |C_2| \end{cases} \tag{1}$$

where $|C_1|$ and $|C_2|$ are the cardinalities of O_1 and O_2, $x_i = j$ means the ith class in O_1 is mapped to jth class in O_2 (in particular, $x_i = 0$ means the ith class in O_1 is mapped to none), and the objective is maximizing X's corresponding alignment's f-measure value [18]. In this work, we first find the concept correspondence set, and then use the class alignment to determine the mappings between datatype properties and object properties, respectively.

2.2. **Linguistic-based Similarity Measure.** The similarity measure is a function that takes as input two entities and outputs their similarity value. According to Ontology Alignment Evaluation Initiative (OAEI)'s report, 80% heterogenous entities can be distinguished through the names or labels of two entities [19]. On this basis, this work propose a linguistic-based similarity measure that uses Wordnet, a popular electronic dictionary, to distinguish the synonymous words, and Similarity Measure for Ontology Alignment (SMOA) [20], an syntax-based similarity measure to calculate two strings' similarity values. In particular, the linguistic-based similarity measure for two strings s_1 and s_2 can be defined as follows:

$$sim(s_1, s_2) = \begin{cases} 1 & \text{if } s_1 \text{ and } s_2\text{'s labels are synonymous in Wordnet} \\ SMOA(s_1, s_2) & \text{otherwise} \end{cases} \tag{2}$$

On this basis, two classes c_1 and c_2's similarity value is equal to $sim(c_1.label, c_2.label)$, and two datatype properties (or object properties) p_1 and p_2's similarity value is calculated as follows:

$$sim_p'(p_1, p_2) = \frac{sim(p_1.domain.label, p_2.domain.label) + sim(p_1.label, p_2.label) + sim(p_1.range.label, p_2.range.label)}{3} \tag{3}$$

3. **Compact Monarch Butterfly Algorithm.** Monarch Butterfly Algorithm (MBA) [13] is a recently proposed SIA, whose main idea is based on the migration characteristics of monarch butterflies. The positions of monarch butterflies in the first land and second land are updated by the migration operator and adjusting operator, respectively, which is repeated till the termination condition is satisfied. In this work, we propose a compact version of MBO to improve the performance of the original MBA. CMBA utilizes two Probability Vectors (PVs) [21] to respectively represent two swarms of butterflies in the first land and second land. This work uses the Gray code, a common binary encoding mechanism, to encode the entity correspondence. Each PV's length is equal to the solution's length, and the element in PV represents the probability of being 1 on the corresponding code. Through a PV, we can generate various solutions, and it is obvious that if all the elements are either 1 or 0, the solutions generated will be the same. In each evolutionary generation, we update PV by moving its element towards 1 or 0 according to the best solution found. In the next, we give the pseudo-code of CMBA in Algorithm 1.

In Algorithm 1 the compact migration operator and Compact Adjusting Operator are respectively designed as follows:

In Algorithms 2 and 3, PV_{land1} and PV_{land2} are respectively two lands' PVs, sol_{land1} and sol_{land2} are respectively two butterflies in the first land and second land, sol_{best} is an elite solution, $randNum()$ generates a random number and $p \in [0, 1]$ is the migration probability.

Algorithm 1 Compact Monarch Butterfly Algorithm

```
********* Initialization *********
initialize all elements in $PV_{land1}$ and $PV_{land2}$ as 0.5;
$sol_{best1} = PV_{land1}.newSolution()$;
$sol_{best2} = PV_{land2}.newSolution()$;
$sol_{best} = compete(sol_{best1}, sol_{best1})$;
while $t < MaxGen$ do
    $sol_{land1} = CMO()$; //Compact Migration Operator, see also Algorithm 2
    $sol_{land2} = CAO()$; //Compact Adjusting Operator, see also Algorithm 3
    $sol_{best1} = compete(sol_{land1}, sol_{best1})$;
    $sol_{best2} = compete(sol_{land2}, sol_{best2})$;
    $sol_{best} = compete(sol_{best1}, sol_{best2}, sol_{best})$;
    ********* Update two PV *********
    for int $i = 0$; $i < sol_{best1}.length$; $i++$ do
        if $sol_{best1,i} == 1$ then
            $PV_{land1} = PV_{land1} + \frac{1}{sol_{best}.length}$;
        else
            $PV_{land1} = PV_{land1} - \frac{1}{sol_{best}.length}$;
        end if
    end for
    for int $i = 0$; $i < sol_{best2}.length$; $i++$ do
        if $sol_{best2,i} == 1$ then
            $PV_{land2} = PV_{land2} + \frac{1}{sol_{best}.length}$;
        else
            $PV_{land2} = PV_{land2} - \frac{1}{sol_{best}.length}$;
        end if
    end for
    $t = t + 1$;
end while
return $sol_{best}$;
```

