Complex & Intelligent Systems (2023) 9:435–462
https://doi.org/10.1007/s40747-022-00814-6

ORIGINAL ARTICLE

A multi-objective particle swarm optimization with density and distribution-based competitive mechanism for sensor ontology meta-matching

Aifeng Geng[1] · Qing Lv[1]

Received: 23 January 2022 / Accepted: 16 June 2022 / Published online: 14 July 2022

Abstract

Sensor ontology is a standard conceptual model that describes information of sensor device, which includes the concepts of various sensor modules and the relationships between them. The problem of heterogeneity between sensor ontologies is introduced because different sensor ontology engineers have different ways of describing sensor devices and different structures for the construction of sensor ontologies. Addressing the heterogeneity of sensor ontologies contributes to facilitate the semantic fusion of two sensor ontologies, enabling the sharing and reuse of sensor information. To solve the above problem, an ontology meta-matching method is proposed by this paper to find out the correspondence between entities in distinct sensor ontologies. How to measure the degree of similarity between entities with a set of suitable similarity measures and how to better integrate multiple measures to determine the equivalent entities are the challenges of the ontology meta-matching problem. In this paper, two approximate measurement methods of the quality for ontology matching results are designed, and a multi-objective optimization model for the ontology meta-matching problem is constructed based on these methods. Eventually, a multi-objective particle swarm optimization (MOPSO) algorithm is propounded to dispose of the problem and optimize the quality of ontology meta-matching results, which is named density and distribution-based competitive mechanism multi-objective particle swarm algorithm (D^{2}CMOPSO). The sophistication of the D^{2}CMOPSO based sensor ontology meta-matching method is verified through experiments. Comparing with other matching systems and advanced systems of Ontology Alignment Evaluation Initiative (OAEI), the proposed method can improve the quality of matching results more effectively.

Keywords Sensor ontology · Ontology meta-matching · Multi-objective particle swarm optimization · Competitive mechanism · Ontology Alignment Evaluation Initiative

Introduction

Sensors are widely used in various areas and play an important role in both military and civilian fields. A distributed intelligent network system consisting of numerous sensor nodes with wireless communication capability which are widely distributed in the field of employment is called sensor network. More specifically, a sensor web mainly consists of the Internet of Things (IoT), the World Wide Web (web), sensor network and databases [1,15,18,32]. However, due to the lack of semantic information about sensor devices, data processing, etc., different sensor webs may have the discrepancies, which lead these sensor data difficult to link together, and results in poor coordination between various fields and departments. To eliminate discrepancies between sensor webs and achieve interoperability, the concept of Semantic Sensor Web (SSW) is proposed. SSW technology is a combination of sensor web and sensor ontologies, which annotates sensor data and information such as capabilities, performance, and conditions of use through ontologies, and builds them into standard conceptual models [49]. Thus, the problem of discrepancies between sensor networks is translated into the problem of heterogeneity of conceptual models: since the same concepts in these conceptual models may be depicted in distinct methods, heterogeneity's trouble is introduced [43].

✉ Qing Lv
lvqing_2006@163.com

Aifeng Geng
gengaifeng_1@163.com

[1] College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan 030024, China

By eliminating the heterogeneity of different sensor ontologies, the semantic fusion and interoperability of sensor ontologies are realized, which leads to the sharing and reuse of sensor information. For this reason, the concept of ontology matching is proposed, and ontology matching method aims at studying how to determine the equivalent relationship between entities in two heterogeneous ontologies, which is generally classifiable into two major categories, that is ontology meta-matching method and ontology entity matching method [50]. Both methods require the use of similarity measures to calculate the similarity of different entities and determine their relationships. The ontology meta-matching problem mainly study how to combine and debug different similarity measures to determine high-quality ontology matching results, and this problem is a current research hotspot in ontology matching's area. To solve this problem, the ontology meta-matching method first determines different similarity matrices by means of similarity measures, and then assigns appropriate weights and threshold to these matrices to obtain the final matching results [24]. Since the weights and threshold take values in the range [0,1] of real numbers, the ontology meta-matching problem is usually modeled as a class of continuous optimization problems, which is solvable by intelligent computation methods.

PSO is one of the swarm intelligence algorithms in intelligent computing methods, first proposed by Eberhart and Kennedy in 1995 [11], imitating the foraging behavior of birds in nature, is a classical meta-heuristic algorithm, and is used to resolve optimization problems in different areas owning to fast convergence and high robustness widely. Some properties of PSO make it also suitable for solving ontology meta-matching problem: (1) First, for the objective functions used to evaluate the results, PSO can easily adjust and change them; (2) PSO can handle large scale inputs and is easy in solving slightly larger scale problems such as matching of large scale ontologies; (3) PSO has a high degree of parallelism, which makes it possible to reduce time consumption and improve matching efficiency when using it to handle ontology meta-matching problems [5]. Different users may have different needs for the two evaluation metrics of ontology meta-matching results, i.e., recall and precision [47], so they need to be optimized simultaneously. Both metrics require experts to provide reference matching results in advance, however, in practice, such matching results do not exist, and constructing reference matching results is time-consuming and costly in many cases [59]. To solve this problem, this work proposes two approximate metrics of recall and precision to measure the quality of matching results. Since these two approximate metrics are contradictory to some extent, a multi-objective optimization model is constructed in this work for the sensor ontology meta-matching problem to better describe the nature of the problem. To solve the multi-objective sensor ontology meta-matching problem, this work further proposes the MOPSO [8] to determine the Pareto Front (PF) solutions set. For MOPSO, how to approach the more cutting-edge PF and how to make the distribution of leader solutions more uniform are two challenging problems. The choice of the leader solution is crucial, and it affects the convergence and diversity of the population. To this end, this work proposes an improved MOPSO algorithm, which selects a better and more uniformly distributed set of leader solutions by calculating the sparsity of PF solutions, and then selects a suitable leader solution for each particle in the set of leader solutions based on the density value and the angle competition criterion, to guide the particle to learn from the elite particle that are closer to it convergence direction and approach the optimal PF. In addition, since the sensor ontology meta-matching problem is a multimodal problem, an oppositive solution strategy is introduced into MOPSO to raise solutions' diversity in solution space. Specifically, the contributions of this work are as follows:

(1) In many previous ontology meta-matching systems, two evaluation metrics of its results, i.e., recall and precision, need to use the standard matching results provided by experts in advance, but it is unreasonable and time-consuming. So this work constructs an approximate recall and an approximate precision for ontology meta-matching problem, which are used to measure the matching results' quality. They do not require the intervention of standard reference matching results and are only related to the similarity matrix, which enhances the rationality of matching process.

(2) The sensor ontology meta-matching problem is constructed as a multi-objective mathematical model in terms of the proposed two matching results approximation evaluation metrics. By optimizing the parameters of the model, the meta-matching problem of sensor ontology is solved and the optimal matching result is found.

(3) Facing the challenging problems of MOPSO with many parameters and uneven distribution of PF solutions, a new D^{2}CMOPSO is proposed to optimize multi-objective sensor ontology meta-matching models. It has fewer parameters, the weights and threshold need to be optimized are encoded into the solutions of the algorithm, and the sparsity index of D^{2}CMOPSO is used to select a set of more uniformly distributed leader solutions. Then each particle is judged to tend to exploration or exploitation according to the density index, so that the corresponding leader solution is selected from the set of leader solutions according to the competitive mechanism, guiding the particle to approach the optimal PF with a more uniform distribution. Further the solution on the optimal PF

is found and output, i.e., a set of parameters that make the meta-matching problem has a highest quality of results.

(4) The feasibility of the approach proposed in this work are verified by comparing with other MOPSO-based matching systems and advanced systems of OAEI in the benchmark track, and the experiment results find that D^{2}CMOPSO can match different heterogeneous ontologies more effectively; in addition, the suggested method is compared with some other well-known systems through several sensor ontology test cases to verify its effectiveness. Experiment results show that D^{2}CMOPSO the same possesses better performance in resolving ontology heterogeneity problems in sensor domain, which helps to achieve knowledge integration and information sharing of different sensor networks.

Combined with the above content, the research in this work is divided into the following modules: Section "Related work" introduces reference alignment and partial reference alignment-based matching systems, discusses several existing ontology matching techniques based on PSO as well as the MOPSO algorithm and its variants; Section "Mathematical model for multi-objective ontology meta-matching" first introduces the related definitions of ontology, ontology heterogeneity, ontology meta-matching and similarity measure methods, and then describes the multi-objective optimization model constructed in this work to solve the sensor ontology meta-matching problem; Section "Methodology" introduces and discusses the specific method, implementation process of the D^{2}CMOPSO; Section "Experiment and analysis" is the part of experiments, where the OAEI's benchmark and several sensor ontologies are tested to illustrate the effectiveness of the proposed method; Section "Conclusion and future work" is the conclusion part, which summarizes the content of the whole text, points out the shortcomings of the existing work and indicates the focus and direction of future research.

Related work

Reference alignment and partial reference alignment-based matching systems

In the ontology matching process, different optimization methods can have different effects on the matching results. According to whether and to what extent reference matching results are introduced in the matching process, ontology matching systems are classified as reference alignment (RA), partial reference alignment (PRA) and no reference alignment (NRA)-based systems. Table 1 shows the shortcomings of diffenrent RA and PRA-based matching systems. In the process of ontology matching, the RA-based matching method compare the solution with the reference alignment, and these systems are mainly found in the literature [25–27,46] and [4]. Although it can improve the precision of the matching result to some extent, but it is not reasonable: because it is time & labor-consuming to build the reference alignment in practice. Earlier, PRA-based matching methods have also been put forward to attempt to solve this problem. The PRA-based matching approach first finds a small set of anchor points or standard matching results, and then uses these small-scale entity pairs to find other entity alignments. It can avoid the unnecessary construction of a large number of standard answers, and also improve the matching accuracy. SAMBO [20] is the most famous ontology matching system based on PRA, which uses PRA in different steps of the matching process. LSD [10] is also an automated ontology matching system based on PRA, which requires the user to provide a portion of matching elements to construct partial reference matching results and then train the set of learners through machine learning techniques. ECOMatch [33] also requires the user to provide a part of matching elements, on the basis of which the system parameters are set and the ontology matching process is further completed. However,

Table 1 Reference alignment and partial reference alignment-based matching systems and their shortcomings

Name	Shortcomings
GOAL [26]	It requires the reference matching result to be given in advance
Martinez and Aldana [27]	It needs the help of reference alignments
Xue et al. [46], Marjit [25], Biniz and El Ayachi [4]	None of these methods escape the constraints of reference alignment
SAMBO [20], LSD [10], ECOMatch [33]	These methods require continuous user participation and the selected representative entities does not accurately represent the original ontology
Xue et al. [47]	It is hard to choose a suitable set of small-scale matching pairs to represent the original ontology

these methods require continuous user participation and the selected representative entities does not accurately represent the original ontology. For this reason, Xue et al. [47] propose a PRA-based system using clustering method, where entities in the ontology are divided into different clusters, and entities that can maximize the representation of the original ontology are selected from these clusters. Then it combined with a memetic algorithm to optimize the ontology meta-matching model. But the semi-supervised approach also has certain drawbacks, i.e., how to choose a suitable set of small-scale matching pairs to represent the original ontology, especially in the case of a large amount of data, the selection of matching samples is crucial. The too small scale is not representative, it is difficult to achieve the desired effect; too large scale can be caught in the time-consuming problem of building standard answers.

