



Research article

Ternary compound ontology matching for cognitive green computing

Wei-Min Zheng¹, Qing-Wei Chai¹, Jie Zhang² and Xingsi Xue^{3,*}

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

² School of Computer Science and Engineering, Yulin Normal University, Yulin 537000, China

³ School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China

* **Correspondence:** Email: jack8375@gmail.com; Tel: +8615359746498.

Abstract: Cognitive green computing (CGC) dedicates to study the designing, manufacturing, using and disposing of computers, servers and associated subsystems with minimal environmental damage. These solutions should provide efficient mechanisms for maximizing the efficiency of use of computing resources. Evolutionary algorithm (EA) is a well-known global search algorithm, which has been successfully used to solve various complex optimization problems. However, a run of population-based EA often requires huge memory consumption, which limited their applications in the memory-limited hardware. To overcome this drawback, in this work, we propose a compact EA (CEA) for the sake of CGC, whose compact encoding and evolving mechanism is able to significantly reduce the memory consumption. After that, we use it to address the ternary compound ontology matching problem. Six testing cases that consist of nine ontologies are used to test CEA's performance, and the experimental results show its effectiveness.

Keywords: cognitive green computing; ternary compound ontology matching; compact evolutionary algorithm

1. Introduction

Cognitive green computing (CGC) amalgams the properties of cognitive computing [1] and green computing [2] to deal with such challenging issues like, energy optimization, carbon emission, resource utilization intelligently. In particular, CGC aims at providing efficient mechanisms for the tasks such as maximizing the efficiency of use of computing resources [3]. Evolutionary algorithm (EA) [4] is a well-known global search algorithm, which has been successfully used to solve various complex optimization problems. EA heuristically explores the decision space by maintaining a population which

might contain hundreds of chromosomes. Thus, a run of population-based EA often requires huge memory consumption, which limited their applications in the memory-limited hardware, e.g., mobile systems, industrial robots, etc. To overcome this drawback, in this work, we propose a Compact EA (CEA) [5] for the sake of CGC, and use it to address the ternary compound ontology matching problem. CEA uses a Probability Array (PA) to represent a population and execute the evolutionary operations, which is able to reduce the algorithm's memory consumption and searching performance.

Ontology is a cutting-edge knowledge modeling technique, which has been widely used to model various domain knowledge. It is impractical to develop and maintain a top-level ontology that is able to meet all the requirements of different applications, and thus, various domain ontologies are developed independently for different purposes [6]. Although they own significant overlapped information, a concept could be described with different granularity, yielding the ontology heterogeneity problem. For instance, describing a patient's record may include using SNOMED-CT for clinical methods employed, LOINC for laboratory analyses and results, ICD-10 for diagnoses and ATC for coding any prescribed antibiotics. Matching ontologies is able to determine semantically identical concepts among heterogenous ontologies, which is a feasible approach to integrate the knowledge in different ontologies. Traditionally, matching ontologies is the procedure that takes input two ontologies, and returns the concept mappings between two ontologies. However, if multiple ontologies are necessary to accurately describe the information and knowledge, to allow their communications, there is a need to establish links among multiple ontologies. This need motivates a new type of complex ontology matching, i.e., compound ontology matching [7], which involves more than two ontologies. A common case of the compound matching is the ternary compound matching, where two concepts from two distinct ontologies are mapped with a concept from the third ontology. For instance, the concept "broad forehead" in the human phenotype ontology (HP) is equivalent to the union of the concept "increased width" in phenotypic quality ontology (PATO) and the concept "forehead" in the foundation model of anatomy (FMA) ontology, i.e., "broad forehead" \equiv "increased width" \cap "forehead".

