

Research Article

Semantic Integration of Sensor Knowledge on Artificial Internet of Things

Yikun Huang,¹ Xingsi Xue^{ID},^{2,3} and Chao Jiang^{ID}⁴

¹Department of Information Technology, Concord University College Fujian Normal University, Fuzhou/350118, China

²Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou/350118, China

³Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin/541004, China

⁴Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou/350118, China

Correspondence should be addressed to Xingsi Xue; jack8375@gmail.com

Received 25 May 2020; Revised 27 June 2020; Accepted 2 July 2020; Published 25 July 2020

Academic Editor: Pei-Wei Tsai

Copyright © 2020 Yikun Huang 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.

Artificial Internet of Things (AIoT) integrates Artificial Intelligence (AI) with the Internet of Things (IoT) to create the sensor network that can communicate and process data. To implement the communications and co-operations among intelligent systems on AIoT, it is necessary to annotate sensor data with the semantic meanings to overcome heterogeneity problem among different sensors, which requires the utilization of sensor ontology. Sensor ontology formally models the knowledge on AIoT by defining the concepts, the properties describing a concept, and the relationships between two concepts. Due to human's subjectivity, a concept in different sensor ontologies could be defined with different terminologies and contexts, yielding the ontology heterogeneity problem. Thus, before using these ontologies, it is necessary to integrate their knowledge by finding the correspondences between their concepts, i.e., the so-called ontology matching. In this work, a novel sensor ontology matching framework is proposed, which aggregates three kinds of Concept Similarity Measures (CSMs) and an alignment extraction approach to determine the sensor ontology alignment. To ensure the quality of the alignments, we further propose a compact Particle Swarm Optimization algorithm (cPSO) to optimize the aggregating weights for the CSMs and a threshold for filtering the alignment. The experiment utilizes the Ontology Alignment Evaluation Initiative (OAEI)'s conference track and two pairs of real sensor ontologies to test cPSO's performance. The experimental results show that the quality of the alignments obtained by cPSO statistically outperforms other state-of-the-art sensor ontology matching techniques.

1. Introduction

Internet of Things (IoT) [1] consists of interconnected things with built-in sensors, and Artificial IoT (AIoT) [2] further integrates Artificial Intelligence (AI) with IoT to create the sensor network that can communicate and process data. To implement the communications and co-operations among intelligent systems on AIoT, it is necessary to annotate sensor data with the semantic meanings to overcome heterogeneity problem among different sensors, which requires the utilization of sensor ontology [3]. Sensor ontology formally models the knowledge on AIoT by defining the concepts, the properties describing a concept, and the relationships between two concepts. Since sensor ontologies are regarded as the solution

to data heterogeneity on AIoT, and in recent years, many sensor ontologies [4] have been developed. However, due to human's subjectivity, the overlapping information in these ontologies could be defined with different terminologies and contexts, yielding the ontology heterogeneity problem. Therefore, before using them, it is necessary to integrate their knowledge by finding the correspondences between their concepts. Ontology matching can bring sensor ontologies into mutual agreement by automatic determining identical concept correspondences (i.e., ontology alignment), which is regarded as an effective technique to address the ontology heterogeneity problem.

Due to high computational complexity in the matching process, the Swarm Intelligence algorithm (SI) has become