In recent years, as the defects of RA-based matching systems are gradually exposed, ontology matching technologies based on NRA have become the focus of research, and a NRA-based matching system that can both save costs and improve the accuracy of matching results is urgently needed. The ontology meta-matching technology presented in this work uses approximate evaluation metrics, which belongs to NRA-based matching system.

PSO-based ontology matching technologies

Current matching techniques are divided into two categories according to whether the computational intelligence technology is used: deterministic ontology matching technologies and computational intelligence-based ontology matching technologies. Deterministic ontology matching systems emerged earlier and are characterized by the fact that the relevant parameters in the matching process are given by experts in advance, and it is difficult to adapt the parameters when dealing with different ontology matching problems, so they are not universally applicable. The computational intelligence based ontology matching technology determines the parameters of the matching process automatically through an optimization algorithm, which considers how to find an optimal set of parameters to make the highest quality of matching results, and can adapt to different matching problems with greater flexibility. PSO is a classical swarm intelligence computational method. It has fast convergence speed and strong robustness, and is suitable for solving ontology matching problem. Table 2 summarises the PSO-based ontology matching techniques, including their shortcomings. Bock and Hettenhausen [5] propose a PSO-based method for single-object entity matching. Semenova and Kureychik [34] also use a binary coding mechanism to model ontology matching problem. Further, for different user preferences, Semenova and Kureychik [35] propose an MOPSO-based ontology meta-matching system. However, either of them ignore the structural information of entities in the ontology and lack the structure-based similarity measure. In addition, Marjit [25] also propose an MOPSO-based ontology meta-matching method earlier. Moreover, Kureychik and Semenova [19] use MOPSO to address the application of ontology meta-matching in large-scale data processing. The saving of matching time and memory are particularly important while handle large-scale ontology matching problems. Recently, a multi-objective discrete optimization model is constructed by Wang et al. [40] for the hydrological ontology matching problem, using a compact MOPSO to tune model's parameters. Meanwhile, Xue et al. [48] also propose a compact MOPSO algorithm to solve the biomedical ontology matching problem, using a method called maximum-minimum to determine the winner solution, and experimental results prove that the approach can ameliorate the matching efficiency effectively. The compact intelligent computing method is a compromise strategy [30], although it can reduce the computation time and memory consumption, however, the virtual population strategy results in a smaller population size, and the accuracy of the matching results may be reduced.

At present, there is no ontology meta-matching system based on MOPSO with using NRA. Considering the advan-

Table 2 PSO-based ontology matching technologies and their shortcomings

Name	Shortcomings
Bock and Hettenhausen [5]	Its parameters still need to be determined in advance via parameter file
Semenova and Kureychik [34]	Either of them ignore the structural information of entities in the ontology and lack the structure-based similarity measure
Marjit [25]	Needs to use reference matching results
Wang et al. [40], Xue et al. [48]	The virtual population strategy results in a smaller population size, and the accuracy of the matching results may be reduced

tages and limitations of the above methods, this work first constructs a multi-objective continuous optimization model for ontology meta-matching, applies different similarity measures to compute similarity, and uses improved MOPSO with a more reasonable decimal coding mechanism to automatically optimize the integrated weights of the measures.

MOPSO algorithm and variants

The improved MOPSO is put forward to optimize the matching model's parameters, purpose to improve the quality of matching results. MOPSO was first proposed in 1999 [29], transforming the traditional single-objective PSO algorithm into the algorithm for solving multi-objective problems, the pbest of particle and gbest of population are selected according to the non-dominance relation. Table 3 summarises some of the MOPSO variants, including their shortcomings. Coello et al. [8] make an important improvement to MOPSO by using the Pareto dominance relation to select a set of elite solutions for the population and store them in an external repository, which is updated for each iteration. In addition, the method introduces the variation strategy into MOPSO to improve the diversity of solutions, which results in better PF solutions. More recently, Zhang et al. [57] innovatively propose an MOPSO based on the competitive mechanism, which makes the particles approach the PF better and faster. However, this approach ignores external repository, which result in the lack of historical optimum information of the population. Yuen et al. [54] further improved the competitive mechanism of Zhang et al., this approach also has some drawbacks, i.e., it is not possible to accurately determine which competitive strategy each non-elite solution should adopt to select the appropriate leader solution.

In addition, there is a well-known framework for solving multi-objective optimization problems, known as "Multiple populations for multiple objectives", which is also a creative technique for dividing a population into several sub-populations, each of which corresponds to an optimization objective, and it is a promising research direction. It was proposed by Zhan et al. and can effectively solve the fitness assignment problem. Literature [55] first proposes a co-evolutionary multi-population MOPSO, which uses an external shared archive to store non-dominant solutions, and different populations communicate through this archive; at the same time, an elitist learning strategy (ELS) strategy is introduced into the update process of the archive, which helps the solutions to approach a more realistic PF. Further, the literature [22] also proposes a co-evolutionary based MOPSO, which uses bottleneck objective learning strategy to improve the convergence of the objective, while using elitist learning strategy and juncture learning strategy to make the algorithm covers different parts of the PF, effectively improving the diversity and convergence of the algorithm's solutions. Similar multiple populations for multiple objectives frameworks have been published in [7] and [56].

Table 3 MOPSO algorithm and variants and their shortcomings

Name	Drawback
Coello et al. [8]	The accuracy of the algorithm needs further improvement
Zhang et al. [57]	This approach ignores external repository, which result in the lack of historical optimum information of the population
Yuen et al. [54]	It is not possible to accurately determine what competitive strategy each non-elite solution should adopt to select the appropriate leader solution

To address the problem mentioned in Table 3, the D^{2}CMOPSO in this work first uses a distributivity metric to select elite solutions with better distribution, proposes a density-based method to decide what competitive strategies different non-elite solutions should choose to select appropriate leader solutions for them among elite solutions, and finally generates opposite solution for each particle to increase diversity. This method not only allows the population to be more uniformly distributed in the target space, but also ensures that the solutions converge to a more realistic PF. It has great advantages in solving multimodal multi-objective problems like sensor ontology meta-matching.

Mathematical model for multi-objective ontology meta-matching

Ontology and ontology heterogeneity problem

Ontologies are used to normalize and formalize heterogeneous information, to solve the problem of semantic and syntactic heterogeneity, and are the basis for information interoperability. The ontology consists of numerous entities, which include classes representing concepts, relational properties between concepts and instances of classes, and can be symbolized using a triple, i.e., (C, P, I), where C, P, I represent classes, properties and instances respectively. Figure 1 shows a sensor ontology, the content in the rounded rectangle represents the class of the sensor ontology, and the line connecting the rounded rectangles represents the relationship between two classes. The bidirectional arrows in Fig. 1 indicate that the two classes they connect have a property relationship with each other: for example, "System" and "Procedure", where System implements Procedure and, conversely, Procedure implements by System; while directional arrows represent a one-way property relationship between the emitting class of

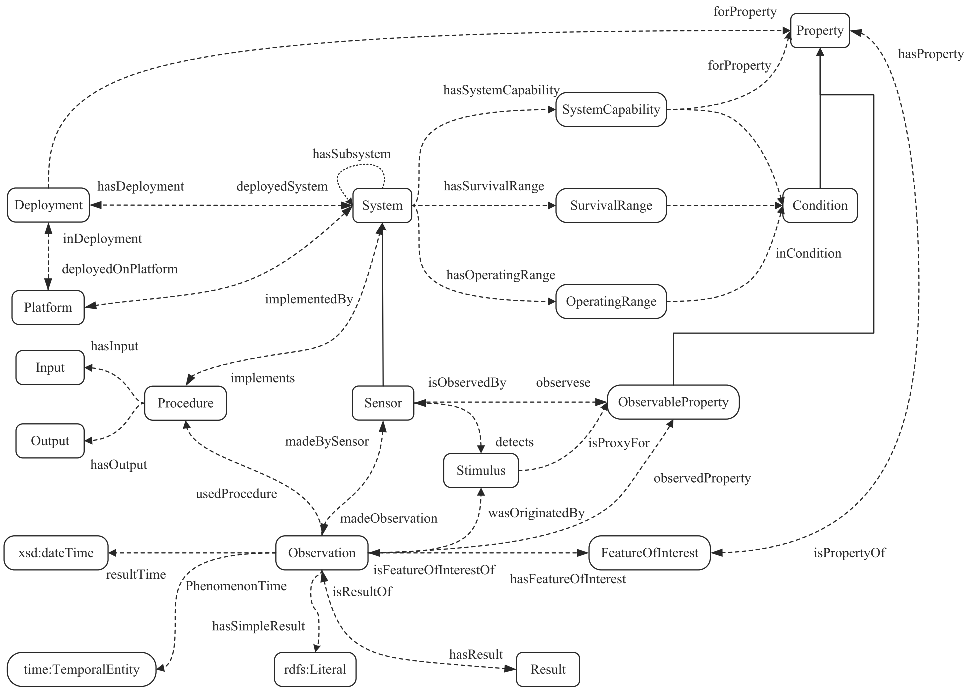

Fig. 1 An example of sensor ontology: contains class and property

the arrow and the receiving class of the arrow: for example, the relationship between “Procedure” and “Output” is that Procedure has Output. However, for different ontology engineers, an ontology for the same domain may have different construction methods, and a concept may be expressed in many distinct terms, which introduces the problem of ontology heterogeneity [16]. Ontology heterogeneity is reflected in three aspects, namely system heterogeneity, syntactic heterogeneity and hierarchical heterogeneity, and solving the heterogeneity of ontology is the key to realize semantic fusion and knowledge sharing. Ontology matching techniques can effectively eliminate semantic heterogeneity and solve the above problem. Figure 2 depicts two heterogeneous sensor ontologies and their matching results. The rectangles and their branches of the figure represent the entities of two ontologies, respectively, and the “≡” represents that the two entities are equivalents, connected by a bidirectional arrow. The names of the entities are shown in the Fig. 2. The names of these entities may be different, but the meanings they represent may be the same, and the two ontologies composed of these entities are heterogeneous. The set of entity pairs connected by bidirectional arrow is the matching result.