Since 2018, ontology alignment evaluation initiative (OAEI)* has presented the complex track that evaluates the matching techniques which generate complex correspondences. But there is still no testing case on the compound ontology matching problem, and currently, there are only two compound ontology matching techniques that are proposed by Pesquita and Oliveira [7, 8]. The former first determines the anchors, i.e., highly similar concept pairs, in two ontologies, and then match them to the third ontology. On the basis of the first work, Oliveira et al. further makes use of agreement maker light (AML) to built for the word matcher, and a weighted Jaccard index to aggregate the similarity values, whose computational complexity is large. Moreover, the completeness and accuracy of the determined compound ontology alignments are not high. Being inspired by the success of EA in the ontology matching domain [9–13], this work proposes to make use of EA to address the ternary compound ontology matching problem. However, population-based EA suffers from significant computational complexity, which hampers their application in the real matching tasks. To maximize the utilization of computer hardware and improve the algorithm's efficiency, this work first construct an optimal model for the ternary compound ontology matching problem, and then presents a concept similarity measure to calculate the confidence of a ternary compound correspondence, finally, a CEA with compact encoding and evolving mechanism is proposed to efficiently address this problem.

The rest of the paper is organized as follows: section 2 defines the ternary compound ontology

*<http://oaei.ontologymatching.org/>

matching problem and the concept similarity measure; section 4 describes the problem-specific CEA in details; section 5 shows the experimental results; and finally, section 6 draws the conclusion.

2. Ternary compound ontology matching problem

There are many definitions on the Ontology, and in this work, for the convenience, an ontology is defined as a 5-tuple (C, DP, OP, I, λ) , where C is the concept set, DP and OP are respectively the sets of inner property and outer property, I is the set of instance, and λ is the set of axiom [14, 15]. In particular, an inner property (or datatype property) describes a concept's individual feature, while an outer property (or object property) describes its relationships with other concepts. A ternary correspondence is also a 3-tuple $(c_1, c_2, c_3, \equiv, sim)$, where c_1, c_2 and c_3 are respectively the concepts of the first, second and third ontologies, \equiv is the relationship of equivalence, i.e., $c_1 \equiv c_2 \cap c_3$, and sim is the confidence or similarity value of this correspondence. The ternary compound matching process takes as input three ontologies and an external background knowledge base (e.g., unified medical language system (UMLS) [16]), and outputs a set of ternary compound correspondences, i.e., the ternary compound ontology alignment.

The traditional evaluation metrics work on the basis of the reference alignment provided by the expert. However, the reference alignment is not always available especially when the entity scale is huge. To overcome this drawback, we propose an approximate metric, which is based on the observation that the more ternary compound alignment correspondences found, the higher probability that it contains more correct correspondences, i.e., the alignment's completeness would be higher; and similarly, the higher mean similarity value of all the correspondences in an alignment, the higher probability that it contains less incorrect correspondences, i.e., the alignment's accuracy would be higher [17]. In particular, the approximate metric for evaluating a ternary compound ontology alignment's quality is defined as follows:

$$f(A) = \sqrt{\frac{|A|}{\max\{|C_1|, |C_2|, |C_3|\}}} \times \frac{\sum_{i=1}^{|A|} sim_i}{|A|} \quad (2.1)$$

where $|A|$, $|C_j|$ are respectively the cardinalities of the alignment A and the j th ontology's concept set, sim_i is the confidence value of the i th correspondence. Here, $\frac{|A|}{\max\{|C_1|, |C_2|, |C_3|\}}$ measures how many ternary compound alignment correspondences are found, $\frac{\sum_{i=1}^{|A|} sim_i}{|A|}$ calculates the average similarity measure, and function $f()$ takes both of them into consideration.

On this basis, the mathematical model of ternary compound ontology matching problem can be defined as follows:

$$\begin{cases} \max & f(X) \\ \text{s.t.} & X = (X^1, X^2, \dots, X^{|C_1|})^T \\ & X^i = (x_1^i, x_2^i)^T, i \in \{1, 2, \dots, |C_1|\} \\ & x_1^i \in \{0, 1, 2, \dots, |C_2|\}, x_2^i \in \{0, 1, 2, \dots, |C_3|\} \end{cases} \quad (2.2)$$

where $|C_j|$ is the cardinality of the j th ontology's concept set, and $X_i, i = 1, 2, \dots, |C_1|$ represents the i th correspondence. In particular, $x_1^i = 0$ (or $x_2^i = 0$) means the i th concept in the first ontology is mapped to none in the second (or third) ontology.

