ICIC Express Letters ICIC International ©2017 ISSN 1881-803X
Volume **11**, Number **1**, January **2017** pp. **53–58**

USING COMPACT MEMETIC ALGORITHM FOR OPTIMIZING ONTOLOGY ALIGNMENT

Xingsi Xue[1,2,*], Pei-Wei Tsai[1,2] and Jinshui Wang[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 350118, P. R. China
*Corresponding author: jack8375@gmail.com

Received June 2016; accepted September 2016

Abstract. *In order to support semantic inter-operability in many domains through disparate ontologies, we need to identify correspondences between the entities across different ontologies, which is commonly known as ontology matching. One of the challenges in ontology matching domain is how to select weights and thresholds in ontology aligning process in order to aggregate the various similarity measures to obtain a satisfactory alignment, so called ontology meta-matching problem. Nowadays, the most suitable methodology to address the ontology meta-matching problem is through Evolutionary Algorithm (EA), and the Memetic Algorithm (MA) based approaches are emerging as a new efficient methodology to face the meta-matching problem. Moreover, for dynamic applications, it is necessary to perform the system self-tuning process at run time, and thus, efficiency of the configuration search strategies becomes critical. To this end, in this paper, we propose a problem-specific compact Memetic Algorithm, in the whole ontology matching process of ontology meta-matching system, to optimize the ontology alignment. The experimental results show that our proposal is able to highly improve the efficiency of determining the optimal alignments through MA based approach while keeping the quality of the alignments obtained.*
Keywords: Compact Memetic Algorithm, Ontology meta-matching problem, Similarity aggregation

1. **Introduction.** Nowadays, numerous alignment systems have arisen and each of them could provide, in a fully automatic or semi-automatic way, a numerical value of similarity between elements from separate ontologies that can be used to determine whether those elements are semantically similar or not. However, how to select weights and thresholds in ontology aligning process in order to aggregate the various similarity measures to obtain a satisfactory alignment, so called meta-matching problem, is still a challenging problem. Recently, Evolutionary Algorithm (EA), appears as the most suitable methodology to address the meta-matching problem [1], and the Memetic Algorithm (MA) based approaches are emerging as a new efficient methodology to face the meta-matching problem.

For dynamic applications, it is necessary to perform the similarity measures combination and system self-tuning at run time, and thus, beside quality (correctness and completeness) of the aligning results, the execution time and main memory of the aligning process are of prime importance. Therefore, the traditional population-based EA can be inadequate, and a memory saving approach must be applied. According to [2], if properly designed, a population-based algorithm with a very small population size can efficiently solve large scale problems. The very first compact EA was the compact Genetic Algorithm (cGA) which was put forward by Harik et al. [3] in 1999. In cGA, the population was represented as a probability distribution over the set of solutions, reducing the heavy memory requirements in the traditional GA's and the probabilistic model update mechanism which was led by the better individual selected from the only two offspring in every generation, greatly enhanced the optimizing speed. Later, Baraglia et al. [4] proposed a memetic variant of cGA to enhance the convergence performance of the algorithm in the presence of a relatively high number of dimensions, and Ahn and Ramakrishna [5] introduced two elitism strategies into cGA, i.e., strong and weak elitism, which significantly improved the performance of the original cGA. In this paper, on the basis of our former work in [7] and [8], we further propose a problem-specific compact MA to efficiently identify both the weights for the similarity measure aggregation task and the similarity threshold regardless of the knowledge about the ontology features, data availability and human intervention. In particular, in this work, a single objective optimal model of ontology meta-matching problem is proposed and a problem-specific compact MA is designed for the ontology meta-matching problem.

The rest of the paper is organized as follows: Section 2 describes the compact Memetic Algorithm for ontology meta-matching problem; Section 3 shows the experiments carried out to show the effectiveness of the proposal; finally, Section 4 draws the conclusions.

2. Compact Memetic Algorithm for Ontology Meta-Matching Problem.

2.1. The single objective optimal model for ontology meta-matching problem.

In our work, we take maximizing the value of MatchFmeasure as the goal, which is a rough alignment evaluation measure [7], and given n similarity measures to aggregate, the single objective optimal model for ontology meta-matching problem can be defined as follows:

$$\left\{\begin{array}{ll} \max & MatchFmeasure(X) \\ \text{s.t.} & X = (x_1, x_2, \dots, x_n, x_{n+1})^T \\ & \sum_{i=1}^{n} x_i = 1 \\ & x_i \in [0,1], i = 1, \dots, n+1 \end{array}\right. \tag{1}$$

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. Chromosome encoding.