Similarity measures

The key to ontology matching is the calculation of similarity, which eliminates heterogeneity by identifying entities with equal relationships through similarity values. Measuring the degree of similarity between entities needs to be considered from three aspects, i.e., syntax, semantics, and structure, because considering the similarity of entities from only one aspect is one-sided and unconvincing. To evaluate the true similarity between two entities, various similarity metric techniques have been recommended and applied to ontology’s process matching. The three types of similarity measurement methods used in this work are: N-Graml [28], Wu&Palmer [41] and numbered hierarchy similarity [44].

Multi-objective ontology meta-matching problem

Ontology meta-matching aims to find the most appropriate set of integration weights for different similarity measurement methods, resulting in the highest quality of matching results. The flow chart of the ontology meta-matching process is shown in Fig. 3. First, parsing two ontologies to be matched to get the entities, then the parsed entities are pre-processed and calculated the similarity values. The different types of similarity values between entities are stored separately in a similarity matrix, i.e., Matrix 1, Matrix 2 and Matrix 3 in Fig. 3. And the different similarity matrices become a combined similarity matrix through the set of weights W_1, W_2 and W_3. The combined similarity matrix is filtered by a threshold to discard low similarity values, and a final similarity matrix is obtained, then the final similarity matrix is used to find the matching result. Find the element in the final similarity matrix that is the maximum of both the row and column in which it is located, and the two entities corresponding to that element are considered to be equivalent.

The evaluation metrics for the quality of ontology meta-matching results are mainly recall, precision and their

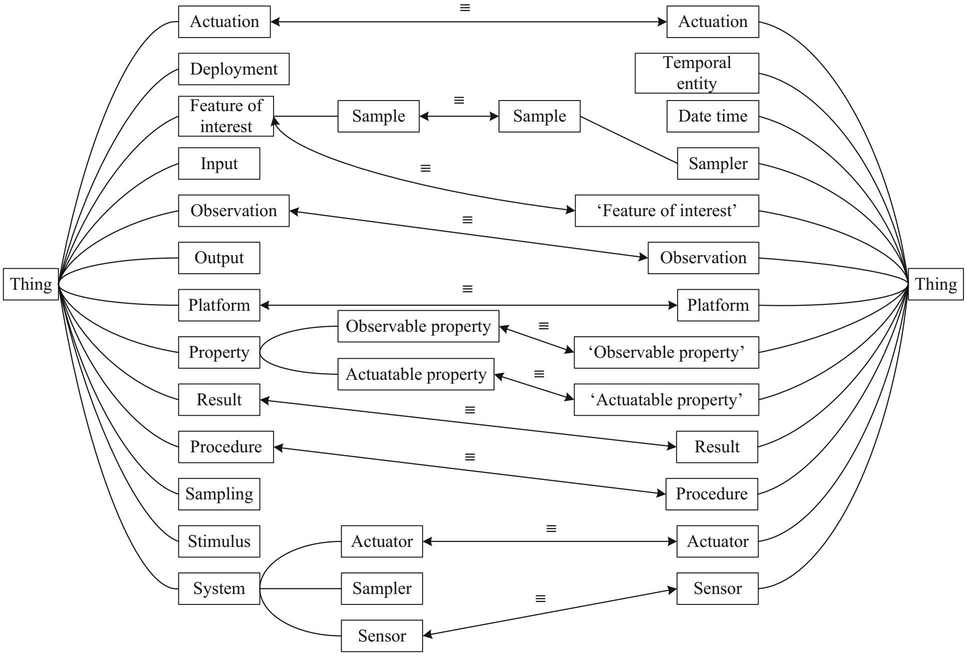

Fig. 2 Two heterogeneous sensor ontologies and their matching results

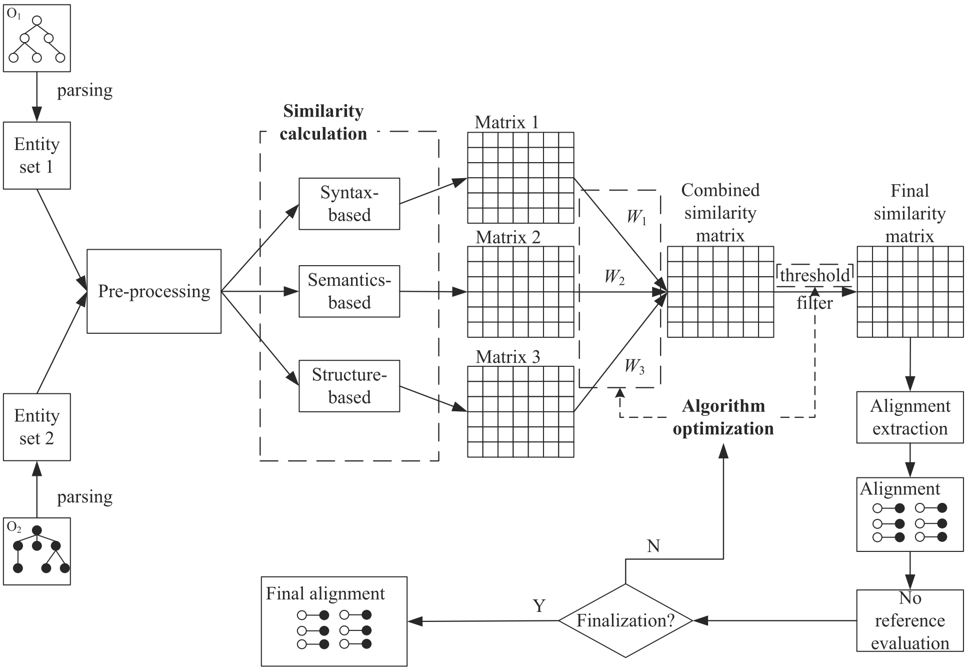

Fig. 3 The flow chat of ontology meta-matching process

harmonic average f-measure [51]. For different users, the importance of different evaluation metrics is also different. Therefore, this work models the sensor ontology meta-matching problem as a multi-objective optimization problem to simultaneously optimize two conflicting metrics, recall and precision, to satisfy the requirements of different users.

Definition of multi-objective problem

Unlike the single-objective optimization problem where one objective is optimized at a time, the Multi-objective Optimization Problem (MOP) requires the simultaneous optimization of more than two conflicting objectives [9].

MOP are common in the real world and many advanced methods have been proposed to solve MOPs in different domains [42,52]. For the maximum optimization problem, the problem is defined as follows:

$$\begin{cases} V-\max & f(x)=[f_1(x), f_2(x), \ldots, f_n(x)]^{\mathrm{T}} \\ s.t. & X \in R^m \end{cases} \tag{1}$$

where V-max denotes vector maximization; $f_k(x)(k = 1, 2, \ldots, n)$ denotes the subobjective of the vector objective function $f(x)$, each subobjective vector is maximized as much as possible; x is the solution to the problem and $X \in R^m$ is the constraints and limitations of the MOP.

Solving an MOP requires determining a set of non-dominated solutions, and the dominance relation of the solutions is defined as follows:

(1) If there are solutions $x_1, x_2 \in X$, for any $k = 1, 2, \ldots, n$ with $f_k(x_1) \geq f_k(x_2)$ and at least one $f_k(x_1) > f_k(x_2)$, then the solution x_1 is said to dominate the solution x_2, denoted as $x_1 \preceq x_2$;
(2) If there are solutions $x_1, x_2 \in X$, for any $k = 1, 2, \ldots, n$, not all of them have $f_k(x_1) \geq f_k(x_2)$, then the solution x_1 and the solution x_2 are said to be mutually non-dominated, denoted as $x_1 \npreceq x_2$.

Formulaic multi-objective ontology meta-matching

The multi-objective ontology meta-matching problem is defined by building a nonlinear optimization model consisting of decision variables, constraints and objective functions. The commonly used metrics to measure alignment's quality are $recall$ and $precision$, which are defined as follows:

$$recall = \frac{|R \bigcap A|}{|R|} \tag{2}$$

$$precision = \frac{|R \bigcap A|}{|A|} \tag{3}$$

where R stands for the standard matching result given by the expert and A stands for the matching result obtained by the matching system. However, in reality, there is no reference result for two matching ontologies, and for large-scale ontologies, the construction of reference results requires high costs, and the approximate evaluation metrics can effectively solve this problem. The two approximate evaluation metrics in this work are $SubstitutiveRecall$ and $SubstitutivePrecision$, which are used as substitutes for $recall$ and $precision$, respectively. They are calculated as follows:

$$substitutiveRecall(M) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\delta(|M_{ij}|)}{\min(m,n)} \tag{4}$$

$$\delta(|M_{ij}|) = \begin{cases} 1, & \text{when element } M_{ij} \text{ is largest in } i\text{th} \\ & \text{row and } j\text{th c-} \\ & \text{olumn in } M \text{ matrix} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where M is the final similarity matrix; M_{ij} denotes the element in the i-th row and j-th column of M; m and n are the cardinalities of the source and target ontologies, respectively.

$$substitutivePrecision(M) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\Upsilon(|M_{ij}|)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\delta(|M_{ij}|)} \tag{6}$$

$$\Upsilon(|M_{ij}|) = \begin{cases} |M_{ij}|, & \text{when element } M_{ij} \text{ is largest} \\ & \text{in } i\text{th row and } j\text{th column} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The above two approximate evaluation metrics no longer require reference alignment, using only the information from the similarity matrix. The ratio of the matching pairs we find to the maximum possible number of matching pairs can be used to replace the $recall$; and the average of the similarity values of the matching pairs we find can be used to replace the $precision$. On this basis, the optimization model for the multi-objective ontology meta-matching problem is defined as follows:

$$\begin{cases} V-\max \quad [substitutiveRecall(M), substitutive \\ \qquad Precision(M)]^{\mathrm{T}} \\ s.t. \quad Inte = (Inte_1, Inte_2, \ldots, Inte_n, Inte_{n+1})^{\mathrm{T}} \\ \quad \sum_{i=1}^{n} Inte_i = 1, Inte_i \in [0,1] \\ \quad Inte_{n+1} \in [0,1] \end{cases} \tag{8}$$

where $Inte$ is the integration vector, the integration factor $Inte_i$ (i=1, …, n) represents the weight of similarity measurement method, and $Inte_{n+1}$ represents the threshold value.

