

Research Article

Solving Sensor Ontology Metamatching Problem with Compact Flower Pollination Algorithm

Wenwu Lian ¹, Lingling Fu ², Xishuan Niu ³, Junhong Feng ³,
and Jian-Hong Wang ⁴

¹*School of Physics and Telecommunication Engineering, Yulin Normal University, Yulin 537000, China*

²*Business School, Yulin Normal University, Yulin 537000, China*

³*School of Computer Science and Engineering, Yulin Normal University, Yulin 537000, China*

⁴*Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan*

Correspondence should be addressed to Lingling Fu; sxyfll@126.com and Xishuan Niu; cookes2000@163.com

Received 22 November 2021; Accepted 1 February 2022; Published 14 March 2022

Academic Editor: Xingsi Xue

Copyright © 2022 Wenwu Lian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To implement co-operation among applications on the Internet of Things (IoT), we need to describe the meaning of diverse sensor data with the sensor ontology. However, there exists a heterogeneity issue among different sensor ontologies, which hampers their communications. Sensor ontology matching is a feasible solution to this problem, which is able to map the identical ontology entity pairs. This work investigates the sensor ontology meta-matching problem, which indirectly optimizes the sensor ontology alignment's quality by tuning the weights to aggregate different ontology matchers. Due to the largescale entity and their complex semantic relationships, swarm intelligence (SI) based techniques are emerging as a popular approach to optimize the sensor ontology alignment. Inspired by the success of the flower pollination algorithm (FPA) in the IoT domain, this work further proposes a compact FPA (CFPA), which introduces the compact encoding mechanism to improve the algorithm's efficiency, and on this basis, the compact exploration and exploitation operators are proposed, and an adaptive switching probability is presented to trade-off these two searching strategies. The experiment uses the ontology alignment evaluation initiative (OAEI)'s benchmark and the real sensor ontologies to test CFPA's performance. The statistical comparisons show that CFPA significantly outperforms other state-of-the-art sensor ontology matching techniques.

1. Introduction

To implement the co-operations among applications on the Internet of Things (IoT) [1], we need to describe the meaning of diverse sensor data with the sensor ontology and express them in a machine-interpretable way. As the kernel technique of the semantic web [2], sensor ontologies, such as the CSIRO sensor ontology (CSIRO) [3], semantic sensor network ontology (SSN) [4], and MMI device ontology (MMI) [5], have been widely used in the IoT domain. Although they own lots of overlapped information, the heterogeneity issue also exists among them. For example, a sensor concept might be defined

with different terminologies or contexts, which hampers their communications. Sensor ontology matching is a feasible solution for this problem, which is able to map the identical ontology entity pairs [6].

To find the semantically identical sensor concepts, it is necessary to use the ontology matcher to measure the two concepts' similarity value. However, due to the limitations of the natural language processing domain, one single ontology matcher is not able to ensure its effectiveness in various heterogeneous contexts. The problem of how to comprehensively combine different ontology matchers to make their advantages and disadvantages complement each other, so as to enhance the final ontology alignment's quality, i.e., sensor

