Optimizing Hydrography Ontology Alignment Through Compact Particle Swarm Optimization Algorithm

Yifeng Wang¹, Hanguang Yao¹, Liangpeng Wan¹, Hua Li¹, Junjun Jiang², Yun Zhang², Fangmin Wu², Junfeng Chen³, Xingsi Xue^{4,5(✉)}, and Cai Dai⁶

¹ China Three Gorges Corporation, Beijing 100038, China
{wang_yifeng1,li_hua5}@ctgpc.com.cn,
{yao_hanguang,wan_liangpeng}@ctg.com.cn

² Hangzhou HuaNeng Engineering Safety Technology Co., Ltd.,
Hangzhou 311121, China
67692059@qq.com, 441079379@qq.com, 329537060@qq.com

³ College of IOT Engineering, Hohai University, Changzhou 213022, Jiangsu, China
chen-1997@163.com

⁴ Fujian Key Lab for Automotive Electronics and Electric Drive,
Fujian University of Technology, Fuzhou 350118, Fujian, China

⁵ Guangxi Key Laboratory of Automatic Detecting Technology and Instruments,
Guilin University of Electronic Technology, Guilin 541004, Guangxi, China
jack8375@gmail.com

⁶ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
cdai0320@snnu.edu.cn

Abstract. With the explosive growth in generating data in the hydrographical domain, many hydrography ontologies have been developed and maintained to describe hydrographical features and the relationships between them. However, the existing hydrography ontologies are developed with varying project perspectives and objectives, which inevitably results in the differences in terms of knowledge representation. Determining various relationships between two entities in different ontologies offers the opportunity to link hydrographical data for multiple purposes, though the research on this topic is in its infancy. Different from the traditional ontology alignment whose cardinality is 1:1, i.e. one source ontology entity is mapping with one target ontology entity and vice versa, and the relationship is the equivalence, matching hydrography ontologies is a more complex task, whose cardinality could be 1:1, 1:n or m:n and the relationships could be equivalence or subsumption. To efficiently optimize the ontology alignment, in this paper, a discrete optimal model is first constructed for the ontology matching problem, and then a Compact Particle Swarm Optimization algorithm (CPSO) based matching technique is proposed to efficiently solve it. CPSO utilizes the compact real-value encoding and decoding mechanism and the objective-decomposing strategy to approximate the PSO's evolving process, which can dramatically reduce PSO's memory consumption and runtime while at the same time ensure the solution's quality. The experiment exploits the Hydrography dataset in Complex track provided by the Ontology

Alignment Evaluation Initiative (OAEI) to test our proposal’s performance. The experimental results show that CPSO-based approach can effectively reduce PSO’s runtime and memory consumption, and determine high-quality hydrography ontology alignments.

Keywords: Hydrography ontology · Complex alignment · Compact Particle Swarm Optimization Algorithm

1 Introduction

To solve water features’s heterogeneity problem, many hydrography ontologies, such as the United States Geological Survey’s Surface Water Ontology (SWO) [9], Spanish National Geographic Institute (IGN)’s hydrOntology [14], the Hydro3 module from the University of Maine’s HydroGazetteer [13] and the Cree surface water ontology [15], have been developed and maintained to describe hydrographic features and the relationships between them. However, these ontologies are independently developed with varying project perspectives and objectives, and therefore, they might use the same terminology for different concepts, they may be at different levels of abstraction, they may not include all of the same concepts, and they may not even be in the same language [2]. For example, SWO describes surface water features from the perspective of the earth’s terrain, the water bodies and flows between them, while Cree surface water ontology presents the surface water features based on the people’s utility for transportation via canoe, which largely focuses on water bodies’ locations relative to one another. Matching ontologies aims at determining various relationships between two entities in different ontologies, which offers the opportunity to link hydrographical data for multiple purposes.