a popular methodology for integrating heterogeneous ontologies [5–9]. Martinez-Gil and Montes [10] propose the Genetics for Ontology Alignments (GOAL), which first generates a similarity matrix for each similarity measure, and then uses the Genetic Algorithm (GA) to optimize the weights for aggregating these matrices. Aggregating weights determined by GOAL can be reused to match the ontologies with similar heterogeneous features. Ginsca and Iftene [11] not only optimize the parameters in the matching process but also the threshold in the alignment filtering process. Acampora et al. [12] try to improve GA’s converging speed as well as the solution’s quality by introducing a local search strategy. Xue and Wang [13] propose a new metric to approximately measure the alignment’s f-measure [14], and on this basis, utilize the hybrid GA to execute the instance-level matching in the Linked Open Data cloud (LOD). More recently, He et al. [15] propose an Artificial Bee Colony algorithm (ABC) based matching technique to aggregate different similarity measures, which can improve the alignment’s quality. These SI-based matching techniques need to first store the similarity matrices determined by the similarity measures, which sharply increase the computational complexity. To this end, Genetic Algorithm based Ontology Matching (GAOM) [16] models the ontology matching as a bipartite graph matching process and tries to use GA to directly determine the alignment with high quality. Since the instance information can effectively improve the alignment’s precision value, Alves et al. [17] first propose an instance-based similarity measure and then utilize a hybrid GA to determine the optimal mappings. MapPSO [18] models the ontology matching as a bipartite graph matching problem, and it proposes to use the Particle Swarm Optimization algorithm (PSO) [19] to address it. MapPSO utilizes the statistical information of the alignment to approximately evaluate its quality and guide the algorithm’s search direction, which can automatically determine high-quality alignments. For dynamic applications on SSW, it is necessary to integrate the sensor ontologies online, and thus, besides the quality of the alignments, the matching efficiency is also of prime importance. Being inspired by the success of compact SI in various applications [20–23], this work proposes a compact PSO (cPSO) to integrate the sensor knowledge in AIoT. Our proposal uses a probabilistic representation of the population to execute the optimizing process, which simulates the population behaviour as it extensively explores the decision space at the beginning of the optimization process and progressively focuses the search on the most promising genotypes and narrows the search radius. Thus, a run of cPSO requires much more limited memory consumption comparing to the standard PSO. In particular, we formally

define the sensor ontology matching problem and propose a problem-specific cPSO to effectively address the problem and integrate the sensor knowledge inside.

The rest of the paper is organized as follows: Section 2 presents the concept similarity measures and the mathematical model of sensor ontology matching problem; Section 3 gives the details of cPSO; Section 4 shows the experimental results; and finally, Section 5 draws the conclusions.

2. Preliminaries

2.1. Concept Similarity Measure. Concept Similarity Measure (CMS) is a function that takes as input two concepts’ information, and output a real number in $[0, 1]$ which represents their similarity value. In general, there are three kinds of CMS, i.e., string-based CMS, linguistic-based CMS, and structure-based CMS. In particular, string-based CMS takes as input two concepts’ labels and compares their syntax information, linguistic-based CMS also compares two concepts’ labels but it uses the external digital dictionary such as Wordnet [24] to calculate their similarity value, and structure-based CMS calculate the similarity value of two concepts based on their direct super-concepts and subconcepts.

Given two concepts c_1 and c_2 , we first remove the meaningless words (such as the stop word) from their labels and convert them into two token sets T_1 and T_2 , then the string-based similarity value is calculated as follows:

$$\text{sim}_s(c_1, c_2) = \min \left\{ \frac{|T_1 \cap T_2|}{|T_1|}, \frac{|T_1 \cap T_2|}{|T_2|} \right\}, \quad (1)$$

where $|T_1|$ and $|T_2|$ are, respectively, the cardinalities of $|T_1|$ and $|T_2|$. The first ratio indicates the overlap fraction of $|T_1|$ with respect to $|T_2|$, the second one indicates $|T_2|$ with respect to $|T_1|$, and the minimum value is selected as their string-based similarity value. The linguistic-based similarity value is defined as follows:

$$\text{sim}_l(c_1, c_2) = \frac{\sum_{i=1}^{|T_1|} \max \left\{ \text{sim}'(T_{1,i}, T_{2,j}) \right\}_{j=1}^{|T_2|}}{|T_1|}, \quad (2)$$

where $T_{1,i}$ denotes the i th token in $|T_1|$ and $T_{2,j}$ is the j th token in $|T_2|$, $\text{sim}'(T_{1,i}, T_{2,j}) = 1$ if they are synonymous in Wordnet, otherwise 0. Finally, supposing superC1 and superC2 are, respectively, the super-concept set of c_1 and c_2 , subC1 and subC2 are, respectively, the direct subconcept set of c_1 and c_2 , the structure similarity value is defined as follows:

$$\text{sim}_u(c_1, c_2) = \frac{\sum_{i=1}^{|\text{superC}_1|} \max \left\{ \text{sim}_l(\text{superC}_{1,i}, \text{superC}_{2,j}) \right\}_{j=1}^{|\text{superC}_2|} + \sum_{i=1}^{|\text{subC}_1|} \max \left\{ \text{sim}_l(\text{subC}_{1,i}, \text{subC}_{2,j}) \right\}_{j=1}^{|\text{subC}_2|}}{|\text{superC}_1| + |\text{subC}_1|}, \quad (3)$$

where $\text{super}C_{1,i}$ denotes the i th super-class of c_1 and $\text{super}C_{2,j}$ the j th super-class of c_2 , $\text{sub}C_{1,i}$ is the i th direct subclass of c_1 and $\text{sub}C_{2,j}$ the j th subclass of c_2 .