ontology meta-matching problem, has attracted many researchers' attention. Fernandez et al. [7] propose a fuzzy theory-based method of aggregating the ontology matchers. Later on, the global and local ontology alignment extracting technique [8] is presented to find the alignment from diverse alignments obtained by different ontology matchers. Their proposal is able to take each correspondence's preference into consideration, which improves the alignment's quality. The generative adversarial network (GAN) [9] is used to iteratively combine different matchers. The Siamese neural network (SNN) [10] is also used to train problem-specific ontology matchers on the basis of the existing ones. The semisupervised learning-based method [11] first requires the expert to provide the partial alignment and then use it to train the Bayesian probability model and determine the rest of the alignment. Multi-objective evolutionary algorithm (MOEA) [12], coevolutionary algorithm (CEA) [13], evolutionary tabu search algorithm (ETSA) [14], co-firefly algorithm (CFA) [15], and differential evolution algorithm (DEA) [16] are also proposed to optimize the sensor ontology alignment. Particle swarm optimization (PSO) [17] is also used to determine highquality sensor ontology alignment, which introduces the simulated annealing strategy (SA) to improve the algorithm's performance by trading off the exploitation and exploration. Inspired by the success of swarm intelligence (SI) in the ontology matching domain, this work investigates a newly emerging SI algorithm, i.e., flower pollination algorithm (FPA) [18], which has successfully been applied in wireless sensor network (WSN) [19] to address complex optimization problem. To overcome the population-based FPA's disadvantages, such as slow convergence speed [20], this work proposes a compact FPA (CFPA) and uses it to address the sensor ontology meta-matching problem. In particular, CFPA uses a probability vector (PV) [21] to present the whole population, and on this basis, it stimulates the original FPA's search process. Since it does not need to tune any parameters and significantly simplify the population-based FPA's evolving operations, which are helpful to improve FPA's searching efficiency. To be specific, the contributions made in this work are as follows: (1) we present the mathematical formula for the sensor ontology meta-matching problem; (2) we propose a problem-specific CFPA to efficiently address the problem, which uses the compact exploitation operator and compact exploration operator to mimic FPA's evolving process and an adaptive switching probability to trade-off CFPA's exploitation and exploration; and (3) we employ CFPA on ontology alignment evaluation initiative (OAEI)'s benchmark and the task of matching sensor ontologies. The results reveal that CFPA is able to efficiently solve the sensor ontology meta-matching problem.

The rest of the paper is organized as follows: the sensor ontology meta-matching problem is defined in Section 2; CFPA is presented in Section 3 in detail; the statistical experimental results are shown in Section 4; and finally, Section 5 draws the conclusions.

2. Sensor Ontology Meta-Matching Problem

The ontology matcher measures two sensor concepts' similarity values with a real number in $[0, 1]$. The higher the similarity value, the more possible it is that two concepts are identical. In general, there are three kinds of ontology matchers, which are based on a string, linguistic, and ontology structure [22]. An ontology matcher calculates the similarity value by taking into consideration only one or two linguistic features, and thus none of them is able to ensure the result's confidence when facing different heterogeneous contexts. Usually, it is necessary to comprehensively aggregate their results, which is of help to enhance the final value's confidence. For the convenience of this work, a sensor ontology O is defined as a 3-tuple (C, P, R) , where C , P , and R are respectively the sensor concept set, concept's property set, and the concepts' relationship set [23]. To overcome two sensor ontologies' heterogeneity issues, we need to find their entity mappings, and each correspondence is defined as 4-tuple $(e_1, e_2, \text{simValue}, \text{rel})$, where e_1 and e_2 are two ontologies' entities, simValue is their similarity value and rel is two entities' semantic relationship [24]. In this work, we aim at finding the identical sensor concepts from two ontologies, and thus, a correspondence's rel is an equivalence. Given two ontologies O_1 and O_2 , an ontology matcher is executed to determine their corresponding alignment, which is a set of entity correspondences [25]. In this work, the alignment is denoted by a matrix with real numbers in $[0, 1]$ as its elements, whose rows and columns are two entity sets, and its element is two corresponding entities' similarity value.

To combine these matchers, we assign the weights for their corresponding similarity matrices and then aggregate these matrices into the final one. The sensor ontology meta-matching problem investigates how to find an optimal weight set to determine a highquality alignment [26]. Here, we model the sensor ontology metamatching problem as a singleobjective optimization problem, which takes maximizing the alignment's quality as the objective. Given a sensor alignment, the more correspondences it has and the higher the mean similarity value of all the correspondences is, the better quality it owns. Based on this, we use the following two quality metrics on an alignment A :

$$f_1(A) = \sqrt{\frac{|A|}{\max\{|O_1|, |O_2|\}}} \in [0, 1], \quad (1)$$

$$f_2(A) = \sqrt{\frac{\sum \text{sim}_i}{|A|}} \in [0, 1],$$

where $|O_1|$, $|O_2|$, and $|A|$ are the number of two ontologies' entities and the correspondences in the alignment and sim_i is i -th correspondence's similarity value. After that, we calculate two metrics' harmony mean to comprehensively measure the alignment's quality, which is defined as follows:

$$f(A) = \frac{2 \times f_1(A) \times f_2(A)}{f_1(A) + f_2(A)} \in [0, 1]. \quad (2)$$