The traditional ontology matching techniques [3, 16–18] dedicate to find the simple alignment whose cardinality is 1:1, i.e. one source ontology entity is mapping with one target ontology entity and vice versa, and the relationship is the equivalence, e.g. a “Wetlands” in one ontology is equivalent to a “Swamp Or Marsh” in another ontology. However, matching hydrography ontologies is a more complex task, whose cardinality could be 1:1, 1:n or m:n and the relationships could be equivalence or subsumption [19]. Therefore, the traditional ontology matchers is not able to determine the high-quality ontology alignment, and the complex ontology matching problem is one of the challenges in the ontology matching domain [11], e.g. a “Watershed” in one ontology is a subclass of the union of the “Lake Or Pond” and “Swamp Or Marsh” in another. Being inspired by the success of Particle Swarm Optimization algorithm (PSO) based ontology matcher [1] in the ontology matching domain, we propose to use it to optimize the ontology alignment’s quality. To this end, we propose a Compact PSO (CPSO), which uses the compact real-value encoding and decoding mechanism and the objective-decomposing strategy to execute the evolving process.

The rest of the paper is organized as follows: Sect. 2 defines the hydrography ontology matching problem and proposes a hybrid similarity measure for calculating the similarity values between two hydrographical entities; Sect. 3 presents

the CPSO in details, which includes the compact encoding mechanism, the Probability Vector (PV) and the crossover operator; Sect. 4 presents the experimental configuration and results; finally, Sect. 5 draws the conclusion.

2 Preliminaries

2.1 Hydrography Ontology Matching Problem

To evaluate the quality of an ontology alignment and the effectiveness of a matching approach, it is necessary to determine whether all correct correspondences have been discovered (completeness) and whether all discovered correspondences are correct (soundness). Normally, the alignment is assessed in terms of two measures, commonly known as precision and recall. Precision (or soundness) measures the fraction of the selected correspondences that are actually correct. Recall (or completeness) measures the fraction of the number of correct mappings discovered against the total number of existing correct alignments. Maximum precision (no false positive) and maximum recall (no false negative) refer to the absence of type I and type II errors respectively. Although a precision of 1 means that all correspondences found are correct, it does not imply that all correct ones have been found. Analogously, a recall of 1 means that all the correct correspondences have been discovered, but it does not provide the information about the number of falsely identified ones. Therefore, precision and recall are often balanced against each other by the so-called f-measure, which is the uniformly weighted harmonic meaning of recall and precision. Since f-measure can better balance the precision and recall, it is the most popular indicator that is utilized to measure the quality of an ontology alignment. Given a Reference Alignment (RA) R , which is the golden ontology alignment provided by the expert, and an alignment A , recall, precision and f-measure are respectively defined as follows [12]:

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

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

$$f - measure = \frac{recall \times precision}{\alpha \times recall + (1 - \alpha) \times precision} \quad (3)$$

where α is the weight to tradeoff recall and precision. Although recall, precision and f-measure can reflect the quality of the resulting alignment, they require domain experts to provide the reference alignment in advance, which is generally unknown for difficult real-life matching problems. 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 [1], we utilize the following two metrics to approximate the f-measure:

$$f(A) = \frac{\phi(A) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|}}{\alpha \times \phi(A) + (1 - \alpha) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|}} \quad (4)$$

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 i th correspondence in A , $\phi(A)$ and $\frac{\sum_{i=1}^{|A|} \delta_i}{|A|}$ respectively approximate recall and precision.

On this basis, the optimal model for the ontology matching problem is defined as follows:

$$\begin{cases} \max & f(X) \\ \text{s.t.} & X = (x_1, x_2, \dots, x_{|O_1|})^T \\ & x_i \in \{1, 2, \dots, |O_2|\}, i = 1, 2, \dots, |O_1| \end{cases} \quad (5)$$

where O_1 and O_2 are two ontologies, $|O_1|$ and $|O_2|$ are respectively the entity number of O_1 and O_2 ; x_i means the i th entity correspondences, i.e. i th source entity is related with x_i target entities; and the objective function is maximize the alignment's $f()$.