Since none of the CSMs can ensure the effectiveness in all context, i.e., distinguishing all the heterogeneous concepts, usually they are combined together to enhance the result's confidence. Due to its flexibility, the weighted average strategy becomes a popular way of aggregating CSMs, which is defined as follows:

$$\text{sim}(c_1, c_2) = w_1 \times \text{sim}_s(c_1, c_2) + w_2 \times \text{sim}_l(c_1, c_2) + w_3 \times \text{sim}_u(c_1, c_2), \quad (4)$$

where $w_i \in [0, 1]$, $i = 1, 2, 3$, $\sum w_i = 1$.

2.2. Alignment Extraction. Each aggregating weight set corresponds to a unique aggregated CSM, which can be further used to construct a similarity matrix M whose element m_{ij} is the similarity value between the i th concept of one ontology and the j th concept of the other. On this basis, we can extract an alignment with the cardinality 1:1 (one concept from the source ontology is only mapped with one concept from the target ontology and vice versa) from M according to the following steps: (1) sort all the similarity values in M in descending order; (2) output the element with the largest value, say m_{ij} , as a concept correspondence (c_i, c_j, m_{ij}) in the extracted alignment; (3) replace the elements that are in the same row or column with m_{ij} as 0; (4) repeat the steps (1) to (3) until all the elements in M are 0. Figure 1 shows an example of the extracting process, where O_1 and O_2 are two ontologies, $c_{i,j}$ denotes the i th ontology's j th concept. Finally, six correspondences are extracted: $(c_{1,1}, c_{2,1}, 0.88)$, $(c_{1,2}, c_{2,2}, 0.88)$, $(c_{1,3}, c_{2,3}, 0.6)$, $(c_{1,5}, c_{2,5}, 0.6)$, $(c_{1,1}, c_{2,1}, 0.88)$, $(c_{1,4}, c_{2,4}, 0.08)$. With respect to the last correspondence, since its similarity value is low, which is regarded as unauthentic, the final alignment consists of top five correspondences.

2.3. Sensor Ontology Matching Problem. Since the quality of an alignment is directly proportional to the mean similarity value of all the correspondences found and the cardinality of the alignment, we utilize the following equation to calculate an alignment A 's quality:

$$f(A) = \sqrt{\frac{|A|}{\max\{|O_1|, |O_2|\}} \times \frac{\text{sim}_i}{|A|}}, \quad (5)$$

where $|O_1|$, $|O_2|$, and $|A|$ are, respectively, the cardinalities of two ontologies O_1 , O_2 , and A , sim_i is the i th correspondence's similarity value.

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

$$\begin{cases} \min F(W) \\ \text{s.t. } W = (w_1, w_2, \dots)^T \\ w_i \in [0, 1] \\ \sum w_i = 1 \end{cases} \quad (6)$$

where w_i , $i = 1, 2, \dots$ represents the i th similarity measure's aggregating weight, and $F(W)$ calculates the aggregating weight set W 's corresponding alignment's quality.

3. Compact Particle Swarm Optimization Algorithm

PSO is inspired by the behaviour of birds, where each bird (particle) has the memory of the best-visited position and moves to a leading bird (elite particle) with some degree of randomization. This procedure can be described with the following update sequence, for the i th particle in generation t :

$$v_i^{t+1} = \Phi_1 v_i^t + \Phi_2 (\text{particle}_{i-lb}^t - \text{particle}_i^t) + \Phi_3 (\text{particle}_{i-gb}^t - \text{particle}_i^t), \quad (7)$$

$$\text{particle}_i^{t+1} = \text{particle}_i^t + v_i^{t+1}, \quad (8)$$

where v_i^t is the velocity, i.e., a perturbation vector, particle_i^t is the i th particle's position in current generation and particle_{i-lb}^t is its best position visited in the history, particle_{i-gb}^t is the best position found by all the particles, and Φ_i , $i = 1, 2, 3$, is the weight vector. Eq. (7) and Eq. (8) indicate that PSO update each particle by exchanging its gene values with both local best particle and global best particle to find a better position. Clearly, the original PSO is a population-based SI, and in this work, we further propose a compact version of PSO to improve the algorithm's performance.