On this basis, the mathematical model of the problem is defined as follows:

$$\begin{cases} \max F(W), \\ \text{s.t. } W = (w_1, w_2, \dots)^T, \\ w_i \in [0, 1] \\ \sum w_i = 1, \end{cases} \quad (3)$$

where w_i is the i -th weight of the ontology matcher's corresponding matching matrix and $F(W)$ first uses W to aggregate all the matching matrices and then use the function $f()$ to calculate the final matrix's corresponding alignment's quality.

3. Compact Flower Pollination Algorithm

FPA is inspired by the pollination of natural flowers, and its evolving process consists of two distinct operators, i.e., global pollination and local pollination, whose formulas are defined in the following equations:

$$x_i^{t+1} = x_i^t + L(x_i^t - x^*), \quad (4)$$

$$x_i^{t+1} = x_i^t + \text{rand}(0, 1)(x_p^t - x_q^t), \quad (5)$$

where t is current generation, x_i^t is i -th pollen in t -th generation, x_p^t and x_q^t are two neighbor pollens, x^* is the best pollen found, and L is the step length that draws from Levy distribution [27]. FPA's exploration and exploitation are controlled by a switching probability $\text{sp} \in [0, 1]$. In each generation, for each pollen, FPA generates a random number in $[0, 1]$ and compares it with p to decide the operation on it, and after that, FPA tries to update the best pollen. Classic FPA suffers from low converging speed, and to overcome this drawback, this work proposes a CFPA, whose main components, i.e., the encoding mechanism and exploration and exploitation operators, which are presented in the following sections, respectively.

3.1. Encoding Mechanism. CFPA uses the gray code (GC) [28], a popular binary encoding mechanism, to encode pollen. To be specific, we use GC to encode the integers in $[0, 100]$, and when decoding, we normalize all the integers to obtain the corresponding weights. For example, given four ontology matchers and we need to encode four integers in a pollen, assuming 20, 20, 40, and 80, and the aggregating weights for the matching matrices are 0.125, 0.125, 0.25, and 0.5, respectively. In this work, we utilize one PV to describe a population, whose dimension is equal to the length of pollen, and its element is the probability of being 1 on the corresponding bit of the pollen. In the beginning, all PV's elements are initialized as 0.5, which is updated at the end of each generation according to the best pollen found.

Given a PV $(0.2, 0.4, 0.6, 0.8)^T$, generate four random numbers in $[0, 1]$, e.g., 0.1, 0.3, 0.5 and 0.9 since $0.1 > 0.2$, the

first bit of new pollen is 1; similarly, since $0.9 > 0.8$, the last bit of newly generated pollen is 0. When updating PV, if the value of the elite pollen is 1 (or 0), its corresponding PV's element will be increased (decreased), which can make the new pollen generated hereafter closer to the elite pollen. It is obvious that when all the probabilities are close to 1 or 0, the CFPA converges.

3.2. Exploration and Exploitation Operators. The exploitation operator aims at searching for particular pollen's neighboring places, while the exploration operator tries to search in an unexplored position. The pseudocode of the exploration and exploitation operators is shown in Algorithm 1 and Algorithm 2, respectively.

Here, we introduce the exponential crossover operator (EC) [29] to implement CFPA's exploration and exploitation operators. Given two pollens, EC randomly copies a certain number of sequential bits' values from the first one to the second one. Essentially, the obtained new pollen is generated by the turbulence on its parents, which is very exploitative. With respect to the exploration operator, we use EC to mix a newly generated pollen $\text{pollen}^{\text{new}}$ and the elite pollen $\text{pollen}^{\text{elite}}$, while in the exploitation, we first mix two newly generated pollens pollen^p and pollen^q to obtain the mediate pollen, then we mix it with the pollen $\text{pollen}^{\text{new}}$. Essentially, the exploration operator generates new pollen by moving it towards the global optima, and the exploitation operator moves the newly generated pollen to the direction determined by its neighbor pollens.