Methodology

Encoding mechanism

In this work, the integration factors, i.e., weights and threshold information, are encoded into the solutions of the algorithm using a decimal encoding mechanism. The specific procedure is as follows: randomly generate n real numbers in the range of [0,1], n is the number of similarity measures; arrange n-1 random numbers in ascending order, and the nth random number represents the threshold; generate weights by calculating the difference between the n-1 ascending numbers, with the following equation:

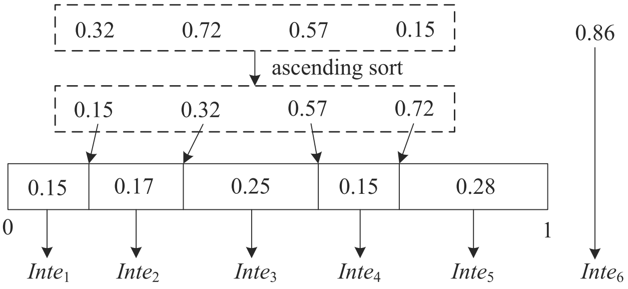

Fig. 4 An example of integration factors encoding mechanism

$$Inte_i = \begin{cases} rand'_1, & i = 1 \\ rand'_i - rand'_{i-1}, & 1 < i < n \\ 1 - rand'_{n-1}, & i = n \end{cases} \tag{9}$$

where $Inte_i$ denotes the weight, i is the ordinal number of the weight, $rand'$ denotes the sorted random number. Figure 4 gives an example to illustrate the encoding mechanism of this work. As shown in Fig. 4, assuming n=5, the five random numbers generated are 0.32, 0.72, 0.57, 0.15 and 0.86, where 0.86 is taken as a threshold and the remaining four numbers are arranged in ascending order, the corresponding five weights are 0.15, 0.17, 0.25, 0.15 and 0.28, respectively. While satisfying the constraints, this encoding method can also reduce dimension.

MOPSO and its challenges

The traditional MOPSO uses the Pareto dominance relationship to select dominated and non-dominated solutions, stores the non-dominated particles in a repository, and updates the external repository at each iteration. Each particle selects an individual historical optimal solution based on the dominance relation and selects the global optimal solution in the external repository, updating the velocity and position based on the Eqs. (10) and (11):

$$vel'[i] = \omega \times vel[i] + r_1 \times (pbest[i] - pos[i]) + r_2 \times (rep[h] - pos[i]) \tag{10}$$

$$pos'[i] = pos[i] + vel'[i] \tag{11}$$

where vel' and vel represent the velocity of the particle after and before updating, respectively; ω is the inertia weight, and r_1 and r_2 are random numbers in interval [0,1]; pos' and pos denote the position of the particle before and after updating, respectively; $pbest$ represents the historical optimal position of the particle, $rep[h]$ is the position of the elite solution selected from the external repository, where h is the index of the elite solution.

The two main challenging problems of MOPSO are (1) how to make solutions better converge to the true PF; (2) how to maintain the diversity of solutions and make the PF's solutions more uniformly distributed. To improve the convergence efficiency and solution quality of the algorithm, an MOPSO based on the competitive mechanism has been proposed in recent years [57], which selects a part of particles in the population as elite solutions by non-dominated sorting, and the particles of the population randomly select two elite solutions at a time for comparison, the elite solution with the smallest angle to that particle is selected as its leader solution. However, there are several problems with this approach: First, the absence of an external repository causes the loss of historical information on the population, thus affecting the diversity and authenticity of the elite solution. Second, since this competitive mechanism selects only two random elite solutions for comparison at a time, other more closer elite solutions may be ignored; moreover, selecting only the elite solution with the smallest angle to the current particle as the leader solution would cause the particle to prefer local exploration and ignore its social part, which is not conducive to the balance of exploration and exploitation for the population. In response to the above problems, Yuen et al. make improvements to competitive mechanism-based MOPSO [54], firstly reintroducing an external repository to save the historical information of the population, secondly each particle dynamically selects multiple elite solutions for comparison according to the number of iterations and selects the appropriate one as the leader solution. In addition, the non-elite solutions are also divided into two equal parts, with one part choosing the elite solution with the smallest angle to it as the leader solution, and the other part choosing the largest. Although this approach can effectively improve the performance of the competitive mechanism-based MOPSO proposed by Zhang et al., it still has some problems in selecting a more efficient set of elite solutions and determining which particles need to be more inclined towards exploration and which particles need to be more inclined towards exploitation.

The proposed D^{2}CMOPSO method

To solve the problems of the above methods, this work further proposes a new MOPSO with density & distribution-based competitive mechanism, which is noted as D^{2}CMOPSO. It is used to optimize the integration factors of the sensor ontology meta-matching problem. A sparsity calculation method [21] is first used to determine the elite solutions, and these better distributed elite solutions are used to lead the particles' update, making the distribution of the PF's solutions more uniform; to approach a better PF and balance the exploration and exploitation, D^{2}CMOPSO determines whether a particle prefers exploration or exploitation based on its den-

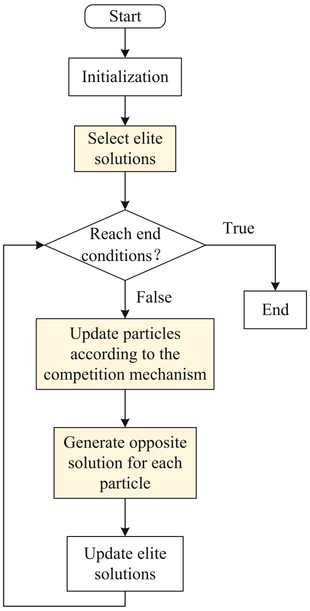
Start
Initialization
Select elite solutions
Reach end conditions?
True
False
End
Update particles according to the competition mechanism
Generate opposite solution for each particle
Update elite solutions

Fig. 5 The flowchart of D^{2}CMOPSO

sity and chooses the appropriate leader solution for each particle; moreover, based on the multimodal nature of the sensor ontology meta-matching model, oppositive solutions are introduced into the updating process of particles to improve the diversity of population in the solution space. The overall flowchart of D^{2}CMOPSO is shown in Fig. 5. The main parts have been highlighted and will be explained in detail in the following subsections.

Selection of elite solutions

The selection of the elite solution is an important step in D^{2}CMOPSO, which is crucial to more accurately guide other particles to the solution with better position and distribution. Sparsity is a criterion to reflect and evaluate the distribution of elite solutions in the objective space, which takes into account both the density and the uniformity of particles, and can more accurately reflect whether the solutions are evenly distributed or not.

After the initialization of the population, all particles are sorted in a fast non-dominated order, 20 (threshold of external repository) solutions from the first, second, …PF are selected in turn and stored in the external repository, and the sparsity value of each particle in the external repository is calculated, which is denoted as $SPAS$; after each iteration, the new non-dominated solutions are merged with the current external repository's solutions, and then the non-dominated sorting is carried out for the combined solutions. The first 20 solutions are selected as the new external repository's solutions, and finally the $SPAS$ values of these solutions are calculated. According to the method proposed in the literature [21], the sparsity value of each dimension of the particle is calculated as follows:

$$SPAS(i) = \frac{Dens(i)}{\max(Dens)} \times \frac{Even(i)}{\max(Even)} \tag{12}$$

where i is the index of the particle; $Dens$ and $Even$ represent the density and uniformity values of the particle in the objective space, respectively, which are calculated according to Eqs. (13) and (14). And each dimension of the particle needs to be sorted in ascending order according to the fitness value before calculation:

$$Dens(i) = \frac{len_{i,1} + len_{i,2}}{Len} \tag{13}$$

$$Even(i) = \frac{\min(len_{i,1}, len_{i,2})}{\max(len_{i,1}, len_{i,2})} \tag{14}$$

where $len_{i,1}$ and $len_{i,2}$ are the distances from the ith particle to the left and right neighbor particle, respectively, and Len is the distance from the first particle to the last one. The density and uniformity of a particle determine its sparsity. The larger $Dens$ indicates that the density is smaller, but the density does not account for the degree of sparsity of one solution, because $len_{i,1}$ and $len_{i,2}$ can be one large and the other small, and therefore cannot prove the distributivity of the solution; therefore, a measure of the uniformity for the solution, $Even$, is introduced to measure the difference between $len_{i,1}$ and $len_{i,2}$, and a larger value indicates a more uniform distribution of the solution. Regardless of which of $Dens$ and $Even$ is too small, the sparsity of particles is not too high, and the solutions with high $SPAS$ can better balance density and uniformity. In solving multidimensional problems, the $SPAS$ value of a particle is the sum of the $SPAS$ values in each dimension, as is the case for $Dens$ and $Even$. The particles in external repository are sorted in descending order of $SPAS$ value, and the top 10 particles are selected as elite solutions, they represent the best distributed solutions in the current population.