3.1. Encoding and Decoding Mechanism. This work uses a binary encoding mechanism, i.e., Gray code, and each particle's gene values can be divided into two parts, one stands for the weight set for aggregating the similarity measures and the other for the similarity threshold for filtering the correspondences with low similarity values. Concerning the characteristics of the weights in Section 2.1, we normalize them when decoding. We utilize one Probability Vectors (PVs) to represent a population, whose element number is equal to the length of a particle. Each PV's element represents the probability of being 1 corresponding to each gene bit of a particle. We can use PV to generate various binary particles through its probability in each dimension, and when each dimension value is closed to 1 or 0, the algorithm is about to converge. In addition, PV should be updated in each generation to move toward the elite, which can make the new particles generated in the next are more closed to the elite.

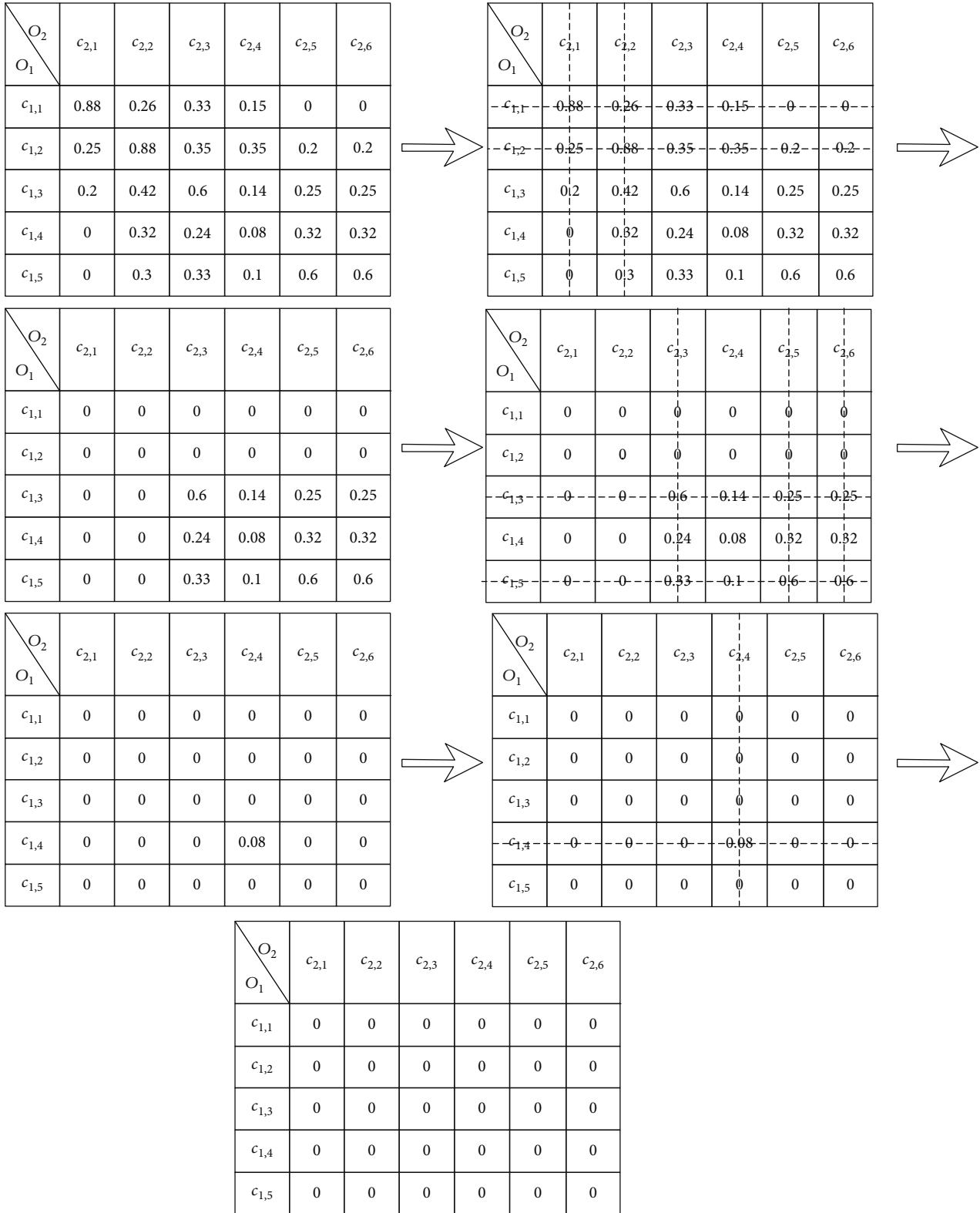


FIGURE 1: An example of extracting an alignment through similarity matrix.

Figure 2 shows an example of generating a particle through PV. Given a PV $(0.1, 0.3, 0.5, 0.9)^T$, generate four random numbers in $[0, 1]$, e.g., 0.2, 0.4, 0.6, and 0.1, and we can determine a new particle by comparing them with PV's

elements accordingly. To be specific, since $0.2 > 0.1$, $0.4 > 0.3$, $0.6 > 0.5$, and $0.1 < 0.9$, the newly generated particle is 0001. In each generation, PV's elements are updated according to the best particle found. If the bit value of the elite particle is

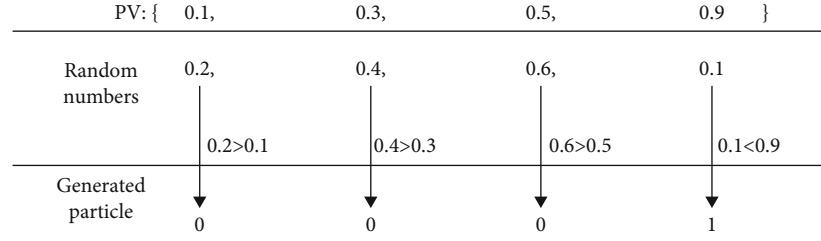


FIGURE 2: An example of generating a particle through PV.

1 (or 0), the corresponding PV's element will be increased (decreased) by step, which can make the newly generated particle more closer to the elite particle. For example, given a PV (0.1, 0.3, 0.5, 0.9)^T, an elite particle 1110 and the step length step = 0.1, since the first-bit value of the elite particle is 1, accordingly, we update the first element of PV by step, which makes the first-bit value of newly generated particle is more likely to be 1 (the same with elite particle's first-bit value). Therefore, after updating all elements of PV, the newly generated particles would be closer to the elite particle in terms of each bit value. When all elements of PV are 1 or 0, the newly generated particles will be the same and the algorithm converges.