3.3. Pseudocode of Compact Flower Pollination Algorithm. The pseudocode of CFPA is presented in Algorithm 3. CFPA first initializes all the elements of PV as 0.5 and then uses them to initialize the elite pollen $\text{pollen}^{\text{elite}}$. In each generation, CFPA adaptively updates the switching probability sp and then uses it to decide whether to execute on exploration or exploitation. Here, sp is the probability of executing the exploitation operator. In the early phase, the algorithm mainly focuses on exploration, i.e., sp is large, while in the late phase, CFPA puts the emphasis on exploitation, i.e., sp is small. At the end of each iteration, CFPA tries to update the $\text{pollen}^{\text{elite}}$ and PV. Finally, when reaching the maximum iteration number $\text{max}T = 3000$, the algorithm terminates and returns $\text{pollen}^{\text{elite}}$. Here, we update PV with the step that is determined by the pollen's length, and how to adaptively set the optimal step length for updating PV is one of our future research directions.

4. Experimental Results and Analysis

4.1. Experimental Setup. The benchmark of the ontology alignment evaluation initiative (OAEI) [30] and the task of matching three real sensor ontologies are used to test CFPA's performance. In Tables 1–4, we compare CFPA with five state-of-the-art sensor ontology matching techniques, i.e., compact coevolutionary algorithm (CCEA) [13], compact evolutionary tabu search algorithm (CETSA) [14], compact co-firefly algorithm (CCFA) [15], compact differential

```

(1)  $pollen^{new} = generatePollen(PV)$ ;
(2)  $int\ num = round(random(0,1) \times pollen.length)$ ;
(3)  $int\ index = 0$ ;
(4) for ( $int\ i = 0; i < pollen.length, i = i + 1$ )
(5)   if ( $index + 1 > num$ )
(6)     break;
(7)   end if
(8)   if ( $(index + 1 > pollen.length)$ )
(9)      $num = 0$ ;
(10)  end if
(11)   $pollen_i^{new} = pollen_i^{elite}$ ;
(12) end for
(13) return  $pollen^{new}$ ;

```

ALGORITHM 1: Exploration operator.

```

(1)  $pollen^{new} = generatePollen(PV)$ ;
(2)  $pollen^p = generatePollen(PV)$ ;
(3)  $pollen^q = generatePollen(PV)$ ;
(4)  $int\ num = round(random(0,1) \times pollen.length)$ ;
(5)  $int\ index = 0$ ;
(6) for( $int\ i = 0; i < pollen.length, i = i + 1$ )
(7)   if ( $index + 1 > num$ )
(8)     break;
(9)   end if
(10)  if ( $(index + 1 > pollen.length)$ )
(11)     $num = 0$ ;
(12)  end if
(13)   $pollen_i^p = pollen_i^q$ ;
(14) end for
(15)  $int\ num = round(random(0,1) \times pollen.length)$ ;
(16)  $int\ index = 0$ ;
(17) for( $int\ i = 0; i < pollen.length, i = i + 1$ )
(18)   if ( $index + 1 > num$ )
(19)     break;
(20)   end if
(21)   if ( $(index + 1 > pollen.length)$ )
(22)      $num = 0$ ;
(23)   end if
(24)    $pollen_i^{new} = pollen_i^p$ ;
(25) end for
(26) return  $pollen^{new}$ ;

```

ALGORITHM 2: Exploitation operator.

evolution algorithm (CDEA) [16], and simulated annealing particle swarm optimization (SAPSO) [17], on all testing cases in terms of recall, precision, and f-measure, respectively.

Three kinds of ontology matchers used in this work are the N-gram distance [31] (string-based ontology matcher), wordnet-based distance [32] (linguistic-based matcher), and profile-based distance [33] (structure-based ontology matcher), and the configurations of SIs are referred to in their literature. The results shown in the tables are the average of thirty independent runs.

To fairly compare with other matching techniques, we use recall, precision, and f-measure [34] to evaluate the

obtained alignments. In Table 5, we briefly describe the testing cases used in the experiment, and in Tables 1–4, the testing cases 1XX, 2XX, and 3XX are the ones starting with the numbers 1, 2, and 3, respectively.