Density-based competitive mechanism for selection of leader solution

The main process of the competitive leader solution selection mechanism in this work is: firstly, the type of the particle to be updated is judged, and if the particle belongs to the elite solution, two solutions are randomly selected from the set of elite solutions as the candidate leader solutions, and these two candidate leader solutions are decided which is the real leader solution through the competitive mechanism, i.e., which candidate solution has the smallest angle with the current particle. Improving the convergence speed of particles while enhancing the diversity of population. If the particle does not belong to the elite solution, the neighborhood density of the particle is judged, i.e., its *Dens* value of the objective space is calculated. The *Dens* of each dimension is calculated according to Eq. (13). The larger the *Dens* value the smaller the density of the particle, the more space around it can be explored. And the particle tends to local search, for which the solution with the smallest angle among the candidate leader solutions is selected; conversely, the smaller the value of *Dens* the greater the density of the particle, the greater the need to explore other unknown regions. So choose the solution with the largest angle among the candidate leader solutions for it.

For different non-elite solutions, the number of candidate leader solutions varies. Particles with small density need to enhance local search, so the number of their candidate leader solutions should gradually become smaller as the number of iterations increases, which ensure the diversity of candidate leader solutions in the early iterations and provide more choices for the particles. The probability that elite solution closer to the particle are selected increases, while the particle also have a relatively greater probability of exploring other regions. The particle can approach different solutions of the PF to avoid local convergence; in the late iteration, the number of alternative leader solutions should be reduced to ensure convergence speed and reduce the number of comparisons. While particles with high density require more global search to get rid of local optimum, at the beginning of the iteration, the population diversity is still high, equipping them with a smaller number of candidate leader solutions to improve the computation speed; as the iteration proceeds, the diversity of the population decreases and the particle needs more candidate leader solutions to choose from, thus enhancing the diversity of solutions. Therefore, the number of candidate leader solutions for particles with high density should increase linearly. Figures 6, 7 and 8 depict the competitive leader solution selection mechanisms for elite and non-elite solutions, respectively, and the solid points in the figures are the candidate leader solutions, labeled with letters; the hollow points are the solutions that need to be updated, and the current solution that needs to be updated is marked with the letter S; the numbers represent the different angles. For the elite solution, choose the one with the smallest angle among the two candidate leader solutions, i.e., point A in Fig. 6; for non-elite solutions with small density, the one with the smallest angle among multiple candidates is chosen, i.e., point C in Fig. 7; while for non-elite solutions with large density, the one with the largest angle among multiple candidates is chosen, i.e., point A in Fig. 8.

The particles in the population update their velocity and position according to their chosen leader solution, and the update equations are:

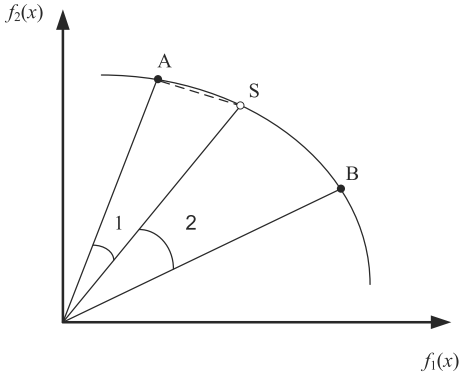

Fig. 6 Leader solution competitive selection mechanism for elite solution

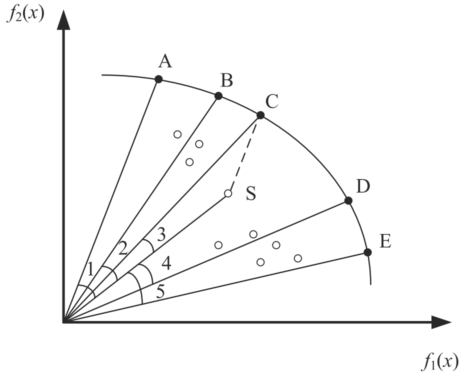

Fig. 7 Leader solution competitive selection mechanism for non-elite solution with small density

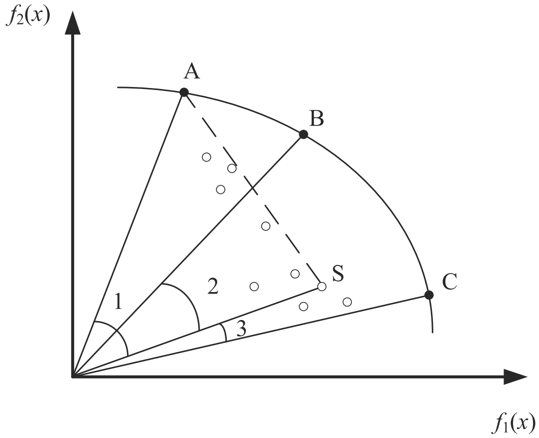

Fig. 8 Leader solution competitive selection mechanism for non-elite solution with large density

$$vel_i^{t+1} = \begin{cases} \omega \times r_1 \times vel_i^t + r_2 \times (pos_l^t - pos_i^t), \\ \quad Dens_i^t \geq MedDens^t \\ \omega \times r_1 \times vel_i^t + r_2 \times (pos_g^t - pos_i^t), \\ \quad Dens_i^t < MedDens^t \end{cases} \tag{15}$$

$$pos_i^{t+1} = pos_i^t + vel_i^{t+1} \tag{16}$$

where t is the number of current iteration; i is the index of the current particle; ω is the inertia weight; r_1 and r_2 are random numbers in interval [0, 1]; vel and pos represent the velocity and position of the particle, respectively; l and g are the indexes of the leader solution with the minimum and maximum angle to particle i, respectively; $MedDens$ denotes the median of the particle density of the population. Whether the particle density is large or small is determined by the relationship between $Dens$ and $MedDens$.

The competitive mechanism can ameliorate the algorithm's convergence speed, it establishes a linkage between the particles convergence direction and the angle, and selects the leader solution among a set of elite solutions based on the angle relationships, thus achieving a balance between exploration and exploitation. The convergence speed and the diversity of solutions also require a reasonable approach to balance. Different particles choose appropriate leader solutions according to whether they are elite solutions and the density of the solutions, which is more conducive to the exploration of particles to unknown regions while ensuring convergence and avoiding falling into local convergence. At the same time, the particle flying towards the well-distributed solutions is also beneficial to guide the update of the elite solutions and obtain the more advanced PF's solutions with better distributivity.

Oppositive solution

Oppositive solution learning (OBL) was first introduced as a new computational intelligence scheme in 2005 [37]. In the last few years, OBL has been successfully applied to various population-based evolutionary algorithms. OBL generates solutions with opposite positions for the algorithm to further improve the space utilization to find the solution with better fitness value. The oppositive candidate solution has a higher probability of reaching the global optimum than a random solution [38]. Therefore, in the algorithm's optimization process, the introduction of oppositive solutions is more helpful to avoid the loss of population diversity and falling into local optimum.

The OBL strategy is applied to solve the sensor ontology meta-matching problem in this work: the upper and lower bounds of the multidimensional space are used to generate oppositive particles thus provide more diverse solutions to D^{2}CMOPSO. The particles generate their corresponding oppositive solutions in the solution space after each update of position, thus making the solutions in the objective space more diverse, which has a positive effect on overcoming the limitations of the solutions for multimodal problems such as sensor ontology meta-matching. The mathematical definition of the oppositive solution is as follows:

$$opp_i^d = low_i^d + upp_i^d - pos_i^d \tag{17}$$

where i is the index of the particle; d is the dimension of the solution; low and upp represent the lower and upper bounds of the solution, respectively; pos is the current position of the solution; and opp is the position of the generated oppositive solution.

Pseudocode and complexity analysis

Combining the methods and strategies discussed in Section "The proposed D^{2}CMOPSO method", the pseudo-code of D^{2}CMOPSO is shown in Algorithm 1:

Algorithm 1 D^{2}CMOPSO

```
Input: Population $P$, population size $N$, external repository $A$, maximum number of iterations $T$, elite solutions set $E$, candidate solutions set $C$ and leader solution $L$
Output: The optimal PF
Initialization of a population $P$
Evaluate the fitness of each particle in $P$, $t = 0$
Calculate $Dens$ and $Even$ for each particle according to Equation 13, Equation 14
Non-dominated sort for each particle and obtain $F_1, F_2, \ldots$
Initialize $A$ according to $F_1, F_2, \ldots$
Evaluate the sparsity value for each particle in $A$ according to Equation 12
Select top 10 particles with best sparsity value into $E$
while $t \le T$ do
    $t$++
    for $i = 1 : N$ do
        if $P[i] \in E$ then
            $candidates = 2$
            $scenario = 1$
        else if $Dens[i] \ge MedDens$ then
            $candidates$ = floor(11 - 9 $*t/T$)
            $scenario = 2$
        else
            $candidates$ = ceil(1 + 4 $*t/T$)
            $scenario = 3$
        end if
        Select $candidates$ particles from $E$ into $C$
        for $j = 1 : candidates$ do
            Calculate angle between $C[j]$ and $P[i]$
        end for
        if $scenario == 1 \| scenario == 2$ then
            $L[i] \leftarrow C[j]$ with smallest angle
        else
            $L[i] \leftarrow C[j]$ with largest angle
            Update the velocity and position of particle $P[i]$ according to Equation 15, Equation 16
            Generate oppositive solution for each particle according to Equation 17
            Update particles according to $P[i]$ and its oppositive solution
            Update $Dens$ and $Even$ for each particle
        end if
    end for
    Update $F_1, F_2, \ldots$
    Update $A$ and $E$
end while
output the optimal PF
```

The first step is the initialization and selection of elite particles: for population initialization, the velocity of each particle is set to 0 and the positions are generated randomly. Two target values are evaluated for each particle and their respective $Dens$ and $Even$ values are calculated using Eqs. (13) and (14), which are used to measure density and uniformity. Determining different PFs based on fast non-dominated sorting and storing the particles from PFs in turn in the external repository A. The SPAS values of particles in A are calculated according to Eq. (12), then the best top 10 particles are selected to add the elite solutions set E. The follows are iterative process, including the competitive selection mechanism for leader solutions: follow the method described previously, if the particle with large $Dens$, a certain number of elite solutions in E are selected as candidate leader solutions, the number of candidate leader solutions decreases linearly with iteration (line 15); conversely, for particles with small $Dens$, the number of candidate leader solutions selected in E increase linearly with iteration (line 18). Each particle updates its velocity and position according to the chosen leader solution, after which an oppositive solution is generated for each particle and the better one is chosen as the updated solution. Finally, the new $Dens$ and $Even$ values for each particle are updated, and the new non-dominated solutions are merged with A, then generate new A by fast non-dominated sorting of the merged A. E for the next iteration are obtained by new A in the same way as in the first step. Output the optimal PF until the end condition is satisfied.