3.2. Crossover Operator. Given two particles $particle_1$ and $particle_2$, the crossover operator generates one offspring by exchanging their gene values. In this work, we generate the offspring particle $particle_{off}$ by copying a sequential gene fragment of from $particle_1$ into the corresponding gene bits of $particle_2$, so that $particle_{off}$ inherits a sequential gene fragment from both $particle_1$ and $particle_2$. For the sake of clarity, the pseudo-code of the crossover operator is shown in Algorithm 1.

4. The Pseudo-Code of Compact Particle Swarm Optimization Algorithm

The pseudo-code of cPSO is presented in Algorithm 2. cPSO first initializes the probability vector PV by setting all the elements as 0.5, which is then used to initialize the local best particle $particle_{lb}$ and global best particle $particle_{gb}$. In each generation, cPSO first tries to update the $particle_{lb}$ through the crossover between it and a new particle, and then the $particle_{gb}$ by exchanging its gene values with $particle_{lb}$. Finally, cPSO updates PV according to $particle_{gb}$ to move the new generated particle towards it.

5. Experimental Results and Analysis

5.1. Experimental Setup. In the experiment, we use the Ontology Alignment Evaluation Initiative (OAEI)'s Conference track "<http://http://oaei.ontologymatching.org/2019/conference/index.html>" and two pairs of real sensor ontologies to test cPSO's performance. The experiment compares cPSO with four state-of-the-art sensor ontology matching techniques, i.e., ASMOV [25], CODI [26], SOBOM [27], and

```

 $particle_{off} = particle_2$ 
 $position = \text{round}(\text{rand}(0, \text{particle}_{off}.\text{length}))$ 
while  $\text{rand}(0, 1) < P_{cr}$  do
     $particle_{off,i} = particle_{1,i}$ ;
     $num = num + 1$ ;
     $position = (position + 1) \bmod \text{particle}_{off}.\text{length}$ ;
end while
return  $particle_{off}$ 

```

ALGORITHM 1. Crossover operator.