4.2. Statistical Experiment. We utilize the statistical testing method *T*-test [35] to compare different competitors' performances in terms of recall, precision, and f-measure, respectively. Tables 1 and 2 show the six SI-based sensor ontology matching techniques' mean recall, precision, and f-measure, and the corresponding standard deviation on all the testing cases, and Tables 3 and 4, respectively, present the *t* value and *p* value on recall, precision, and f-measure.

```

(1) **Initialization**
(2) generation  $t = 0$ ;
(3) set all elements in PV as 0.5;
(4)  $pollen^{elite} = generatePollen(PV)$ ;
(5) **Iteration**
(6) while  $t < \max T$  do
(7)    $sp = e^{-t/\max T}$ ;
(8)   ***UpdatePollen***
(9)   if  $random(0, 1) < sp$ 
(10)     $pollen^{new} = exploration$ ;
(11)   else
(12)     $pollen^{new} = exploration$ ;
(13)   end if
(14)    $[winner, loser] = compete(pollen^{new}, pollen^{elite})$ ;
(15)   if ( $winner = pollen^{new}$ )
(16)     $pollen^{elite} = pollen^{new}$ ;
(17)   end if
(18)   **** Update PV****
(19)   for ( $i = 0$ ;  $i < PV.length$ ;  $i = i + 1$ )
(20)    if  $pollen_i^{elite} = 1$  then
(21)      $PV_i = PV_i + (1/pollen.length)$ ;
(22)    else
(23)      $PV_i = PV_i - (1/pollen.length)$ ;
(24)    end if
(25)   end for
(26)    $t = t + 1$ ;
(27) end while
(28) return  $pollen^{elite}$ ;

```

ALGORITHM 3: Compact flower pollination algorithm.

TABLE 1: Comparison among swarm intelligence-based sensor ontology matching techniques in terms of alignment quality.

Testing Case	CCEA $f(r, p)$	CETSA $f(r, p)$	CCFA $f(r, p)$	CDEA $f(r, p)$	SAPSO $f(r, p)$	CFPA $f(r, p)$
1XX	1.00 (1.00, 1.00)	0.81 (0.72, 0.90)	0.95 (0.94, 0.95)	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
2XX	0.92 (0.91, 0.93)	0.72 (0.84, 0.61)	0.82 (0.79, 0.86)	0.91 (0.88, 0.94)	0.73 (0.72, 0.74)	0.94 (0.95, 0.94)
3XX	0.78 (0.81, 0.74)	0.42 (0.37, 0.48)	0.71 (0.67, 0.77)	0.89 (0.85, 0.94)	0.82 (0.85, 0.79)	0.92 (0.89, 0.96)
MMI-SSN	0.90 (0.86, 0.94)	0.92 (0.90, 0.95)	0.92 (0.90, 0.95)	0.94 (0.92, 0.95)	0.88 (0.90, 0.87)	0.95 (0.91, 0.98)
CSIRO-SSN	0.92(0.89, 0.96)	0.94 (0.94, 0.95)	0.94 (0.94, 0.94)	0.94 (0.94, 0.95)	0.90 (0.88, 0.93)	0.96 (0.95, 0.96)
MMI-CSIRO	0.86 (0.88, 0.91)	0.90 (0.87, 0.94)	0.90 (0.87, 0.94)	0.92 (0.90, 0.93)	0.90 (0.87, 0.94)	0.94 (0.92, 0.97)
Average	0.89 (0.88, 0.92)	0.78 (0.77, 0.80)	0.87 (0.85, 0.90)	0.93 (0.91, 0.95)	0.87 (0.87, 0.87)	0.95 (0.93, 0.96)

The symbols f , r , and p , respectively, stand for f-measure, recall, and precision.

TABLE 2: Comparison among swarm intelligence-based sensor ontology matching techniques in terms of standard deviation.