In the above D^{2}CMOPSO algorithm, assuming that the population size is N, in the worst case, the number of computations required to initialize A is $N * (N - 1)/2$, and its computational complexity is O(N^2). The number of computations required for each particle to select the leader solution and update, the update of A is $N * candidates + N * (N - 1)/2$, the total computational complexity is O($T * N^2$) at a number of iterations T. So the computational complexity of the algorithm is O($(T + 1) * N^2$).

Experiment and analysis

Experimental design

The OAEI's benchmark track [36] and 7 test cases consisting of 5 sensor ontologies are used to validate the effectiveness and advancement of the proposed method, where the 5 sensor ontologies are SSN, SOSA, IoT, SN and OSSN. Their detailed descriptions are given in Section "Comparison with OAEI's advanced systems" and Section "Results and analysis of sensor ontology matching". The experiment is split into four portions: the first part is the verification of the effectiveness of the proposed method's strategy and framework, which compares the NRA-based D^{2}CMOPSO matching system with the RA-based D^{2}CMOPSO matching system, NRA-based MOPSO matching system, NRA-based MCMOPSO (a competitive mechanism-based MOPSO without density and sparsity calculations) matching system and NRA-based D^{2}CMOPSO matching system without OBL. Since the ontology meta-matching problem is considered as an MOP in this work, and the PF of the ontology matching problems could not be known in advance, so HV is used to compare their solutions' convergence and distributivity in benchmark test set. In addition, because the knee solution (representative solution) is often used in the verification of MOP-based ontology matching's result [45,46], so another comparison is used to illustrate the quality of the representative solution of different methods, which further demonstrate the superiority of the matching framework without reference alignment and effectiveness of the proposed strategies. The second part is comparison among D^{2}CMOPSO, MOPSO and other three latest related algorithms: SS-MOPSO [31], MO_Ring_PSO_SCD [53] and TriMOEA-TA&R [23], the convergence and distributivity of the solutions obtained by different algorithms are shown by their HV values in benchmark track. The third portion of the experiment compares the matching system proposed in this work with participants of OAEI, which are some start-of-art matchers, to demonstrate its sophistication. The test set used in the first three parts of the experiment is benchmark. The forth portion of the experiment applies the proposed system to the field of sensor ontology matching and compares it with other sensor ontology matching systems, to prove that the method of this work is also applicable to solve the problems in sensor field. Two objective function values, i.e., $recall$ and $precision$ are counted in this work, and these average values are the results after 30 independent runs. Before the proposed method is validated by experiments, a set of suitable parameters need to be configured into the D^{2}CMOPSO-based matching system to maximize the performance of the method.

Parameter configuration

In this work, parameter sensitivity experiment is used to configure an optimal set of parameters for D^{2}CMOPSO so that it performs as best as possible on all test cases. For D^{2}CMOPSO, several important parameters are: the population size N, the inertia weight ω and the maximum number of iterations T, which possess an important effect on the algorithm's performance, so these parameters are determined by experiments. There are three scenarios, each with one variable parameter and two other fixed parameters, as shown in Table 4. All tests in this section are executed on the 304 test case. Tables 5, 6 and 7 record the three scenarios of Table 4 respectively, the numbers before and in parentheses represent

Table 4 Determination of the three main parameters, using the control variable method

Scenario ID	Fix value	Changed value
1	$\omega = 0.5$, $T = 50$	$N = 10, 15, \ldots, 60$
2	$N = 55$, $T = 50$	$\omega = 0.1, 0.2, \ldots, 1.0$
3	$N = 55$, $\omega = 0.5$	$T = 10, 20, \ldots, 100$

Table 5 Results of sensitivity experiments for population size N

Population size N	*recall*	*precision*
10	0.659(0.418)	0.925(0.143)
15	0.771(0.369)	0.876(0.165)
20	0.921(0.057)	0.969(0.013)
25	0.948(0.056)	0.959(0.023)
30	0.935(0.065)	0.966(0.011)
35	0.968(0.006)	0.962(0.004)
40	0.952(0.044)	0.956(0.008)
45	0.955(0.045)	0.959(0.007)
50	0.932(0.059)	0.957(0.012)
55	**0.969(0.003)**	**0.960(0.001)**
60	0.937(0.057)	0.958(0.010)

the mean and standard deviation, respectively, and optimal results are bolded.

Population size N: Table 5 shows the experimental results for the first scenario in Table 4. Considering the quality of solution, the best performance of D^{2}CMOPSO is achieved when the population size is 55. Whether the population size is smaller or larger than 55, the final result is not the best, because the population size is too small, its search speed becomes lower; while the large population size leads to the individual optimal solution is difficult to dominate the evolution direction of overall solutions, and bring redundant information, thus reducing the performance of the algorithm. Therefore, the population size of D^{2}CMOPSO is 55.

Inertia weight ω: The parameter ω is the inertia weight, which is used to control the effect of the particle's velocity in the previous generation on the next generation, so that the particle maintains the inertia of motion and gives it the tendency to expand the search space and the ability to explore new regions. Table 6 shows the experimental results for the second scenario: the algorithm gives the best results when ω is 0.3 or 0.5. If the inertia weight is too large or too small, the result will be affected, because if the ω is large, then the global search ability of the D^{2}CMOPSO is stronger and the search space of the particles is relatively large; conversely, the local search ability of the D^{2}CMOPSO is stronger and the search space of the particles is relatively small. To balance the local and global search capability of D^{2}CMOPSO, its inertia weight value are supposed to set to 0.3 or 0.5. In this paper, 0.5 is selected as the inertia weight value.

Table 6 Experimental results on the sensitivity of inertia weights ω

Inertia weight ω	*recall*	*precision*
0.1	0.966(0.007)	0.956(0.007)
0.2	0.965(0.008)	0.956(0.008)
0.3	0.970(0.000)	0.959(0.003)
0.4	0.938(0.064)	0.955(0.014)
0.5	**0.969(0.003)**	**0.960(0.001)**
0.6	0.938(0.059)	0.955(0.011)
0.7	0.954(0.045)	0.960(0.008)
0.8	0.909(0.073)	0.951(0.010)
0.9	0.915(0.066)	0.955(0.012)
1.0	0.710(0.223)	0.764(0.310)

Maximum number of iterations T: If T is too small, the population can hardly converge; in the later stages of the iteration, when the results of the algorithm hardly change, a larger T will cause the waste of time and storage space. In the third scenario in Table 4, the values of the two parameters N and ω are determined by the above procedure, and the number of iterations is changed from 10 to 100 with a step size of 10. The results are shown in Table 7: D^{2}CMOPSO performs best when the maximum number of iterations is 50. If the number of iterations exceeds 50, the experimental results no longer change significantly. Therefore, 50 is considered to be the most suitable number of iterations for D^{2}CMOPSO.

Validation of the D^{2}CMOPSO strategy and the effectiveness of the NRA-based matching framework

The benchmark track of OAEI is used to validate the effectiveness of the density and distribution-based competitive mechanism strategy and the matching model proposed in this work. The benchmark track is an ontology test cases set containing five different heterogeneous cases covering a wide range of topics, which is suitable for evaluating the strengths and weaknesses of different matchers, the relevant descriptions of it are given in Table 8. Table 9 shows the HV values of NRA-based D^{2}CMOPSO, RA-based D^{2}CMOPSO, NRA-based MOPSO, NRA-based MCMOPSO matching system and NRA-based D^{2}CMOPSO matching system without OBL in benchmark track, including mean and standard deviation. A larger HV indicates better convergence and a more uniform distribution of the solutions, and the HV value of the best performing system on each test case is bolded. As can be seen from Table 9, the HV values of the proposed D^{2}CMOPSO matching system are better than those of MOPSO, MCMOPSO and other two D^{2}CMOPSO matching systems on most test cases: On test cases 101∼104,

Table 7 Experimental results on the sensitivity of the maximum number of iterations *T*

Maximum number of iterations *T*	*recall*	*precision*
10	0.944(0.029)	0.964(0.014)
20	0.962(0.012)	0.962(0.004)
30	0.951(0.034)	0.954(0.010)
40	0.954(0.045)	0.959(0.007)
50	**0.969(0.003)**	**0.960(0.001)**
60	0.965(0.044)	0.956(0.008)
70	0.965(0.036)	0.957(0.009)
80	0.965(0.036)	0.959(0.012)
90	0.969(0.013)	0.960(0.007)
100	0.969(0.006)	0.959(0.004)

Table 8 Relevant descriptions of OAEI's benchmark track

Ontology ID	Lexical feature	Linguistic feature	Structural feature
101∼104	Same	Same	Same
201∼206	Different	Different	Same
221∼247	Same	Same	Different
250∼261	Different	Different	Different
301∼304	–	–	–

203, 221, 222, 224∼237, 241 and 247, the highest HV values are obtained for almost all of these methods, while on the other test cases, NRA-based D^{2}CMOPSO and RA-based D^{2}CMOPSO achieved better HV values than the other methods. Furthermore, the number of test cases in which NRA-based D^{2}CMOPSO achieved the best HV value is higher than that of RA-based D^{2}CMOPSO, which proves the effectiveness of the proposed strategy and matching framework.