3. Concept similarity measure

Concept similarity measure (CSM) is a function measuring to what extent the concepts in a correspondence are similar, and its effectiveness directly affects the alignment's quality [18]. Essentially, CSM used in this work borrows the idea from Oliveira et al. [7], but we more focus on the concept's lexical and linguistic features, which are the most commonly used information in the ontology matching domain [19]. Given three concepts c_1 , c_2 and c_3 , and their labels as $label_1$, $label_2$ and $label_3$, we first pre-process these labels by: (1) remove their numbers, punctuations and stop-words; (2) split the labels into words and convert them into lower-case; (3) lemmatizing and stemming the English words. Then, we get the corresponding word set W_1 , W_2 and W_3 , which are used to calculate c_1 , c_2 and c_3 's similarity value in two sequential steps. First, we calculate the similarity of c_1 , c_2 as follows:

$$sim(c_1, c_2) = \frac{\sum_{i=1}^{|W_1|} \max \{sim'(w_{1,i}, w_{2,j})\}}{|W_1|} \quad (3.1)$$

where $j = 1, 2, \dots, |W_2|$. Here, for each word in W_1 , say $w_{1,i}$, we find its most similar word in W_2 , say $w_{2,j}$, and calculate their similarity value $sim'(w_{1,i}, w_{2,j})$. After that, we sum all the similarity values with respect to the words in W_1 , and divide it by the cardinality of W_1 to obtain two concepts' similarity value. In particular, $sim'()$ is defined as follows:

$$sim'(w_1, w_2) = \begin{cases} 1, & \text{if } w_1 \text{ and } w_2 \text{ are synonymous in UMLS} \\ sim_N(w_1, w_2), & \text{otherwise} \end{cases} \quad (3.2)$$

where UMLS is first used to determine whether two words w_1 and w_2 are synonymous terms, and if not, $sim_N()$ will be used to calculate the lexical similarity with N-gram distance [20]. If $sim(c_1, c_2) < 0.4$, similarity value among c_1 , c_2 and c_3 is set as 0; otherwise, we reduce W_1 by removing all the words that have already been matched with W_2 to get W'_1 . After that, c_1 , c_2 and c_3 's similarity value is determined by the $sim(W'_1, W_3)$. In particular, the threshold for $sim(W'_1, W_3)$ is set as 0.8, which is referred to the configurations in Oliveira et al. [7].

4. Compact evolutionary algorithm

This work presents a problem-specific CEA, and use it to solve the ternary compound ontology matching problem. CEA uses the binary encoding mechanism, i.e., Gray code, to encode the ternary compound correspondences in a solution. Supposing $|C_1|$, $|C_2|$ and $|C_3|$ are respectively the number of concepts in three ontologies, the length of a chromosome is $\log |C_{src}| \times (\log |C_{tgt1}| + \log |C_{tgt2}|)$. The i -th gene bit consists of two parts of information, one for the index of the first target concept and the other for the second. When decoding, the i -th source concept is mapped with these two target concepts.

A PA is used to represent a population and execute the evolving process. In particular, the length of a PA is equal to that of a chromosome, where each element inside is the probability of being one with respect to the corresponding gene bit. Here is an example of generating a new solution through PA $(0.3, 0.7, 0.6, 0.9)^T$. First, generate four random numbers, such as 0.4, 0.5, 0.8 and 0.1. Then compare the numbers with the elements in PA accordingly to determine the new generated individual's

Table 1. Description on nine biomedical ontologies.

Full Name	Acronym	Scale
Human phenotype ontology [22]	HP	10,593 classes
Mammalian phenotype ontology [23]	MP	10,730 classes
Neurobehavior ontology [24]	NBO	1,070 classes
<i>C. elegans</i> phenotype vocabulary [25]	WBP	2,185 classes
Phenotypic quality ontology [26]	PATO	2,444 classes
Cell ontology [27]	CL	5,901 classes
Foundational model of anatomy [28]	FMA	83,283 classes
Gene ontology [29]	GO	41,300 classes
Uber anatomy ontology [30]	UBERON	12,808 classes