2.2 Similarity Measure on Hydrographical Class

This work utilizes a hybrid similarity measure to calculate the similarity value between two hydrographical classes. Given two hydrographical classes, we first extract their label and comments from the corresponding ontologies. Before calculating their similarity value, we utilize the natural language processing technique and Babelnet Translate which covers 271 different languages and becomes an appropriate machine translation tool in cross-lingual ontology matching domain, to process their labels l_1 and l_2 and comments c_1 and c_2 . In particular, this process consists in the following successive steps:

- remove the numbers, punctuations and stop-words;
- split the strings into words;
- translate the words into English, and convert them into lower-case;
- lemmatizing and stemming the English words;

Then, four similarity values, i.e. $\text{sim}(l_1, l_2)$, $\text{sim}(c_1, c_2)$, $\text{sim}(l_1, c_2)$ and $\text{sim}(c_1, l_2)$, are calculated with soft TF-IDF [7] where two words are identical when they are the same literally or they are synonymous in the English Wordnet [6], and the maximum one is selected as two classes' similarity value.

3 Compact Particle Swarm Optimization Algorithm

In the next, CPSO for solving the ontology matching problem is presented in details, which approximates the population-based PSO's evolving process through a PV [5]. In the next, we first describe the encoding mechanism and PV, and then present the crossover operator and CPSO's pseudo-code.

3.1 Encoding and Decoding Mechanism

We utilize the real-number encoding mechanism. Since we need to encode an alignment in a solution whose kernel elements are two mapped concepts, we can simply make use of their indices in the ontologies. In an individual, each gene bit encode a positive integer that encoding the information that a source class with the same index as it are mapped with several target classes. In an alignment, if the i th source class is mapped with the target classes with index $j_1, j_2, \dots, j_n, j_n < O_2$ the i th gene bit's value will be $(2, 3, 5, \dots, primeNumber_{O_2}) \otimes (j_1, j_2, \dots, j_n, 0, 0, \dots) = 2^{j_1} \times 3^{j_2} \times 5^{j_3} \times \dots \times primeNumber_m^{j_n} \times primeNumber_{m+1}^0 \times \dots$. An example of the encoding mechanism is shown in the Fig. 1. As can be seen from the figure, the source concept “exocrine pancreas” with index 1 is mapped to target concepts “Exocrine_Pancreas” with index 1 and “Oropharynx_Epithelium” with index 4, and therefore, the first gene bit's value is $2^1 \times 3^4 \times 5^0 \times 7^0 = 162$.

Fig. 1. An example of encoding mechanism.

The decoding process is show in Algorithm 1.

In the obtain alignment, we utilize two thresholds, i.e. the upper threshold and lower threshold, to filter the class correspondences. When the class pair's similarity value is bigger than upper threshold, its correspondence's relationship is “equivalent”; when the class pair's similarity value is between upper threshold and lower threshold, its correspondence's relationship is “subsumption”; when the class pair's similarity value is lower than lower threshold, this correspondence will be removed. Next, for each object class, we utilize the approach proposed by D. Oliveira et al. [8] to further determine the relationships of “union” and “intersection” among its subsumed target classes.

3.2 Probability Vector

In CPSO, the Probability Vector (PV) [5] is used to represent the entire population. PV is defined as a $n \times 2$ matrix $[\mu^t, \delta^t]$ where t indicates the generation, μ

Algorithm 1. Decoding Mechanism**Input:** an individual *ind*, a positive prime number list *pNum***Output:** an alignment *align*

```

1: for int  $i = 1$ ;  $i \leq |O_1|$ ,  $i++$  do
2:   int  $count = 0$ ;
3:   while  $ind_i \bmod pNum_i = 0$  do
4:      $count = count + 1$ ;
5:   end while
6:    $align_i.add(count)$ 
7: end for
8: return align;