Algorithm 2 Compact Migration Operator

```
$sol_{land1} = PV_{land1}.newSolution()$;
for int $i = 0$; $i < sol_{land1}.length$; $i++$ do
    if $randNum() < p$ then
        $sol_{land1,i} = PV_{land1,i}.newCode()$;
    else
        $sol_{land1,i} = PV_{land2,i}.newCode()$;
    end if
end for
return $sol_{land1}$;
```

4. **Experiment.** To test the effectiveness of CMBA-based ontology matching technique, the experiment utilizes the well-known OAEI's Biblio testing cases [1], and a brief description is shown in Table 1. There are three files in each testing case, i.e. two ontologies to be matched and one reference alignment for evaluating the alignment obtained.

[1] http://oaei.ontologymatching.org/2016/benchmarks/

Algorithm 3 Compact Adjusting Operator

```
$sol_{land2} = PV_{land2}.newSolution();$
for int $i = 0$; $i < sol_{land1}.length$; $i++$ do
    if $randNum() < p$ then
        $sol_{land2,i} = sol_{best,i};$
    else
        $sol_{land2,i} = PV_{land1,i}.newCode();$
    end if
end for
return $sol_{land2};$
```

Table 1. The brief description on biblio testing cases

Testing Case	Brief description
101-104	Two ontologies are the same.
201-210	Two ontologies are heterogeneous in terms of the language features.
221-247	Two ontologies are heterogeneous in terms of the conceptual structure.

Table 2. Comparison on the quality of the alignment.

Matching Technique	recall	precision	f-measure
edna	0.51	0.35	0.41
AML	0.24	1.00	0.38
CroMatch	0.83	0.96	0.89
Lily	0.83	0.97	0.89
LogMap	0.39	0.93	0.55
LogMapLt	0.50	0.43	0.46
XMap	0.40	0.95	0.56
LogMapBio	0.24	0.48	0.32
GA	0.74	0.86	0.79
MBA	0.82	0.94	0.87
CGA	0.70	0.91	0.79
CCEA	0.85	0.95	0.89
CMBA	0.88	0.97	0.91

We compare our approach with GA [8], MBA [13], CGA [11], CCEA [12] and OAEI's participants in terms of alignment's quality, which is measured by f-measure value. MBA, CGA and CCEA are referred to their literatures, and the configuration of CMBA is as follows: maximum generation $maxGen = 2000$, migration probability $p = 0.4$, similarity threshold $threshold = 0.95$. The results of the SIA-based ontology matching techniques are shown in Table 2 are the average results of 30 independent runs, and OAEI's participants' results can be found on OAEI's web site.

As can be seen from Table 2, CMBA has high recall value, which shows its capability of finding correct entity correspondences in the large search space; its precision value ranks the second, which show the effectiveness of the proposed linguistic-based similarity measure; and finally, its f-measure value is the highest, which show that it can effectively determine high-quality ontology alignment. In particular, comparing with MBA, the gains in solution quality are achieved respectively due to CMBA's particular competitive learning, which is effective to lead the algorithm to determine the optimal solution, and the simplicity of CMBA, which does not require all the mechanisms of MBA, rather the few steps in the algorithm are small and simple.

5. **Conclusion.** To bridge the semantic gap between two heterogeneous ontologies, it is necessary to execute the ontology matching process and find the semantic correspondences between ontology entities. To improve the quality of the ontology alignment, being inspired by the success of SIA-based ontology matching technique in the ontology matching domain, this work further propose a CMBA-based ontology matching technique. In particular, we propose two compact evolutionary operators based on two PVs to approximate the population-based MBA's behaviour, and a linguistic-based similarity measure to distinguish the heterogeneous entity mappings. The experimental results show that CMBA outperforms the state-of-the-art ontology matching systems and SIA-based ontology matching techniques.