gene values, e.g., since $0.4 > 0.3$, the first gene bit's value of the new solution is 0, and similarly, the remaining gene bits' values are 1, 0 and 1, respectively. In this way, the new solution we obtain is 0101. By repeating this procedure, we can obtain various individuals. In each generation, PA is updated to move towards the best solution found so far, which is implemented by increasing the elements' values in PA when the best solution's corresponding element's value is 1, and decreasing the values when 0. Given the $step = 0.1$, if the gene value of the elite is 1 (or 0), the corresponding element of PA will increase (or decrease) by st , and the updated PA is $(0.2, 0.8, 0.5, 1.0)^T$. When PA's elements are all approach 1 or 0, CEA converges, and thus, the step length affects the algorithm's searching performance. When the step length is large, the algorithm will converge quicker but tends to get stuck in the local optima; and when it is too small, the algorithm's learning ability is lower, which makes it converges slower. This work set the step length is $\sqrt{n} \log n$ by referring to Sudholt et al. [21]. Given the length of a chromosome (or PA) len , the step length for updating PA st and the maximum generation for CEA $maxGen$, the pseudo-code of CEA is shown in Algorithm 1.

5. Experiment

5.1. Experimental configuration

To test the performance of CEA, we use six testing cases that consists of nine ontologies, which are suggested by Oliveira et al. [7]. Table 1 shows a brief descriptions on these ontologies. We denote a testing case with the pattern "source ontology-the first target ontology-the second target ontology", e.g., "MP-CL-PATO".

The alignment's quality are evaluated with the traditional recall, precision and f-measure [31], which are defined as follows:

$$recall(A) = \frac{|A \cup RA|}{|RA|} \quad (5.1)$$

$$precision(A) = \frac{|A \cup RA|}{|A|} \quad (5.2)$$

$$f - measure(A) = \frac{2 \times recall \times precision}{recall + precision} \quad (5.3)$$

Algorithm 1 Compact evolutionary algorithm

```

*****Initialization*****
for  $i = 0; i < len; i ++$  do
     $PA[i] = 0.5;$ 
end for
generate a solution through PA and set it as  $sol_{elite};$ 
 $gen = 0;$ 
*****Evolving Process *****
while  $gen < maxGen$  do
    generate a solution  $sol_{new}$  through PA;
     $[winner, loser] = competition(sol_{new}, sol_{elite});$ 
    if  $winner == sol_{new}$  then
         $sol_{elite} = sol_{new};$ 
    end if
    *****Update PA *****
    for  $i = 0; i < len; i ++$  do
        if  $sol_{elite}[i] == 1$  then
             $PA[i] = PA[i] + st;$ 
        else
             $PA[i] = PA[i] - st;$ 
        end if
    end for
    for  $i = 0; i < len; i ++$  do
        if  $sol_{elite}[i] >= 1$  then
             $PA[i] = 1 - \frac{1}{len};$ 
        end if
        if  $sol_{elite}[i] <= 0$  then
             $PA[i] = \frac{1}{len};$ 
        end if
    end for
     $gen = gen + 1;$ 
end while

```

Table 2. Comparison with meta-heuristics based matching techniques in terms of alignment's quality. f and $stDev$ stand for f-measure and standard deviation, respectively.

Testing Case	EA f ($stDev$)	MA f ($stDev$)	DE f ($stDev$)	CEA f ($stDev$)
MP-CL-PATO	0.36 (0.02)	0.42 (0.01)	0.36 (0.02)	0.55 (0.01)
MP-GO-PATO	0.55 (0.01)	0.55 (0.02)	0.61 (0.01)	0.65 (0.02)
MP-NBO-PATO	0.41 (0.02)	0.50 (0.01)	0.47 (0.02)	0.58 (0.01)
MP-UBERON-PATO	0.47 (0.02)	0.64 (0.02)	0.64 (0.01)	0.69 (0.01)
WBP-GO-PATO	0.30 (0.03)	0.35 (0.01)	0.35 (0.02)	0.43 (0.01)
HP-FMA-PATO	0.35 (0.02)	0.40 (0.03)	0.23 (0.02)	0.47 (0.01)

Table 3. T-Test statistical analysis on the alignment's quality.