Testing Case	CCEA $f_d(r_d, p_d)$	CETSA $f_d(r_d, p_d)$	CCFA $f_d(r_d, p_d)$	CDEA $f_d(r_d, p_d)$	SAPSO $f_d(r_d, p_d)$	CFPA $f_d(r_d, p_d)$
1XX	0.01 (0.01, 0.01)	0.03 (0.02, 0.02)	0.01 (0.03, 0.02)	0.01 (0.01, 0.01)	0.01 (0.01, 0.01)	0.01 (0.01, 0.01)
2XX	0.01 (0.02, 0.01)	0.04 (0.02, 0.04)	0.02 (0.01, 0.03)	0.02 (0.02, 0.02)	0.03 (0.02, 0.03)	0.01 (0.01, 0.01)
3XX	0.02 (0.01, 0.03)	0.05 (0.02, 0.04)	0.02 (0.01, 0.03)	0.01 (0.02, 0.01)	0.02 (0.02, 0.01)	0.02 (0.01, 0.01)
MMI-SSN	0.01 (0.02, 0.02)	0.03 (0.03, 0.02)	0.03 (0.03, 0.02)	0.02 (0.03, 0.01)	0.04 (0.03, 0.02)	0.01 (0.01, 0.01)
CSIRO-SSN	0.03 (0.01, 0.01)	0.03 (0.02, 0.01)	0.03 (0.02, 0.02)	0.05 (0.03, 0.02)	0.03 (0.02, 0.02)	0.02 (0.01, 0.02)
MMI-CSIRO	0.03 (0.02, 0.01)	0.03 (0.01, 0.01)	0.02 (0.02, 0.03)	0.01 (0.01, 0.01)	0.04 (0.02, 0.04)	0.01 (0.01, 0.01)

The symbols f_d , r_d , and p_d , respectively, stand for the standard deviation of f-measure, recall, and precision.

As can be seen from Tables 1 and 2, CFPA's results are much better than those of other SI-based sensor ontology matching techniques. Thanks to the adaptive switching

probability, CFPA is able to better trade-off the algorithm's exploitation and exploration, which not only ensures the solution's quality but also the algorithm's stability. From

TABLE 3: T -test's t value.

Testing Case	(CCEA, CFPA) $f_t(r_t, p_t)$	(CETSA, CFPA) $f_t(r_t, p_t)$	(CCFA, CFPA) $f_t(r_t, p_t)$	(CDEA, CFPA) $f_t(r_t, p_t)$	(SAPSO, CFPA) $f_t(r_t, p_t)$
1XX	0.00 (0.00, 0.00)	-180 (-375, -134)	-106 (-56, -67)	0.00 (0.00, 0.00)	0.00 (0.00, 0.00)
2XX	-42 (-53, -21)	-160 (-147, -240)	-160 (-339, -75)	-40 (-93, 0.00)	-199 (-308, -189)
3XX	-148 (-169, -208)	-278 (-697, -349)	-222 (-466, -180)	-0.40 (-53, -42)	-106 (-53, -360)
MMI-SSN	-106 (-67, -53)	-28 (-9, -40)	-28 (-9, -40)	-13 (-9, -63)	-50 (-9, -147)
CSIRO-SSN	-33 (-127, 0.00)	-16 (-13, -13)	-16 (-13, -21)	-11 (-9, -10)	-49 (-93, -31)
MMI-CSIRO	-75 (-53, -127)	-37 (-106, -63)	-53 (-67, -28)	-42 (-42, -84)	-29 (-67, -21)

The symbols f_t , r_t , and p_t , respectively, stand for the t -value on f-measure, recall, and precision.

TABLE 4: T -test's p value.