Tables 10 and 11 show the results of representative solutions of NRA-based D^{2}CMOPSO matching system, RA-based D^{2}CMOPSO matching system, NRA-based MOPSO matching system, NRA-based MCMOPSO matching system and NRA-based D^{2}CMOPSO matching system without OBL strategy on different test case, where Table 10 shows the mean and standard deviation of their *recall* and Table 11 shows the mean and standard deviation of their *precision*. The numbers in front of the parentheses represent the mean *recall*/*precision* values, and the numbers in parentheses represent the standard deviation, which is used to measure the stability of the system. In addition, *t*-test are used to measure the difference in performance of different matchers. Specifically, the t-statistical test used in this work is as follows: a hypothesis test is conducted to compare the performance of two different algorithms. The null hypothesis is that the performance of the D^{2}CMOPSO matching system proposed in this work is the same as the other compared systems, and the alternative hypothesis is that the performance of the system proposed in this work is different from the other systems. Calculate the t-value and determine whether the t-value is in the rejection domain, if the t-value is in the rejection domain, the null hypothesis is invalid, that is, there is a difference between the two systems. Then the strength and weakness of the systems are determined by comparing their average function values; otherwise the null hypothesis is valid and there is no difference between the two algorithms. The significance level used in this work is 0.05 and the total sample size is 30. Since the original hypothesis is that there is no difference between the two systems, the *t*-test is a bilateral test and the rejection region of the *t*-test is $|t| \geq 2.045$. Tables 12 and 13 show the t-values of *recall* and *precision* for the NRA-based D^{2}CMOPSO matching system and the other four comparison systems, respectively. The values indicating significant differences have been bolded and the "Score" in last row indicate that the number of net winning cases. It can be seen that the t-values on most of the test cases are greater than 2.045 except for some cases in 101∼104 and 221∼247, because all these cases can be found best result by different systems, i.e., 1.00, indicating that the performance of the proposed NRA-based D^{2}CMOPSO matching system and the other four matching systems are significantly different. Then, by comparing their average *recall* and *precision* values in Tables 10 and 11, it can be found that the framework based on NRA can guarantee the quality of matching results even at the expense of a certain degree of accuracy, and the framework can achieve unattended automatic matching, which is more superior. While the NRA-based D^{2}CMOPSO system outperforms the NRA-based MOPSO system, NRA-based MCMOPSO system and NRA-based D^{2}CMOPSO system without OBL, indicating that the competitive mechanism

Table 9 HV values of different matching systems in benchmark track

ID	NRA-based D^2CMOPSO	RA-based D^2CMOPSO	NRA-based MOPSO	NRA-based MCMOPSO	NRA-based D^2CMOPSO without OBL
101	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
103	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
104	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
201	0.94(8.62e−3)	**0.95(8.01e−3)**	0.90(1.64e−2)	0.92(5.98e−2)	0.92(8.67e−3)
203	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
204	**0.98(7.67e−3)**	0.97(7.46e−3)	0.87(1.84e−2)	0.93(1.88e−2)	0.95(7.75e−3)
205	**0.94(2.72e−3)**	0.94(7.85e−3)	0.85(7.63e−2)	0.89(2.13e−2)	0.92(7.84e−3)
206	0.96(4.43e−3)	**0.96(3.98e−3)**	0.88(8.33e−2)	0.90(6.67e−2)	0.95(4.30e−3)
221	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
222	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
223	**0.97(5.39e−3)**	0.96(7.26e−3)	0.90(5.60e−2)	0.95(3.91e−2)	0.96(6.57e−3)
224	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
225	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
228	**1.00(0.00e0)**	**1.00(0.00e0)**	0.99(1.00e−2)	**1.00(0.00e0)**	**1.00(0.00e0)**
232	**1.00(0.00e0)**	**1.00(0.00e0)**	0.98(2.00e−2)	**1.00(0.00e0)**	**1.00(0.00e0)**
233	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
236	**1.00(0.00e0)**	**1.00(0.00e0)**	0.99(1.00e−2)	**1.00(0.00e0)**	**1.00(0.00e0)**
237	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
238	0.96(5.09e−3)	**0.97(6.02e−3)**	0.85(2.68e−2)	0.92(2.55e−2)	0.96(6.64e−3)
239	**0.98(4.09e−3)**	0.98(6.70e−3)	0.94(5.09e−2)	0.96(2.88e−2)	0.96(5.37e−3)
240	**0.98(6.39e−3)**	0.98(6.58e−3)	0.94(2.77e−2)	0.95(8.45e−2)	0.97(6.48e−3)
241	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
246	**0.98(8.22e−3)**	0.98(9.02e−3)	0.93(6.31e−2)	0.95(3.21e−2)	0.96(8.81e−3)
247	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
250	0.95(7.17e−3)	**0.97(7.60e−3)**	0.88(5.31e−2)	0.91(8.51e−3)	0.94(7.66e−3)
257	0.95(8.58e−3)	**0.95(7.17e−3)**	0.88(2.70e−2)	0.90(3.14e−2)	0.94(8.02e−3)
258	0.87(9.15e−3)	**0.88(9.33e−3)**	0.76(3.29e−2)	0.80(2.77e−2)	0.85(5.98e−3)
259	**0.91(3.30e−3)**	0.91(6.90e−3)	0.82(5.97e−2)	0.85(5.43e−2)	0.89(6.37e−3)
260	0.83(5.08e−3)	**0.85(5.81e−3)**	0.75(7.35e−2)	0.78(4.70e−2)	0.83(5.97e−3)
261	**0.75(2.58e−3)**	0.75(6.61e−3)	0.68(6.93e−2)	0.70(4.36e−2)	0.75(5.24e−3)
301	**0.76(4.19e−3)**	0.76(4.98e−3)	0.66(1.72e−2)	0.69(8.27e−3)	0.74(4.83e−3)
302	**0.66(5.02e−3)**	0.66(7.85e−3)	0.54(5.89e−2)	0.60(8.66e−2)	0.64(8.17e−3)
303	**0.62(3.63e−3)**	0.62(4.68e−3)	0.53(6.23e−2)	0.58(3.01e−2)	0.60(4.10e−3)
304	**0.96(3.28e−3)**	0.96(7.25e−3)	0.87(7.51e−2)	0.90(7.34e−2)	0.93(3.96e−3)

strategy based on density and distribution and OBL strategy can effectively improve the accuracy of the MOPSO algorithm.

Comparison with other algorithms

The proposed method is compared with other latest advanced algorithms and tested in benchmark track using HV metric. These algorithms all adopt the NRA-based matching framework, and the larger the HV value is, the more advantageous the method is. Table 14 shows the HV values of the different algorithms, with the best-performing data shown in bold. It can be seen that all methods except MOPSO can obtain maximum HV values in cases 101∼104, 221, 222, 224∼237, 241 and 247. SS-MOPSO performed best and MOPSO performed worst in the other test cases, followings are some specific analysis: for 101∼104 cases, each solution in the resulting solution set is optimal, regardless of the algorithm, because the lexical, linguistic, and structural features of these cases are the same, and any combination of weights and threshold can yield (1.00, 1.00); case 203 does

Table 10 Mean and standard deviation of *recall* for different matching systems

ID	NRA-based D^{2}CMOPSO	RA-based D^{2}CMOPSO	NRA-based MOPSO	NRA-based MCMOPSO	NRA-based D^{2}CMOPSO without OBL
101	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
103	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
104	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
201	0.93(0.01)	0.96(0.02)	0.90(0.02)	0.93(0.02)	0.93(0.02)
203	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
204	0.99(0.01)	0.98(0.01)	0.89(0.02)	0.96(0.02)	0.97(0.01)
205	0.92(0.02)	0.92(0.02)	0.82(0.03)	0.85(0.02)	0.91(0.01)
206	0.92(0.02)	0.92(0.02)	0.88(0.02)	0.90(0.02)	0.92(0.02)
221	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
222	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
223	0.96(0.02)	0.96(0.01)	0.89(0.03)	0.93(0.02)	0.94(0.01)
224	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
225	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
228	1.00(0.00)	1.00(0.00)	0.97(0.02)	1.00(0.00)	1.00(0.00)
232	1.00(0.00)	1.00(0.00)	0.98(0.02)	1.00(0.00)	1.00(0.00)
233	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
236	1.00(0.00)	1.00(0.00)	0.97(0.02)	1.00(0.00)	1.00(0.00)
237	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
238	0.95(0.02)	0.97(0.02)	0.88(0.03)	0.95(0.02)	0.95(0.02)
239	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
240	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
241	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
246	1.00(0.00)	1.00(0.00)	0.93(0.02)	1.00(0.00)	1.00(0.00)
247	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
250	0.94(0.01)	0.98(0.02)	0.90(0.02)	0.90(0.02)	0.92(0.02)
257	0.94(0.01)	0.94(0.01)	0.89(0.03)	0.92(0.02)	0.92(0.02)
258	0.83(0.02)	0.87(0.02)	0.77(0.03)	0.80(0.03)	0.82(0.01)
259	0.88(0.02)	0.88(0.02)	0.80(0.03)	0.84(0.02)	0.86(0.02)
260	0.83(0.02)	0.86(0.01)	0.78(0.02)	0.80(0.02)	0.83(0.02)
261	0.79(0.01)	0.79(0.01)	0.79(0.02)	0.79(0.01)	0.79(0.01)
301	0.82(0.01)	0.82(0.01)	0.75(0.03)	0.76(0.02)	0.80(0.01)
302	0.65(0.01)	0.63(0.02)	0.62(0.02)	0.62(0.02)	0.63(0.02)
303	0.83(0.01)	0.83(0.01)	0.76(0.03)	0.79(0.02)	0.81(0.01)
304	0.97(0.00)	0.97(0.01)	0.91(0.03)	0.93(0.03)	0.95(0.02)

not have high requirements on the accuracy of the lexical and linguistic based matcher, the lexical and linguistic similarity measurement method in this work can effectively measure the similarity. Therefore, algorithms with good convergence can converge to the optimal solutions in this case; for cases 221, 222, 224∼237, 241 and 247, the lexical and linguistic features are the same, the matcher based on good structural similarity measurement method and algorithms with good convergence can obtain solution (1.00, 1.00) on all these cases. In other test cases, SS-MOPSO achieved the highest number of best HV values, followed by TriMOEA-TA&R, although the proposed method is slightly inferior to them, it is still within the acceptable range and better than MOPSO. In conclusion, the similarity measurement method used in this work is more accurate and the matching framework is more reasonable. Since the ontology meta-matching problem is not a complex multimodal problem, D^{2}CMOPSO and SS-MOPSO, MO_Ring_PSO_SCD and TriMOEA-TA&R can obtain satisfactory solutions on benchmark track due to their good convergence and distribution.