Testing Case	EA vs CEA	MA vs CEA	DE vs CEA
	t - value (p - value)	t - value (p - value)	t - value (p - value)
MP-CL-PATO	-46.54 (0.0068)	-50.34 (0.0063)	-46.54 (0.0068)
MP-GO-PATO	-24.49 (0.0129)	-19.36 (0.0164)	-9.79 (0.0324)
MP-NBO-PATO	-41.64 (0.0076)	-30.98 (0.0102)	-26.94 (0.0118)
MP-UBERON-PATO	-53.88 (0.0059)	-12.24 (0.0259)	-19.36 (0.0164)
WBP-GO-PATO	-22.51 (0.0141)	-30.98 (0.0102)	-19.59 (0.0162)
HP-FMA-PATO	-29.39 (0.0108)	-12.12 (0.0262)	-58.78 (0.0054)

where $|A|$ and $|RA|$ are respectively the cardinalities of the alignment A and reference alignment RA . In particular, recall and precision respectively measure the coverage of correct correspondences by referring to RA and A , and f-measure is their harmonic mean value. The experiment is carried out with an Intel Core i9-8950HK CPU @ 2.90GHz x 12 and 25GB allocated RAM.

5.2. Comparison on alignment's quality and matching efficiency

First, we compare CEA with EA based matching technique [32], Memetic algorithm (MA) based matching technique [33], Differential Evolution Algorithm (DE) based matching technique [34], whose configurations are referred to their literatures. Table 2 shows their mean f-measure and standard deviation of the alignments obtained by EA based matching techniques, and Table 3 presents the T-Test statistical analysis [35] of the values in Table 2. The results of all the meta-heuristics based matching techniques are the mean value of 30 independent runs.

In our experiment, a level of significance $\alpha = 0.05$ is chosen. Data analysis of Tables 2 and 3 depicts that CEA statistically outperforms other meta-heuristics based matching techniques on f-measure at 5% significance level. In addition, CEA does not need to tune any parameters except for the maximum generation 3000, which makes it more stable than other meta-heuristics based matching techniques.

Tables 4 compares CEA with two state-of-the-art ternary compound ontology matching techniques, i.e., AML based matching techniques with top-one ranked selector and top-two ranked selector [7], in terms of recall, precision and f-measure. As shown in Table 4, CEA's recall outperforms the other two competitors in all testing cases, which show that the compact evolving mechanism is able to effectively search in large searching space. Next, CEA's precision values also rank the first in all matching tasks,

Table 4. Comparison with state-of-the-art ternary compound ontology matching techniques.

Testing Case	AML _{top-one}	AML _{top-two}	CEA
Comparison in terms of recall			
MP-CL-PATO	0.24	0.53	0.58
MP-GO-PATO	0.60	0.61	0.62
MP-NBO-PATO	0.39	0.41	0.55
MP-UBERON-PATO	0.46	0.51	0.72
WBP-GO-PATO	0.10	0.13	0.48
HP-FMA-PATO	0.20	0.22	0.54
Average	0.33	0.40	0.58
Comparison in terms of precision			
MP-CL-PATO	0.24	0.34	0.53
MP-GO-PATO	0.62	0.41	0.69
MP-NBO-PATO	0.50	0.42	0.60
MP-UBERON-PATO	0.55	0.52	0.64
WBP-GO-PATO	0.11	0.11	0.37
HP-FMA-PATO	0.17	0.24	0.41
Average	0.36	0.34	0.54
Comparison in terms of f-measure			
MP-CL-PATO	0.24	0.42	0.55
MP-GO-PATO	0.61	0.49	0.65
MP-NBO-PATO	0.44	0.41	0.58
MP-UBERON-PATO	0.50	0.52	0.69
WBP-GO-PATO	0.10	0.12	0.43
HP-FMA-PATO	0.23	0.23	0.47
Average	0.35	0.36	0.56

which show the proposed similarity measure's effectiveness. Finally, the f-measure of CEA based matching technique show that it is able to effectively optimize the alignment's quality.

From Table 5, we can see that, in all testing cases, CEA dramatically improves EA's memory consumption and runtime per generation. Specifically, the highest improvement degree is on average by 82.95% and 80.93%, respectively. These gains are due to that CEA introduces the compact encoding mechanism, which is able to significantly save the memory consumption, and the simplified evolving mechanism, which is able to improve the algorithm's runtime per generation. To conclude, CEA is able to improve the quality of ternary compound ontology alignment, and significantly reduce the memory consumption and runtime per generation.