In our proposal, a Probability Vector (PV) [2], which is a binary vector whose gene's value is in $[0,1]$, is utilized to characterize the entire population in population-based MA, and it can be divided into two parts: one stands for the correspondences in the alignment and the other for a threshold. We represent both the correspondences and threshold through the binary coding mechanism in the field of computer according to the number of target ontology entities and the numerical accuracy of threshold. When decoding, we calculate the corresponding decimal numbers. In the first part, the numbers obtained represent the indexes of the target entities, and in particular, the value 0 means corresponding source instance does not map to any entity. While in the second part, the decimal number should be plus the numerical accuracy. This is because given the number accuracy *numAccuracy*, the threshold value will be expressed by an integer in $\left[0, \frac{1}{numAccuracy}\right]$ with a binary code. Moreover, after decoding, if a decimal number a obtained is larger than the upper boundary u, we will replace it with $\frac{u}{a}$.

2.3. Binary crossover.

In this work, we utilize a binary crossover operator to implement the local search process [6]. To be specific, given an elite solution $solution_{elite}$ and a new generated solution $solution_a$, a new solution is generated as follows: a bit of $solution_a$ is randomly selected, e.g., $solution_a[i]$, and copied into the ith bit of $solution_{elite}$. Subsequently, a set of random numbers in $[0,1]$ is generated, and as long as $rand(0,1)$ is smaller than the crossover possibility cr, the bit values from $solution_a$ are copied into $solution_{elite}$. The first time that $rand(0,1)$ is larger than p_c, the copy process is interrupted.

2.4. **Detail of compact Memetic Algorithm.** In this work, we present a compact version of MA to face the ontology meta-matching problem, which simulates the behaviour of population-based MA by employing, instead of a population of solutions, the probabilistic representation of the population. Thus, a run of our proposal is able to highly improve the efficiency of solving large scale matching problem in terms of both runtime and memory consumption. In the following, the pseudo-code of the compact MA is given:

```
Input:
• $instanceSet_1$ and $instanceSet_2$: two instance sets to match;
• $num$: the length of chromosome;
• $maxGen$: maximum number of generations;
• $minVP$: minimum virtual population;
• $maxVP$: maximum virtual population;
• $p_c$: exponential crossover probability;
• $p_m$: PV mutation probability;
• $lPoss$: PV's lower possibility boundary;
• $uPoss$: PV's upper possibility boundary.
Output: $ind_{elite}$: the solution with best f-measure.

Step 1) Initialization:
  1.  generation = 0;
  2.  for ($i = 0$; $i < num$; $i$++)
  3.    $PV[i] = 0.5$;
  4.  end for
  5.  generate an individual $ind_{elite}$ by means of $PV$;

Step 2) Update PV:
  Step 2.1 Exploitation:
  6.    generate $ind_{temp}$ by means of $PV$;
  7.    $ind_{new} = binCross(ind_{elite}, ind_{temp})$;
  8.    $[winner, loser] = compete(ind_{elite}, ind_{new})$;
  9.    if ($winner == ind_{new}$)
  10.     $ind_{elite} = ind_{new}$;
  11.   end if
  12.   for ($i = 0$; $i < num$; $i$++)
  13.     if ($winner[i] == 1$)
  14.       $PV[i] = PV[i] + VP$;
  15.     else
  16.       $PV[i] = PV[i] - VP$;
  17.     end if
  18.   end for
  Step 2.2 Exploration:
  19.   if (all PV bits $> uPoss$ or $< lPoss$)
  20.     for ($i = 0$; $i < num$; $i$++)
  21.       if ($rand(0, 1) < p_m$)
  22.         $PV[i] = 1 - PV[i]$;
  23.       end if
  24.     end for
  25.   end if

Step 3) Stopping Criteria:
```

```
26. if ($maxGen$ is reached or each bit of PV is either 1 or 0)
27.   stop and output $ind_{elite}$;
28. else
29.   generation = generation+1;
30.   go to Step 2);
31. end if
```

In the evolutionary process (Step 2), we first execute the exploitation through the exponential crossover and update the PV with self-adaptive virtual population in Step 2.1. Then, in Step 2.2, we judge the exploration condition by checking the values of all PV bits to see whether they are all larger than $uPoss$ or smaller than $lPoss$. And if it is so, PV bits will be flipped according to the PV mutation probability p_m. In particular, if all PV bits are all larger than $uPoss$ or smaller than $lPoss$, the individuals generated by PV will be approximately the same, i.e., the algorithm is about to converge. Therefore, we apply a strong mutation on PV to preventing the premature convergence. When the algorithm approaches the $maxGen$, $maxVP$ will ensure the algorithm to converge, and in this way, we balance the exploitation and exploration of the algorithm. In this work, we set $lPoss = 0.3$, $uPoss = 0.7$, $p_m = 0.6$ and $maxVP = 0.35$.