FuzzyAlign [28], on all testing cases in terms of f -measure. We empirically set cPSO's crossover probability as 0.8 and maximum generation as 3000, and cPSO's results in the tables are the average of thirty independent runs. In Table 1, we briefly describe the ontologies in these testing cases.

5.2. Statistical Comparison. We utilize two popular statistical testing methods, i.e., Friedman's Test (FT) [29] and Holm's Test (HT) [30], to compare different competitors' performance. In particular, FT aims at checking whether there are differences among the competitors, and HT is further used to find whether one competitor statistically outperforms others. First, we need to reject HT's null-hypothesis that all competitors' performances are the same. To this end, the computed value X_r^2 must be equal to or greater than the tabled critical chi-square value at the specified level of significance $\alpha = 0.05$. In this work, since we are comparing 5 matchers, the critical value for 4 degrees of freedom $X_{0.05}^2$ is 9.488.

In Table 2, since $X_r^2 = 85.09$, which is greater than 9.488, and therefore, the null hypothesis is rejected. Then, HT is further carried out. Since cPSO ranks with the lowest value, it is set as a control matcher that will be compared with others. HT's z value is the testing statistic for comparing the i th and j th competitors, which is used for finding the p value that is the corresponding probability from the table of the normal distribution. The p value is then compared with $\alpha/k - i$, and according to Table 3, we can state that cPSO statistically outperforms other competitors on f -measure at 5% significance level.

```

1: **Initialization **
2: set generation  $t = 0$  ;
3: set all elements in  $PV$  as 0.5;
4: generate one particle through  $PV$  to initialize the local best particle  $particle_{lb}$  and global best particle  $particle_{gb}$  ;
5: **Evolution **
6: while  $t < maxGen$  do
7:   generate a new particle  $particle_{new}$  through  $PV$  ;
8:   *** Update Particle ***
9:    $particle_{new} = crossover(particle_{new}, particle_{lb})$ ;
10:   $[winner, loser] = compete(particle_{new}, particle_{lb})$  ;
11:  if winner ==  $particle_{new}$  then
12:     $particle_{lb} = particle_{new}$  ;
13:  end if
14:   $particle_{new} = crossover(particle_{lb}, particle_{gb})$  ;
15:   $[winner, loser] = compete(particle_{new}, particle_{gb})$  ;
16:  if winner ==  $particle_{new}$  then
17:     $particle_{gb} = particle_{new}$  ;
18:  end if
19:  **** Update PV ****
20:  for  $i = 0 ; i < PV.length ; i ++$  do
21:    if  $particle_{gb}, i == 1$  then
22:       $PV_i = PV_i + (1/particle_{gb}.length)$  ;
23:    else
24:       $PV_i = PV_i - 1/particle_{gb}.length$  ;
25:    end if
26:  end for
27:   $t = t + 1$  ;
28: end while
29: return;

```

ALGORITHM 2. Compact particle Swarm Optimization algorithm.

TABLE 1: Descriptions on testing cases.

Testing case	Ontology	Scale
Conference track	Cmt	36 classes
	Conference	60 classes
	Confof	38 classes
	Edas	104 classes
	Ekaw	77 classes
	Iasted	140 classes
	Sigkdd	49 classes
Real sensor ontology	CSIRO sensor ontology ²	33,205 classes
	Semantic sensor network ontology (SSN) ³	32,298 classes
	MMI device ontology ⁴	24,034 classes

6. Conclusion

AIoT aims at creating a sensor network that can communicate and process data, which can be technically implemented by using the sensor ontologies to annotate sensor data with the semantic meanings. To support the co-operations among AIoT applications based on ontologies, it is necessary to integrate these sensor ontologies by finding the alignment

between them. In this work, a novel matching framework is proposed, which aggregates three kinds of CSMs and an alignment extraction approach to determine the ontology alignment. We propose a compact PSO and use it to optimize the aggregating weights for the CSMs and a threshold for filtering the alignment, which ensures the quality of the results. The experimental results show that our proposal can effectively match different sensor ontologies, and the quality of

TABLE 2: Friedman’s test on f -measure of the alignment, and the number in round parentheses is the corresponding computed rank.