6. Conclusions

For the sake of the cognitive green computing, this work proposes a CEA to efficiently address the ternary compound ontology matching problem, which is able to bridge the semantic gap among three heterogeneous ontologies. In particular, we use the compact encoding and evolving mechanisms to

Table 5. Comparison of the memory (MegaByte) consumed per generation.

Testing Case	EA	MA	DE	CEA
Comparison of the memory (MegaByte) consumed per generation				
MP-CL-PATO	524.30	1,027.72	490.26	115.21
MP-GO-PATO	547.75	1,109.44	583.90	145.32
MP-NBO-PATO	384.20	710.32	272.46	94.41
MP-UBERON-PATO	924.32	1,420.97	918.16	382.54
WBP-GO-PATO	640.22	1,302.61	712.32	217.25
HP-FMA-PATO	1,350.21	2,558.98	1,254.13	433.57
Average	728.50	1,355.01	705.20	231.38
Comparison on the runtime (second) taken per generation				
MP-CL-PATO	4.88	6.82	3.64	0.82
MP-GO-PATO	4.21	7.04	3.22	0.86
MP-NBO-PATO	2.69	3.55	1.93	0.68
MP-UBERON-PATO	6.14	8.81	6.12	1.85
WBP-GO-PATO	6.68	7.12	5.64	1.73
HP-FMA-PATO	9.70	12.93	9.10	2.88
Average	5.72	7.71	4.94	1.47

reduce the algorithm's memory consumption and runtime when searching for the global optima. We construct an optimal model for the problem, and present a concept similarity measure to calculate a ternary compound correspondence's similarity value. The experimental results show that CEA-based matching technique is able to efficiently determine ternary compound ontology alignments.

When facing a very large-scale matching task, e.g., an ontology owns tens of thousands of concepts, a pre-screening method should be introduced. In the future, we will be interested in partitioning the ontologies into several semantic segments, which is of help to reduce the search space of CEA. In addition, a parallel matching mechanism would be of help to further improve the matching efficiency.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 11974373) and Natural Science Foundation of Fujian Province (No. 2020J01875).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, R. Singh, Cognitive computing, *Commun. ACM*, **54** (2011), 62–71.
2. P. Kurp, Green computing, *Commun. ACM*, **51** (2008), 11–13.

3. M. Chen, F. Herrera, K. Hwang, Cognitive computing: architecture, technologies and intelligent applications, *IEEE Access*, **6** (2018), 19774–19783.
4. S. Mirjalili, Genetic algorithm, in *Evolutionary Algorithms and Neural Networks*, Springer, 2019.
5. G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, *IEEE Trans. Evol. Comput.*, **3** (1999), 287–297.
6. N. Pour, A. Algergawy, R. Amini, D. Faria, I. Fundulaki, I. Harrow, et al., Results of the ontology alignment evaluation initiative 2020, *Proceedings of the 15th International Workshop on Ontology Matching (OM 2020)*, 2020.
7. D. Oliveira, C. Pesquita, Improving the interoperability of biomedical ontologies with compound alignments, *J. Biomed. Semantics*, **9** (2018), 1–13.
8. C. Pesquita, M. Cheatham, D. Faria, J. Barros, E. Santos, F. M. Couto, Building reference alignments for compound matching of multiple ontologies using obo cross-products, in *OM*, (2014), 172–173.
9. G. Acampora, U. Kaymak, V. Loia, A. Vitiello, Applying nsga-ii for solving the ontology alignment problem, in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, (2013), 1098–1103.
10. X. Xue, Y. Wang, Optimizing ontology alignments through a memetic algorithm using both matchmeasure and unanimous improvement ratio, *Artif. Intell.*, **223** (2015), 65–81.
11. M. Shamsfard, B. Helli, S. Babalou, Omega: Ontology matching enhanced by genetic algorithm. in *2016 Second International Conference on Web Research (ICWR)*, IEEE, (2016), 170–176.
12. Q. Lv, C. Jiang, H. Li, Solving ontology meta-matching problem through an evolutionary algorithm with approximate evaluation indicators and adaptive selection pressure, *IEEE Access*, **2020** (2020), 3046–3064.
13. N. Ferranti, S. S. R. F. Soares, J. F. de Souza, Metaheuristics-based ontology meta-matching approaches, *Expert Syst. Appl.*, **173** (2021), 114578.
14. G. Acampora, V. Loia, A. Vitiello, Enhancing ontology alignment through a memetic aggregation of similarity measures, *Inf. Sci.*, **250** (2013), 1–20.
15. X. Xue, Y. Wang, Using memetic algorithm for instance coreference resolution, *IEEE Trans. Knowl. Data Eng.*, **28** (2015), 580–591.
16. O. Bodenreider, The unified medical language system (umls): integrating biomedical terminology, *Nucleic Acids Res.*, **32** (2004), D267–D270.
17. X. Xue, J. S. Pan, An overview on evolutionary algorithm based ontology matching, *J. Inf. Hiding Multimed. Signal Proc.*, **9** (2018), 75–88.
18. X. Xue, J. Zhang, Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm, *Appl. Soft Comput.*, **106** (2021), 107343.
19. I. Mott, *Investigating semantic similarity for biomedical ontology alignment*. Ph.D thesis, Univer-sida De DeLisboa, 2017.