Testing Case	(CCEA, CFPA) $f_p(r_p, p_p)$	(CETSA, CFPA) $f_p(r_p, p_p)$	(CCFA, CFPA) $f_p(r_p, p_p)$	(CDEA, CFPA) $f_p(r_p, p_p)$	(SAPSO, CFPA) $f_p(r_p, p_p)$
1XX	0.50 (0.50, 0.50)	0.001 (0.0008, 0.0023)	0.003 (0.005, 0.004)	0.50 (0.50, 0.50)	0.50 (0.50, 0.50)
2XX	0.007 (0.006, 0.015)	0.001 (0.002, 0.001)	0.001 (0.002, 0.001)	0.007 (0.003, 0.50)	0.001 (0.001, 0.001)
3XX	0.002 (0.001, 0.001)	0.001 (0.0004, 0.0009)	0.001 (0.0006, 0.001)	0.378 (0.006, 0.007)	0.003 (0.006, 0.0008)
MMI-SSN	0.003 (0.004, 0.006)	0.011 (0.035, 0.007)	0.011 (0.035, 0.007)	0.024 (0.035, 0.005)	0.006 (0.035, 0.002)
CSIRO-SSN	0.009 (0.002, 0.50)	0.019 (0.024, 0.024)	0.019 (0.024, 0.015)	0.028 (0.035, 0.031)	0.006 (0.003, 0.010)
MMI-CSIRO	0.004 (0.006, 0.002)	0.008 (0.003, 0.005)	0.006 (0.004, 0.011)	0.007 (0.007, 0.003)	0.010 (0.004, 0.015)

The symbols f_p , r_p , and p_p , respectively, stand for the p -value on f-measure, recall, and precision.

TABLE 5: Descriptions of the ontologies in the testing cases.

Testing case	Ontology	Scale
OAIE's benchmark	Bibliographic ontology	97 entities
Real sensor ontology	CSIRO sensor ontology (CSIRO)	33,205 entities
	Semantic sensor network ontology (SSN)	32,298 entities
	MMI device ontology (MMI)	24,034 entities

Tables 3 and 4, except those testing cases where the results of CFPA and other competitors are the same, our approach outperforms other SI-based sensor ontology matching techniques on a 5% significant level. Since CFPS does not need to tune any parameters, it is more stable than other SIs. In addition, CFPA's adaptive switching mechanism and two compact evolutionary operators are able to significantly improve the algorithm's performance, which makes it efficiently search for better solutions.

5. Conclusion

To support communication among IoT applications, it is necessary to describe the sensor data at a semantic level. Recently, sensor ontology has become a popular knowledge modeling technique in the IoT, which is able to provide semantic meanings for diverse sensor data. However, there exists the heterogeneity issue between different sensor ontologies, which hampers IoT applications' co-operation. Sensor ontology matching is a feasible solution to this problem, which aims to find identical sensor concepts at the semantic level. This work investigates a sensor ontology meta-matching problem, which aims to indirectly optimize the sensor ontology alignment's quality by tuning the weights to aggregate different ontology matchers. Inspired

by the success of FPA in the IoT domain, we further propose a CFPA to efficiently address the sensor ontology meta-matching problem. In particular, we introduce the compact encoding mechanism to improve the algorithm's searching efficiency and the adaptive switching parameter to trade-off the algorithm's exploitation and exploration. The experiment compares CFPA with five state-of-the-art sensor ontology matching techniques based on SIs, and the experimental results show that CFPA outperforms other SI-based sensor ontology matching techniques.

In the future, we will further improve CFPA to match the largescale sensor ontologies, especially at the instance level. We are also interested in further improving CFPA to match the specific ontologies in the biomedical domain and geographical domain. When dealing with largescale matching tasks, efficiency-improving strategies should be introduced, such as ontology partition and correspondence pruning. Also, the problem that how to choose the suitable background knowledge base to distinguish the complex entity correspondence also needs to be addressed.

Data Availability

The data used to support this study can be found in <http://oaei.ontologymatching.org>.

Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

Acknowledgments

This work was supported in part by the Improvement Project of Basic Ability for Young and Middle-aged Teachers in Guangxi Universities (No. 2017KY0535), the Science Research Foundation for High-level talents of Yulin Normal University (No. G2021ZK17), and the Construction of China-Ukraine Joint Carbon Black Research Center (I), (No. 2022HXZK01).