3. Experimental 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) 2015 [16] and commonly used for experimentation about ontology alignment problem. In detail, each test case, see Table 1, represents a specific alignment task devoted to align a reference ontology with its variation.

Table 1. Brief description of benchmarks in OAEI 2015

ID	Brief description
101-104	The ontologies under alignment are the same or the first one is the OWL Lite restriction of the second one.
201-210	The ontologies under alignment have the same structure, but different lexical and linguistic features.
221-231	The ontologies under alignment have the same lexical and linguistic features, but different structure.
301-304	The ontologies under alignment are real world cases.

3.1. Experiment configuration.

In order to compute the confidence value, which represents the similarity level existing between the two entities composing a correspondence, the similarity measure is used. According to the literature [11], the similarity measure could be categorized into syntactic, linguistic and taxonomy-based measures. In this work the similarity measures are as follows:

- SMOA distance [12] (syntactic measure);
- Wordnet-based distance [14] (linguistic measure);
- Similarity Flooding based distance [15] (taxonomy-based measure).

The compact MA uses the following parameters which represent a trade-off 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:

- Numerical accuracy = 0.01;
- The fitness is the value of MatchFmeasure;

- Maximum number of generations: 300;
- Minimum virtual population: 0.02;
- Maximum virtual population: 0.35;
- Exponential crossover probability: 0.8;
- PV mutation probability: 0.6;
- PV's lower possibility boundary: 0.3;
- PV's upper possibility boundary: 0.7.

3.2. **Results and analysis.** Table 2 presents the comparison of the f-measure, average executing time and main memory consumption per generation by the approach based on GA with our way, respectively. All the values shown in Table 2 are the average figures in ten independent runs.

As can be seen from Table 2 that, in all the benchmarks, the alignments' quality obtained by two approaches are identical to each other. Moreover, our approach dramatically improves the executing time and the main memory consumption. Specifically, the improvement degree is on average by 68.42% and 88.92% respectively. According to the experiment results shown above, comparing with the approach based on MA, the utilization of compact MA is able to highly reduce the executing time and main memory consumption of the parameter tuning process while at the same time ensuring the correctness and completeness of the alignments.

Table 2. Comparison of the alignments obtained by Memetic Algorithm based approach with our approach

ID	*f-measure* (*time*(*ns*) – *memory*(*byte*)) (Memetic Algorithm based approach)	*f-measure* (*time*(*ns*) – *memory*(*byte*)) (compact Memetic Algorithm based approach)
101	1.00 (1,455,158,844 – 53,020,008)	1.00 (813,249,456 – 24,235,573)
103	1.00 (1,536,963,526 – 58,485,120)	1.00 (862,056,615 – 34,945,742)
104	1.00 (3,722,034,212 – 64,511,680)	1.00 (1,379,325,674 – 23,429,349)
201	0.94 (25,257,739,836 – 803,790,664)	0.94 (1,230,340,246 – 81,847,440)
203	0.99 (27,560,007,326 – 819,355,924)	0.99 (8,496,399,655 – 84,344,216)
204	0.98 (27,358,214,633 – 735,096,978)	0.98 (8,560,623,843 – 92,834,128)
205	0.93 (25,128,261,706 – 829,535,204)	0.93 (8,489,204,916 – 62,311,092)
206	0.70 (25,473,189,358 – 806,270,340)	0.70 (8,496,638,349 – 91,960,442)
221	1.00 (24,230,903,087 – 785,948,732)	1.00 (8,362,466,595 – 96,327,008)
222	1.00 (22,245,786,725 – 919,267,364)	1.00 (8,752,852,325 – 93,217,368)
223	0.99 (25,347,676,347 – 822,302,384)	0.99 (7,404,065,227 – 85,383,034)
224	1.00 (1,345,778,565 – 824,281,424)	1.00 (831,696,314 – 85,530,230)
225	1.00 (16,947,875,524 – 887,774,568)	1.00 (6,984,969,343 – 51,192,335)
228	1.00 (13,389,655,330 – 874,386,136)	1.00 (5,304,332,934 – 84,822,962)
230	1.00 (13,790,248,622 – 776,847,868)	1.00 (7,521,722,323 – 84,754,825)
231	1.00 (26,996,779,235 – 684,253,948)	1.00 (6,224,469,398 – 76,237,452)
301	0.75 (2,622,022,072 – 85,142,512)	0.75 (1,340,283,344 – 68,458,400)
302	0.74 (27,129,635,292 – 620,275,960)	0.74 (7,448,452,230 – 62,334,237)
304	0.93 (20,210,866,347 – 603,368,408)	0.93 (6,258,229,340 – 51,127,014)
Avg.	0.94 (17,460,462,978 – 634,416,590)	0.94 (5,513,756,744 – 70,278,570)