Testing case	SOBOM	CODI	ASMOV	FuzzyAlign	cPSO
Cmt-conference	0.50 (5)	0.75 (3)	0.59 (4)	0.87 (2)	0.92 (1)
Cmt-Confof	0.21 (5)	0.38 (3)	0.28 (4)	0.45 (2)	0.57 (1)
Cmt-Edas	0.48 (4)	0.75 (3)	0.42 (5)	0.86 (2)	0.88 (1)
Cmt-Ekaw	0.52 (5)	0.75 (3)	0.59 (4)	0.88 (2)	0.93 (1)
Cmt-Iasted	0.54 (4)	0.78 (3)	0.50 (5)	0.87 (2)	0.90 (1)
Cmt-Sigkdd	0.14 (5)	0.68 (2)	0.34 (4)	0.61 (3)	0.70 (1)
Conference-Confof	0.40 (5)	0.75 (3)	0.51 (4)	0.86 (1.5)	0.86 (1.5)
Conference-Edas	0.44 (5)	0.75 (3)	0.50 (4)	0.88 (2)	0.92 (1)
Conference-Ekaw	0.38 (4)	0.70 (3)	0.32 (5)	0.87 (1.5)	0.87 (1.5)
Conference-Iasted	0.54 (4)	0.73 (3)	0.50 (5)	0.89 (2)	0.94 (1)
Conference-Sigkdd	0.40 (5)	0.70 (3)	0.59 (4)	0.80 (2)	0.86 (1)
Confof-Edas	0.14 (5)	0.36 (3)	0.27 (4)	0.59 (2)	0.73 (1)
Confof-Ekaw	0.25 (5)	0.41 (3)	0.28 (4)	0.52 (2)	0.64 (1)
Confof-Iasted	0.20 (5)	0.70 (3)	0.31 (4)	0.71 (2)	0.72 (1)
Confof-Sigkdd	0.35 (5)	0.61 (3)	0.48 (4)	0.82 (2)	0.89 (1)
Edas-Ekaw	0.10 (5)	0.25 (4)	0.39 (3)	0.44 (2)	0.57 (1)
Edas-Iasted	0.32 (4)	0.72 (3)	0.20 (5)	0.84 (2)	0.88 (1)
Edas-Sigkdd	0.25 (5)	0.58 (3)	0.33 (4)	0.66 (2)	0.75 (1)
Ekaw-Iasted	0.30 (5)	0.64 (3)	0.37 (4)	0.87 (2)	0.94 (1)
Ekaw-Sigkdd	0.22 (5)	0.78 (1)	0.25 (4)	0.74 (3)	0.76 (2)
Iasted-Sigkdd	0.17 (5)	0.72 (3)	0.20 (4)	0.77 (2)	0.85 (1)
MMI device-SSN	0.77 (4)	0.80 (3)	0.73 (5)	0.88 (2)	0.95 (1)
CSIRO-SSN	0.78 (4)	0.79 (3)	0.75 (5)	0.88 (2)	0.94 (1)
MMI device-CSIRO	0.72 (5)	0.78 (3)	0.75 (4)	0.87 (2)	0.92 (1)
Average	0.38 (4.70)	0.66 (2.91)	0.43 (4.24)	0.76 (2.04)	0.82 (1.08)

TABLE 3: Holm’s test on the alignment’s quality.

i	Approach	z value	Unadjusted p value	$\frac{\alpha}{k-i}, \alpha = 0.05$
4	FuzzyAlign	2.04	0.035	0.050
3	CODI	3.91	5.64×10^{-10}	0.025
2	ASMOV	4.24	4.51×10^{-12}	0.012
1	SOBOM	4.70	2.22×10^{-15}	0.008

the alignments obtained by cPSO statistically outperforms other state-of-the-art sensor ontology matching techniques.