References

- [1] P. Pande and A. Padwalkar, "Internet of Things—a future of Internet: a survey," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, pp. 354–361, 2014.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [3] H. Neuhaus and M. Compton, "The semantic sensor network ontology," in *Proceedings of the AGILE Workshop on Challenges in Geospatial Data Harmonisation*, pp. 1–33, Hannover, Germany, 2009.
- [4] M. Compton, P. Barnaghi, L. Bermudez et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.
- [5] C. Rueda, N. Galbraith, and R. A. Morris, "The MMI device ontology: enabling sensor integration," in *Proceedings of the AGU Fall Meeting Abstracts*, pp. 1–35, San Francisco, CA, USA, 2010.
- [6] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2011.
- [7] S. Fernandez, I. Marsa-Maestre, J. Velasco, and B. Alarcos, "Ontology alignment architecture for semantic sensor web integration," *Sensors*, vol. 13, no. 9, pp. 12581–12604, 2013.
- [8] C. Touati, M. Benaissa, and Y. Lebbah, "An efficient model for extracting an optimal alignment with multiple cardinalities in ontology alignment," *International Journal of Metadata Semantics and Ontologies*, vol. 11, no. 2, pp. 71–81, 2016.
- [9] X. Xue and Q. Huang, "Generative adversarial learning for optimizing ontology alignment," *Expert Systems*, vol. 2022, pp. 1–12, 2022.
- [10] S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Class-balanced siamese neural networks," *Neurocomputing*, vol. 273, pp. 47–56, 2018.
- [11] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.
- [12] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2018.
- [13] X. Xue and J.-S. Pan, "A compact co-evolutionary algorithm for sensor ontology meta-matching," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 335–353, 2018.
- [14] D. Costa, "An evolutionary tabu search algorithm and the NHL scheduling problem," *INFOR: Information Systems and Operational Research*, vol. 33, no. 3, pp. 161–178, 1995.
- [15] X. S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm intelligence*, vol. 1, no. 1, pp. 36–50, 2013.
- [16] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.
- [17] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [18] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, pp. 240–249, Espoo, Finland, 2012.
- [19] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.
- [20] J. Pan, J. Zhuang, and H. Luo, "Multi-group flower pollination algorithm based on novel communication strategies," *Journal of Internet Technology*, vol. 22, no. 2, pp. 257–269, 2021.
- [21] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [22] J. Martinez-Gil and J. F. Aldana-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.
- [23] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: a literature review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015.
- [24] W. Hu and Y. Qu, "Falcon-AO: a practical ontology matching system," *Journal of Web Semantics*, vol. 6, no. 3, pp. 237–239, 2008.
- [25] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Journal of Web Semantics*, vol. 7, no. 3, pp. 235–251, 2009.
- [26] N. Ferranti N, S. S. R. F. Soares, and J. F. de Souza, "Meta-heuristics-based ontology meta-matching approaches," *Expert Systems with Applications*, vol. 173, pp. 1–15, 2021.
- [27] F. J. O'Reilly and R. Rueda, "A note on the fit for the Levy distribution," *Communications in Statistics-Theory and Methods*, vol. 27, no. 7, pp. 1811–1821, 1998.
- [28] R. Doran, "The gray code," *Journal of Universal Computer Science*, vol. 13, no. 11, pp. 1573–1597, 2007.
- [29] S.-Z. Zhao and P. N. Suganthan, "Empirical investigations into the exponential crossover of differential evolutions," *Swarm and Evolutionary Computation*, vol. 9, pp. 27–36, 2013.
- [30] M. Achichi, M. Cheatham, and Z. Dragisic, "Results of the ontology alignment evaluation initiative 2016," in *Proceedings of the 11th International Workshop on Ontology Matching Co-Located with the 15th International Semantic Web Conference (ISWC 2016)*, pp. 73–129, Kobe, Japan, 2016.
- [31] G. Kondrak, "N-gram similarity and distance," *String Processing and Information Retrieval*, Springer, Berlin, Germany, pp. 115–126, 2005.
- [32] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [33] G. Acampora, V. Loia, and A. Vitiello, "Enhancing ontology alignment through a memetic aggregation of similarity measures," *Information Sciences*, vol. 250, pp. 1–20, 2013.
- [34] G. Hripcsak and A. Rothschild, "Agreement, the f-measure, and reliability in information retrieval," *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 2005.
- [35] D. Semenic, "Tests and measurements: the T-test," *National Strength & Conditioning Association Journal*, vol. 12, no. 1, pp. 36–37, 1990.