4. **Conclusion.** One of these challenges in ontology matching domain is how to select weights and thresholds in ontology aligning process in order to aggregate the various similarity measures to obtain a satisfactory alignment, which is called ontology meta-matching problem. The most suitable methodology to address the meta-matching problem is through EA, and MA based approaches are emerging as a new efficient methodology to face the meta-matching problem. Moreover, for dynamic applications, it is necessary to perform the system self-tuning process at run time, and thus, efficiency of the configuration

search strategies becomes critical. To this end, in this paper, we present a problem-specific compact MA, in the whole similarity aggregation step of meta-matching system, to determine the solutions of weights and thresholds for the ontology meta-matching system. The experimental results show that our proposal is able to highly improve the efficiency of determining the optimal alignments through MA based approach while keeping the quality of the alignments obtained.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61503082 and 61402108), Natural Science Foundation of Fujian Province (No. 2016J05145 and 2014J01218), Science and Technology Project in Fujian Province Department of Education (No. JA15256) and China Scholarship Council.

REFERENCES

[1] J. Martinez-Gil and J. F. A. Montes, Evaluation of two heuristic approaches to solve the ontology meta-matching problem, *Knowledge and Information Systems*, vol.26, no.2, pp.225-247, 2011.
[2] K. E. Parsopoulos, Cooperative micro-differential evolution for high dimensional problems, *Proc. of the Conference on Genetic and Evolutionary Computation*, Montreal, Canada, pp.531-538, 2009.
[3] G. R. Harik, F. G. Lobo and D. E. Goldberg, The compact genetic algorithm, *IEEE Trans. Evolutionary Computation*, vol.3, no.4, pp.287-297, 1999.
[4] R. Baraglia, J. I. Hidalgo and R. Perego, A hybrid heuristic for the traveling salesman problem, *IEEE Trans. Evolutionary Computation*, vol.5, no.6, pp.613-622, 2001.
[5] C. W. Ahn and R. S. Ramakrishna, Elitism based compact genetic algorithms, *IEEE Trans. Evolutionary Computation*, vol.7, no.4, pp.367-385, 2003.
[6] F. Neri, G. Iacca and E. Mininno, Compact optimization, in *Intelligent Systems Reference Library*, vol.38, pp.337-364, Springer, Berlin, Germany, 2013.
[7] X. Xue, J. Liu, P. Tsai, X. Zhan and A. Ren, Optimizing ontology alignment by using compact genetic algorithm, *The 11th International Conference on Computational Intelligence and Security*, Guangzhou, China, pp.231-234, 2016.
[8] X. Xue and Y. Wang, Using memetic algorithm for instance coreference resolution, *IEEE Trans. Knowledge and Data Engineering*, vol.28, no.2, pp.580-591, 2016.
[9] J. Wang, Z. Ding and C. Jiang, Gaom: Genetic algorithm based ontology matching, *Proc. of IEEE Asia-Pacific Conference on Services Computing*, Guangzhou, China, pp.617-620, 2006.
[10] G. Acampora, V. Loia, S. Salerno and A. Vitiello, A hybrid evolutionary approach for solving the ontology alignment problem, *International Journal of Intelligent Systems*, vol.27, pp.189-216, 2012.
[11] E. Rahm and P. Bernstein, A survey of approaches to automatic schema matching, *The International Journal on Very Large Data Bases*, vol.10, no.4, pp.334-350, 2001.
[12] G. S. G. Stoilos and S. Kollias, A string metric for ontology alignment, *Proc. of the 4th International Semantic Web Conference*, Galway, Ireland, pp.623-637, 2005.
[13] W. Winkler, *The State Record Linkage and Current Research Problems*, Technical Report RR99-04, Statistics of Income Division, Washington, D.C., USA, 1999.
[14] G. A. Miller, Wordnet: A lexical database for English, *Communications of the ACM*, vol.38, no.11, pp.39-41, 1995.
[15] H. G.-M. S. Melnik and E. Rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, *The 18th International Conference on Data Engineering*, Shanghai, China, pp.117-182, 2002.
[16] OAEI, *Ontology Alignment Evaluation Initiative – 2015 Campaign*, http://oaei.ontologymatching.org/2015, 2015.