```

and δ are, respectively, vectors containing, for each gene value, mean and standard deviation values of a Gaussian Probability Distribution Function (PDF) truncated within the interval $[-1,1]$. In particular, for a gene bit indexed by i , a truncated Gaussian PDF characterized by a mean value μ_i , and a standard deviations δ_i is associated, and the formula of the PDF is as follows:

$$PDF(truncNorm(x)) = \frac{e^{-\frac{(x-\mu_i)^2}{2\delta_i^2}} \sqrt{\frac{2}{\pi}}}{\delta_i(erf(\frac{\mu_i+1}{\sqrt{2}\delta_i}) - erf(\frac{\mu_i-1}{\sqrt{2}\delta_i}))} \quad (6)$$

where *erf* is the error function [10]. From the PDF, the corresponding Cumulative Distribution Function (CDF) is constructed by means of Chebyshev polynomials [4], whose codomain of CDF is $[0,1]$. In order to sample the design variable x_i from PV a random number $rand(0, 1)$ is sampled from a uniform distribution. The inverse function of CDF, in correspondence of $rand(0, 1)$, is then calculated, whose round value is x_i .

In each generation, PV is updated to moved to the elite so that the newly generated solutions could be closer to the elite. In particular, the update rules for each element of μ and δ are respectively given as follows:

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(winner_i - loser_i) \quad (7)$$

$$(\delta_i^{t+1})^2 = (\delta_i^t)^2 + (\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2) \quad (8)$$

where N_p represents the population scale.

3.3 Crossover Operator

Given two individuals (called parents), the crossover operator generates one children, which are obtained by mixing the genes of the parents. Crossover is applied with a certain probability, a parameter of the PSO. Here, we use the widely used

Algorithm 2. Crossover Operator**Input:** two individuals ind_1 and ind_2 , crossover probability p_{cr} **Output:** a new individual ind_{new}

```

1: for int  $i = 1$ ;  $i \leq |O_1|$ ,  $i++$  do
2:   if  $rand(0, 1) < p_{cr}$  then
3:      $align_1 = decode(ind_{1,i})$ ;
4:      $align_2 = decode(ind_{2,i})$ ;
5:     int  $cutPoint =$  a random number in  $\{1, 2, \dots, ind.length\}$ ;
6:     for int  $j = 1$ ;  $j \leq cutPoint$ ,  $j++$  do
7:        $align_{new,i,j} = align_{1,i,j}$ ;
8:     end for
9:     for int  $j = cutPoint$ ;  $j \leq |O_1|$ ,  $j++$  do
10:       $align_{new,i,j} = align_{2,i,j}$ ;
11:    end for
12:   end if
13: end for
14:  $ind_{new} = encode(align_{new})$ ;
15: return  $ind_{new}$ ;

```

one-cut-point crossover operator. First, two individuals are decoded to obtain the corresponding alignments, and a cut position in two parent alignments is randomly determined and this position is a cut point which cuts each parent alignment into two parts: the left part and the right part. Then, the left part of one parent and the right part of the other are combined to form the children. For the sake of clarity, the pseudo-code of crossover operator is shown in Algorithm 2.

3.4 The Pseudo-code of Compact Particle Swarm Optimization Algorithm

The pseudo-code of CPSO is presented in Algorithm 3. CPSO first divides the problem into three sub-problems that respectively maximizes $f_{\alpha=0}$, $f_{\alpha=0.5}$ and $f_{\alpha=1}$. We initialize three PVs and local best individuals for three sub-problems, and then initialize the global best individual by selecting the one with best $f_{0.5}$ from them. In each generation, CPSO tries to solve each sub-problem by approximating PSO's position updating strategy, i.e. crossover an individual with the local best individual and global best individual to obtain a new individual, and then use the new one to update the local best individual and PV. After solving each sub-problem, we try to update the global individual with three new local best individuals.