Acknowledgment. This research was supported by Natural Science Foundation of Guangxi Province (No. 2018JJA170050), and Improvement Project of Basic Ability for Young and Middle-aged Teachers in Guangxi Universities (No. 2017KY0541).

REFERENCES

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34-43, 2001.

[2] Xingsi Xue and Yuping Wang. Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio. *Artificial Intelligence*, 223:65-81, 2015.

[3] Fuquan Zhang, Tsu-Yang Wu, Yiou Wang, Rui Xiong, Gangyi Ding, Peng Mei, and Laiyang Liu. Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction. *IEEE Access*, 8:104555-104564, 2020.

[4] Lanlan Kang, Ruey-Shun Chen, Yeh-Cheng Chen, Chung-Chei Wang, Xingguan Li, and Tsu-Yang Wu. Using cache optimization method to reduce network traffic in communication systems based on cloud computing. *IEEE Access*, 7:124397-124409, 2019.

[5] Lanlan Kang, Ruey-Shun Chen, Naixue Xiong, Yeh-Cheng Chen, Yu-Xi Hu, and Chien-Ming Chen. Selecting hyper-parameters of gaussian process regression based on non-inertial particle swarm optimization in internet of things. *IEEE Access*, 7:59504-59513, 2019.

[6] Xingsi Xue and Junfeng Chen. Using compact evolutionary tabu search algorithm for matching sensor ontologies. *Swarm and Evolutionary Computation*, 48:25-30, 2019.

[7] Xingsi Xue, Haiyan Yang, and Jie Zhang. Using population-based incremental learning algorithm for matching class diagrams. *Data Science and Pattern Recognition*, 3(1):1-8, 2019.

[8] Jorge Martinez-Gil, Enrique Alba, and José F Aldana-Montes. Optimizing ontology alignments by using genetic algorithms. *In Proceedings of the workshop on nature based reasoning for the semantic Web. Karlsruhe, Germany*, 2008.

[9] Xingsi Xue and Yuping Wang. Using memetic algorithm for instance coreference resolution. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):580-591, 2015.

[10] Jürgen Bock and Jan Hettenhausen. Discrete particle swarm optimisation for ontology alignment. *Information Sciences*, 192:152-173, 2012.

[11] Xingsi Xue, Jianhua Liu, Pei-Wei Tsai, Xianyin Zhan, and Aihong Ren. Optimizing ontology alignment by using compact genetic algorithm. *In 2015 11th International Conference on Computational Intelligence and Security (CIS)*, pp. 231-234. IEEE, 2015.

[12] Xingsi Xue, Jie Chen, Junfeng Chen, and Dongxu Chen. Using compact coevolutionary algorithm for matching biomedical ontologies. *Computational intelligence and neuroscience*, 2018:1-8, 2018.

[13] Gai-Ge Wang, Xinchao Zhao, and Suash Deb. A novel monarch butterfly optimization with greedy strategy and self-adaptive. *In 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*, pp. 45-50. IEEE, 2015.

[14] George A Miller.Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39-41, 1995.

[15] Xingsi Xue and Jiawei Lu. A compact brain storm algorithm for matching ontologies. *IEEE Access*, 8:43898-43907, 2020.

[16] Xingsi Xue. A compact firefly algorithm for matching biomedical ontologies. *Knowledge and Information Systems*, pp. 1-17, 2020.

[17] Xingsi Xue and Junfeng Chen. Optimizing sensor ontology alignment through compact co-firefly algorithm. *Sensors*, 20(7):2056, 2020.

[18] Cornelis Joost Van Rijsbergen. Foundation of evaluation. *Journal of documentation*, 30(4):365-373, 1974.

[19] Xingsi Xue and Xin Yao. Interactive ontology matching based on partial reference alignment. *Applied Soft Computing*, 72:355-370, 2018.

[20] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. *In International Semantic Web Conference*, pp. 624-637. Springer, 2005.

[21] Xingsi Xue and Jianhua Liu. Collaborative ontology matching based on compact interactive evolutionary algorithm. *Knowledge-Based Systems*, 137:94-103, 2017.