In the future, we will further improve cPSO to match the large-scale sensor ontologies, and address the problem of Instance Coreference Resolution (ICR) in the sensor network domain, which requires matching large-scale sensor instances in the Linked Open Data cloud (LOD). We also want to extend cPSO to match the ontologies in the specific domains such as the biomedical domain and geographical domain. The particular strategies and techniques need to be proposed and used to improve the alignment’s precision and recall because these matching tasks require specific background knowledge base and complex forms of alignment.

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 is supported by the Fujian province undergraduate universities teaching reform research project (No. FBjG20190156), the 2018 Program for Outstanding Young Scientific Researcher in Fujian Province University, the Program for New Century Excellent Talents in Fujian Province University (No. GY-Z18155), the Scientific Research Foundation of Fujian University of Technology (No. GY-Z17162), the Science and Technology Planning Project in Fuzhou City (No. 2019-G-40), the Foreign Cooperation Project in Fujian Province (No. 2019I0019), and the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (No. YQ20206).

References

- [1] P. Pande and A. R. 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] G. Katare, G. Padihar, and Z. Qureshi, "Challenges in the integration of artificial intelligence and internet of things," *International Journal of System and Software Engineering*, vol. 6, pp. 10–15, 2018.
- [3] A. Sheth, C. Henson, and S. S. Sahoo, "Semantic sensor web," *IEEE Internet Computing*, vol. 12, no. 4, pp. 78–83, 2008.
- [4] X. Wang, X. Zhang, and M. Li, "A survey on semantic sensor web: sensor ontology mapping and query," *International Journal of u-and e-Service, Science and Technology*, vol. 8, pp. 325–342, 2015.
- [5] X. Xue and J. S. Pan, "An overview on evolutionary algorithm based ontology matching," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, pp. 75–88, 2018.
- [6] X. Xue and Y. Wang, "Optimizing ontology alignments through a Memetic algorithm using both MatchFmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.
- [7] X. Xue, J. Chen, and X. Yao, "Efficient user involvement in semi-automatic ontology matching," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2018, pp. 1–11, 2018.
- [8] X. Xue and J. Chen, "Using compact evolutionary Tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.
- [9] S. Chu, X. Xue, J.-S. Pan, and X. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–23, 2020.
- [10] 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.
- [11] A. L. Ginsca and A. Iftene, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," in *9th Roedunet International Conference*, pp. 118–122, Sibiu, Romania, June 2010.
- [12] 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.
- [13] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 580–591, 2015.
- [14] C. J. V. Rijsberge, *Information Retrieval*, University of Glasgow, Butterworth, London, 1975.
- [15] Y. He, X. Xue, and S. Zhang, "Using artificial bee colony algorithm for optimizing ontology alignment," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, pp. 766–773, 2017.
- [16] J. Wang, Z. Ding, and C. Jiang, "GAOM: genetic algorithm based ontology matching," *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, pp. 617–620, 2006.
- [17] A. Alves, K. Revoredo, and F. Baião, "Ontology alignment based on instances using hybrid genetic algorithm," *Proceedings of the 7th International Conference on Ontology Matching-Volume 946. CEUR-WS.org*, pp. 242–243, 2012.
- [18] J. Bock and J. Hettenhausen, "Discrete particle swarm optimization for ontology alignment," *Information Sciences*, vol. 192, pp. 152–173, 2012.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, Australia, December 1995.
- [20] S. Chu, "An optimal deployment wireless sensor network based on compact differential evolution," *Journal of Network Intelligence*, vol. 2, no. 3, pp. 263–274, 2017.
- [21] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.
- [22] T. T. Nguyen, J.-S. Pan, and T. K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Applied Sciences*, vol. 9, no. 10, pp. 1–18, 2019.
- [23] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, p. 333, 2020.
- [24] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2264–2275, 2015.
- [25] J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt, "Leveraging terminological structure for object reconciliation," in *The Semantic Web: Research and Applications. ESWC 2010. Lecture Notes in Computer Science, vol 6089*, pp. 334–348, Springer, Berlin, Heidelberg, 2010.
- [26] 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.
- [27] P. Xu, Y. Wang, L. Cheng, and T. Zang, "Alignment results of SOBOM for OAEI 2010," *Ontology Matching*, vol. 203, pp. 7–11, 2010.
- [28] 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.
- [29] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
- [30] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, pp. 65–70, 1979.