Algorithm 3. Compact Particle Swarm Optimization Algorithm

Input: maximum generation $maxGen$, crossover rate p_{cr} , virtual population N_p

Output: global best individual ind_{elite}

```

1: ** Initialization **
2: generation  $t = 0$ ;
3: initialize  $PV_{\alpha=0}$ ,  $PV_{\alpha=0.5}$  and  $PV_{\alpha=1}$  by setting  $\mu_i^t = 0$  and  $\delta_i^t = 10$ ;
4: generate three individuals through  $PV_{\alpha=0}$ ,  $PV_{\alpha=0.5}$  and  $PV_{\alpha=1}$  to respectively
   initialize three local best individuals  $ind_{\alpha=0,elite}$ ,  $ind_{\alpha=0.5,elite}$  and  $ind_{\alpha=1,elite}$ ;
5: initialize  $ind_{elite} = opti\{ind_{\alpha=0,elite}, ind_{\alpha=0.5,elite}, ind_{\alpha=1,elite}\}$ ;
6: ** Evolving Process **
7: while  $t < maxGen$  do
8:   ** Updating  $ind_{\alpha=0,elite}$  **
9:   generate an individual  $ind_{\alpha=0,new}$  through  $PV_{\alpha=0}$ ;
10:   $ind_{\alpha=0,new} = crossover(ind_{\alpha=0,new}, ind_{\alpha=0,elite})$ ;
11:   $ind_{\alpha=0,new} = crossover(ind_{\alpha=0,new}, ind_{elite})$ ;
12:   $[winner, loser] = compete(ind_{\alpha=0,new}, ind_{\alpha=0,elite})$ ;
13:   $updatePV(winner, loser)$ ;
14:  if  $winner == ind_{\alpha=0,new}$  then
15:     $ind_{\alpha=0,elite} = ind_{\alpha=0,new}$ ;
16:  end if
17:   ** Updating  $ind_{\alpha=0.5,elite}$  **
18:   generate an individual  $ind_{\alpha=0.5,new}$  through  $PV_{\alpha=0.5}$ ;
19:    $ind_{\alpha=0.5,new} = crossover(ind_{\alpha=0.5,new}, ind_{\alpha=0.5,elite})$ ;
20:    $ind_{\alpha=0.5,new} = crossover(ind_{\alpha=0.5,new}, ind_{elite})$ ;
21:    $[winner, loser] = compete(ind_{\alpha=0.5,new}, ind_{\alpha=0.5,elite})$ ;
22:    $updatePV(winner, loser)$ ;
23:   if  $winner == ind_{\alpha=0.5,new}$  then
24:      $ind_{\alpha=0.5,elite} = ind_{\alpha=0.5,new}$ ;
25:   end if
26:   ** Updating  $ind_{\alpha=1,elite}$  **
27:   generate an individual  $ind_{\alpha=1,new}$  through  $PV_{\alpha=1}$ ;
28:    $ind_{\alpha=1,new} = crossover(ind_{\alpha=1,new}, ind_{\alpha=1,elite})$ ;
29:    $ind_{\alpha=1,new} = crossover(ind_{\alpha=1,new}, ind_{elite})$ ;
30:    $[winner, loser] = compete(ind_{\alpha=1,new}, ind_{\alpha=1,elite})$ ;
31:    $updatePV(winner, loser)$ ;
32:   if  $winner == ind_{\alpha=1,new}$  then
33:      $ind_{\alpha=1,elite} = ind_{\alpha=1,new}$ ;
34:   end if
35:   ** Updating  $ind_{elite}$  **
36:    $ind_{current} = opti\{ind_{\alpha=0,elite}, ind_{\alpha=0.5,elite}, ind_{\alpha=1,elite}\}$ ;
37:    $[winner, loser] = compete(ind_{current}, ind_{elite})$ ;
38:   if  $winner == ind_{current}$  then
39:      $ind_{elite} = ind_{current}$ ;
40:   end if
41:    $t = t + 1$ ;
42: end while
43: return  $ind_{elite}$ ;

```

4 Experiment

In order to test the performance of CPSO, the experiment exploits Hydrography dataset in Complex track provided by the Ontology Alignment Evaluation Initiative (OAEI). The hydrography dataset is composed of 4 source ontologies (Hydro3, hydrOntology_native, hydrOntology_translated and Cree) that each should be aligned to a single target Surface Water Ontology (SWO). The source ontologies vary in their similarity to the target ontology C Hydro3 is similar in both language and structure, hydrOntology_native and hydrOntology_translated are similar in structure but hydrOntology_translated is in Spanish rather than English, and Cree is very different in terms of both language and structure. We compare the alignment's quality among PSO-based matcher [1] and OAEI's participants in Table 1, and we also compare the runtime and memory consumption between PSO-based matcher and CPSO-based matcher in Table 2. PSO and CPSO's results are the mean values of thirty independent executions. PSO's configuration is referred to its literature, and CPSO uses the following parameters which represent a trade-off setting obtained in an empirical way to achieve the highest average alignment quality on all testing cases.