20. G. Kondrak, N-gram similarity and distance, in *International symposium on string processing and information retrieval*, Springer, 2005, 115–126.
21. D. Sudholt, C. Witt, Update strength in edas and aco: How to avoid genetic drift, in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016.
22. P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, S. Mundlos, The human phenotype ontology: a tool for annotating and analyzing human hereditary disease, *Am. J. Hum. Genet.*, **83** (2008), 610–615.
23. C. L. Smith, C. A. W. Goldsmith, J. T. Eppig, The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information, *Genome Biol.*, **6** (2005), R7.
24. G. V. Gkoutos, P. N. Schofield, R. Hoehndorf, The neurobehavior ontology: an ontology for annotation and integration of behavior and behavioral phenotypes, in *Int. Rev. Neurobiol.*, **103** (2012), 69–87.
25. G. Schindelman, J. S. Fernandes, C. A. Bastiani, K. Yook, P. W. Sternberg, Worm phenotype ontology: integrating phenotype data within and beyond the *c. elegans* community, *BMC Bioinformatics*, **12** (2011), 32.
26. G. V. Gkoutos, E. C. J. Green, A. M. Mallon, J. M. Hancock, D. Davidson, Using ontologies to describe mouse phenotypes, *Genome Biol.*, **6** (2005), R8.
27. J. Bard, S. Y. Rhee, M. Ashburner, An ontology for cell types, *Genome Biol.*, **6** (2005), R21.
28. C. Rosse, J. L. V. Mejino Jr, A reference ontology for biomedical informatics: the foundational model of anatomy, *J. Biomed. Inf.*, **6** (2003), 478–500.
29. Gene Ontology Consortium, The gene ontology (go) database and informatics resource, *Nucleic Acids Res.*, **32** (2004), D258–D261.
30. C. J. Mungall, C. Torniai, G. V. Gkoutos, S. E. Lewis, M. A. Haendel, Uberon, an integrative multi-species anatomy ontology, *Genome Biol.*, **13** (2012), R5.
31. C. J. Van Rijsbergen, Foundation of evaluation, *J. Doc.*, **30** (1974), 365–373.
32. J. M. V. Naya, M. M. Romero, J. P. Loureiro, C. R. Munteanu, A. P. Sierra, Improving ontology alignment through genetic algorithms, in *Soft computing methods for practical environment solutions: Techniques and studies*, (2010), 240–259.
33. G. Acampora, P. Avella, V. Loia, S. Salerno, A. Vitiello, Improving ontology alignment through memetic algorithms, in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, IEEE, (2011), 1783–1790.
34. X. Xue, J. Chen, Matching biomedical ontologies through compact differential evolution algorithm with compact adaption schemes on control parameters, *Neurocomputing*, **2020** (2020), 1–9.
35. L. Schmetterer, E. L. Lehmann, Testing statistical hypotheses, *Econometrica*, **30** (1962), 462–465.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)