- Maximum generation: $maxGen = 3000$;
- Crossover probability: $p_{cr} = 0.6$;
- Virtual population: $N_p = 20$;
- Upper threshold: $threshold_{up} = 0.8$;
- Lower threshold: $threshold_{lo} = 0.5$;

As can be seen from Table 1, CPSO can determine better alignments than PSO-based matcher and OAEI's participants in terms of both precision and recall. In Table 2, CPSO can significantly improve the converging speed and reduce the memory consumption, which shows the effectiveness of the compact encoding mechanism and the compact evolutionary operators. CPSO combines the mechanisms of a classic PSO with a competitive learning, and the results achieved by CPSO are better and are attained faster than classic PSO. The gains in solution quality and algorithm's performance are achieved respectively due to CPSO's particular competitive learning, which is effective to lead the algorithm to determine the optimal solution, and the simplicity of CPSO, which does not require all the mechanisms of a PSO, rather the few steps in the algorithm are small and simple.

Table 1. Comparisons on the alignment’s quality.

Matcher	Precision	Recall	F-measure
Hydro3-SWO			
ABC	0.47	0.28	0.36
ALOD2Vec	1.00	0.13	0.23
DOME	1.00	0.13	0.23
FMapX	0.83	0.23	0.36
KEPLER	1.00	0.15	0.27
LogMap	1.00	0.15	0.27
POMAP++	1.00	0.13	0.23
XMap	0.83	0.23	0.36
PSO	0.79	0.21	0.33
CPSO	0.93	0.32	0.47
HydrOntology_translated-SWO			
ABC	0.17	0.05	0.09
DOME	0.26	0.02	0.04
LogMap	0.77	0.03	0.06
POMAP++	0.66	0.01	0.02
PSO	0.75	0.13	0.22
CPSO	0.82	0.23	0.35
HydrOntology_native-SWO			
ABC	0.25	0.005	0.01
PSO	0.21	0.13	0.16
CPSO	0.28	0.20	0.23
Cree-SWO			
ABC	0.83	0.13	0.24
ALOD2Vec	1.00	0.06	0.12
DOME	0.11	0.06	0.08
FMapX	1.00	0.04	0.08
KEPLER	1.00	0.04	0.08
PSO	0.80	0.12	0.20
CPSO	0.84	0.26	0.39

Table 2. Comparison on the runtime and memory consumption.

Testing case	PSO	CPSO	PSO	CPSO
	Runtime (second)	Runtime (second)	Memory (byte)	Memory(byte)
Hydro3-SWO	92	34	12,124,407	1,570,205
HydrOntology_translated-SWO	154	67	51,124,173	6,117,447
HydrOntology_native-SWO	142	64	49,143,746	5,219,357
Cree-SWO	64	28	34,109,561	2,749,235
Average	113	48	36,625,471	3,914,061

5 Conclusion

To efficiently optimize the hydrography ontology alignment’s quality, in this paper, a discrete optimal model is constructed for the ontology matching problem, and a CPSO-based ontology matching technique is presented to solve it. CPSO utilizes the compact real-value encoding mechanism to approximate PSO’s evolving process, which can dramatically reduce PSO’s runtime and memory consumption while at the same time ensure the solution’s quality. The experimental results show the effectiveness of our proposal.

Acknowledgements. This work is supported by the National Key Research and Development Project (No. 2016YFC0401607), the National Natural Science Foundation of China (No. 61503082), the Natural Science Foundation of Fujian Province (No. 2016J05145), the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (No. YQ20206), the Program for New Century Excellent Talents in Fujian Province University (No. GY-Z18155), the Program for Outstanding Young Scientific Researcher in Fujian Province University (No. GY-Z160149), the Scientific Research Foundation of Fujian University of Technology (Nos. GY-Z17162 and GY-Z15007), the Fundamental Research Funds for the Central Universities (No. 2019B22314), the National Key R&D Program of China (No. 2018YFC0407101), the Science and Technology Planning Project in Fuzhou City (No. 2019-G-40) and the Foreign Cooperation Project in Fujian Province (No. 2019I0019).

References

1. Bock, J., Hettenhausen, J.: Discrete particle swarm optimisation for ontology alignment. *Inf. Sci.* **192**, 152–173 (2012)
2. Cheatham, M., Varanka, D., Arauz, F., Zhou, L.: Alignment of surface water ontologies: a comparison of manual and automated approaches. *J. Geograph. Syst.* **22**, 1–23 (2019)
3. Chu, S.C., Xue, X., Pan, J.S., Wu, X.: Optimizing ontology alignment in vector space. *J. Internet Technol.* **21**(1), 15–22 (2020)
4. Cody, W.J.: Rational Chebyshev approximations for the error function. *Math. Comput.* **23**(107), 631–637 (1969)
5. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. *IEEE Trans. Evol. Comput.* **3**(4), 287–297 (1999)
6. Miller, G.A.: WordNet: a lexical database for english. *Commun. ACM* **38**(11), 39–41 (1995)

7. Ngo, D.H., Bellahsene, Z., Todorov, K.: Extended Tversky similarity for resolving terminological heterogeneities across ontologies. In: Meersman, R., et al. (eds.) OTM 2013. LNCS, vol. 8185, pp. 711–718. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41030-7_52
8. Oliveira, D., Pesquita, C.: Improving the interoperability of biomedical ontologies with compound alignments. *J. Biomed. Semant.* **9**(1), 1–13 (2018)
9. Sinha, G., et al.: An ontology design pattern for surface water features. In: Duckham, M., Pebesma, E., Stewart, K., Frank, A.U. (eds.) GIScience 2014. LNCS, vol. 8728, pp. 187–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11593-1_13
10. Temme, N.: Error functions, Dawsons and Fresnel integrals. In: NIST Handbook of Mathematical Functions, pp. 159–171 (2010)
11. Thiéblin, E., Haemmerlé, O., Hernandez, N., Trojahn, C.: Survey on complex ontology matching. In: Semantic Web (Preprint), pp. 1–39 (2019)
12. Van Rijsbergen, C.J.: Foundation of evaluation. *J. Doc.* **30**(4), 365–373 (1974)
13. Vijayasankaran, N.: Enhanced place name search using semantic gazetteers (2015)
14. Vilches-Blázquez, L., Ramos, J., López-Pellicer, F.J., Corcho, O., Nogueras-Iso, J.: An approach to comparing different ontologies in the context of hydrographical information. In: Popovich, V.V., Claramunt, C., Schrenk, M., Korolenko, K.V. (eds.) Information fusion and geographic information systems, pp. 193–207. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00304-2_13
15. Wellen, C.C., Sieber, R.E.: Toward an inclusive semantic interoperability: the case of cree hydrographic features. *Int. J. Geogr. Inf. Sci.* **27**(1), 168–191 (2013)
16. Xue, X., Chen, J., Yao, X.: Efficient user involvement in semiautomatic ontology matching. *IEEE Trans. Emerg. Top. Comput. Intell.* 1–11 (2018)
17. Xue, X., Wang, Y.: Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio. *Artif. Intell.* **223**, 65–81 (2015)
18. Xue, X., Wang, Y.: Using memetic algorithm for instance coreference resolution. *IEEE Trans. Knowl. Data Eng.* **28**(2), 580–591 (2015)
19. Zhou, L., Cheatham, M., Krisnadhi, A., Hitzler, P.: A complex alignment benchmark: GeoLink dataset. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11137, pp. 273–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_17