Table 11 Mean and standard deviation of *precision* for different matching systems

ID	NRA-based D^2CMOPSO	RA-based D^2CMOPSO	NRA-based MOPSO	NRA-based MCMOPSO	NRA-based D^2CMOPSO without OBL
101	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
103	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
104	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
201	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
203	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
204	1.00(0.00)	1.00(0.00)	0.96(0.03)	0.98(0.02)	1.00(0.00)
205	0.96(0.02)	0.96(0.02)	0.89(0.03)	0.92(0.02)	0.94(0.01)
206	0.98(0.01)	0.98(0.01)	0.92(0.02)	0.94(0.02)	0.96(0.02)
221	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
222	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
223	0.98(0.01)	0.98(0.01)	0.94(0.03)	0.96(0.02)	0.98(0.01)
224	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
225	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
228	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
232	1.00(0.00)	1.00(0.00)	0.98(0.01)	1.00(0.00)	1.00(0.00)
233	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
236	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
237	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
238	0.97(0.02)	0.98(0.01)	0.94(0.02)	0.97(0.02)	0.97(0.02)
239	0.97(0.01)	0.97(0.01)	0.97(0.02)	0.97(0.01)	0.97(0.01)
240	0.97(0.01)	0.97(0.01)	0.97(0.02)	0.97(0.01)	0.97(0.01)
241	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
246	0.97(0.01)	0.97(0.01)	0.96(0.01)	0.97(0.01)	0.97(0.01)
247	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)
250	1.00(0.00)	1.00(0.00)	0.98(0.01)	0.98(0.01)	1.00(0.00)
257	1.00(0.00)	1.00(0.00)	0.96(0.02)	0.98(0.01)	1.00(0.00)
258	1.00(0.00)	1.00(0.00)	0.96(0.02)	0.99(0.01)	1.00(0.00)
259	1.00(0.00)	1.00(0.00)	0.92(0.03)	0.96(0.02)	1.00(0.00)
260	0.96(0.01)	0.98(0.01)	0.96(0.02)	0.96(0.01)	0.96(0.01)
261	0.93(0.02)	0.93(0.02)	0.87(0.03)	0.93(0.02)	0.93(0.02)
301	0.91(0.01)	0.91(0.01)	0.88(0.02)	0.88(0.02)	0.89(0.02)
302	0.89(0.01)	0.88(0.01)	0.85(0.02)	0.85(0.02)	0.87(0.02)
303	0.67(0.01)	0.67(0.01)	0.62(0.02)	0.64(0.01)	0.66(0.01)
304	0.96(0.00)	0.96(0.00)	0.96(0.01)	0.96(0.01)	0.96(0.01)

Comparison with OAEI's advanced systems

To verify the sophistication of the matching model proposed in this work, comparative experiments are conducted in benchmark. D^2CMOPSO is compared with other advanced systems participating in OAEI, which are edna, AML [13], CroMatch [14], LogMap [17], LogMapLt [17], XMap [12] and LogMapBio [17]. Tables 15 and 16 show the results of comparing the OAEI's participants with D^2CMOPSO in terms of *recall* and *precision* values in benchmark, respectively. The numbers in the first column indicate the ID number of the test case, and the numbers in the remaining columns indicate the *recall*/*precision* values. The numbers in the last row of Tables 15 and 16 are the average of the *recall*/*precision* for all test cases. In addition, Tables 17 and 18 respectively show the significance test results of recall and precision in benchmark for the participants of OAEI and proposed method. As the 7 comparison methods are deterministic methods, Friedman test is used to verify whether there are significant differences among different methods, and the significance level is 0.05. The numbers in the first column indicate ID, the numbers in the remaining columns

Table 12 *t*-test for *recall* between different systems

ID	Versus RA-based D^{2}CMOPSO	Versus NRA-based MOPSO	Versus NRA-based MCMOPSO	Versus NRA-based D^{2}CMOPSO without OBL
101	0.00	0.00	0.00	0.00
103	0.00	0.00	0.00	0.00
104	0.00	0.00	0.00	0.00
201	**7.35**	**7.35**	0.00	0.00
203	0.00	0.00	0.00	0.00
204	**2.45**	**19.36**	**5.81**	**7.75**
205	0.00	**15.19**	**13.56**	**2.45**
206	0.00	**7.75**	**3.87**	0.00
221	0.00	0.00	0.00	0.00
222	0.00	0.00	0.00	0.00
223	0.00	**10.63**	**5.81**	**4.90**
224	0.00	0.00	0.00	0.00
225	0.00	0.00	0.00	0.00
228	0.00	**8.22**	0.00	0.00
232	0.00	**5.48**	0.00	0.00
233	0.00	0.00	0.00	0.00
236	0.00	**8.22**	0.00	0.00
237	0.00	0.00	0.00	0.00
238	**−3.87**	**10.63**	0.00	0.00
239	0.00	0.00	0.00	0.00
240	0.00	0.00	0.00	0.00
241	0.00	0.00	0.00	0.00
246	0.00	**19.17**	0.00	0.00
247	0.00	0.00	0.00	0.00
250	**−9.80**	**9.80**	**9.80**	**4.90**
257	0.00	**8.66**	**4.90**	**4.90**
258	**−7.75**	**9.11**	**4.56**	**2.45**
259	0.00	**12.15**	**7.75**	**3.87**
260	**−7.35**	**9.68**	**5.81**	0.00
261	0.00	0.00	0.00	0.00
301	0.00	**12.12**	**14.70**	**7.75**
302	**4.90**	**7.35**	**7.35**	**4.90**
303	0.00	**12.12**	**9.80**	**7.75**
304	0.00	**10.95**	**7.30**	**5.48**
Score	−1	19	13	11

and last row represent the rank of methods and sum rank of each method respectively. After calculation, the test values are both greater than the critical value 14.07, so there are obvious differences among the proposed method and other 7 comparison methods. Then according to Tables 15 and 16, the average values of both *recall* and *precision* of the proposed method are higher than most other matching systems, and only lower than CroMatch. The experimental results show that the method in this paper works well in the face of different heterogeneous situations of ontology matching problems, it can effectively solve different kinds of heterogeneous problems, and has a positive effect on the realization of knowledge integration. The specific analysis is as follows:

For test cases 101∼104, all of the above systems are able to achieve good results because these ontologies have the same lexical, linguistic and structural features, and the matcher based on any of the above features can accurately measure the similarity of the two heterogeneous ontologies and

Table 13 *t*-test for *precision* between different systems

ID	Versus RA-based D^{2}CMOPSO	Versus NRA-based MOPSO	Versus NRA-based MCMOPSO	Versus NRA-based D^{2}CMOPSO without OBL
101	0.00	0.00	0.00	0.00
103	0.00	0.00	0.00	0.00
104	0.00	0.00	0.00	0.00
201	0.00	0.00	0.00	0.00
203	0.00	0.00	0.00	0.00
204	0.00	**7.30**	**5.48**	0.00
205	0.00	**10.63**	**7.75**	**4.90**
206	0.00	**14.70**	**9.80**	**4.90**
221	0.00	0.00	0.00	0.00
222	0.00	0.00	0.00	0.00
223	0.00	**6.93**	**4.90**	0.00
224	0.00	0.00	0.00	0.00
225	0.00	0.00	0.00	0.00
228	0.00	0.00	0.00	0.00
232	0.00	**10.95**	0.00	0.00
233	0.00	0.00	0.00	0.00
236	0.00	0.00	0.00	0.00
237	0.00	0.00	0.00	0.00
238	**−2.45**	**5.81**	0.00	0.00
239	0.00	0.00	0.00	0.00
240	0.00	0.00	0.00	0.00
241	0.00	0.00	0.00	0.00
246	0.00	**3.87**	0.00	0.00
247	0.00	0.00	0.00	0.00
250	0.00	**10.95**	**10.95**	0.00
257	0.00	**10.95**	**10.95**	0.00
258	0.00	**10.95**	**5.48**	0.00
259	0.00	**14.61**	**10.95**	0.00
260	**−7.75**	0.00	0.00	0.00
261	0.00	**15.19**	0.00	0.00
301	0.00	**7.35**	**7.35**	**4.90**
302	**3.87**	**9.80**	**9.80**	**4.90**
303	0.00	**12.25**	**11.62**	**3.87**
304	0.00	0.00	0.00	0.00
Score	−1	15	11	5

obtain more correct matching results. The 201∼202 test cases have ontologies with different lexical and linguistic features, which are similar only in terms of structural features, and the difficulty of matching has increased. Matchers with better and more accurate syntax and semantics-based similarity techniques can achieve better results. Except for CroMatch, the proposed method performs better than other systems in these cases, and although the *precision* is slightly lower than AML, the *recall* is much higher than it. Since CroMatch better combines syntax and semantics-based matchers, it achieves more effective lexical and semantic mapping. So CroMatch's performance is considered to be the best among these participants. However, our method requires less similarity techniques, and it can be regarded as a more reasonable method for solving small-scale matching problems that can guarantee the quality of matching results while improving matching speed and efficiency. The test cases 221∼247 have the same lexical and linguistic characteristics and are less

Table 14 HV values of other related algorithms-based matching systems in benchmark track

ID	D^{2}CMOPSO	SS-MOPSO	MO_Ring_PSO_SCD	TriMOEA-TA&R	MOPSO
101	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
103	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
104	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
201	0.94(8.62e−3)	**0.96(8.71e−3)**	0.95(6.62e−3)	0.95(6.03e−3)	0.90(1.64e−2)
203	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
204	0.98(7.67e−3)	**0.99(7.63e−3)**	0.98(7.60e−3)	0.98(7.68e−3)	0.87(1.84e−2)
205	0.94(2.72e−3)	0.94(2.75e−3)	**0.95(9.79e−3)**	0.94(7.82e−3)	0.85(7.63e−2)
206	0.96(4.43e−3)	**0.96(3.52e−3)**	0.96(6.43e−3)	0.96(5.09e−3)	0.88(8.33e−2)
221	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
222	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
223	0.97(5.39e−3)	**0.97(1.67e−3)**	0.97(7.68e−3)	0.97(5.93e−3)	0.90(5.60e−2)
224	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
225	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
228	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	0.99(1.00e−2)
232	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	0.98(2.00e−2)
233	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
236	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	0.99(1.00e−2)
237	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
238	0.96(5.09e−3)	**0.97(3.55e−3)**	0.97(7.93e−3)	0.97(7.62e−3)	0.85(2.68e−2)
239	0.98(4.09e−3)	0.98(7.45e−3)	0.98(3.79e−3)	**0.98(2.84e−3)**	0.94(5.09e−2)
240	**0.98(6.39e−3)**	0.98(8.26e−3)	0.98(9.45e−3)	0.98(8.38e−3)	0.94(2.77e−2)
241	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
246	0.98(8.22e−3)	0.98(8.22e−3)	0.98(9.97e−3)	**0.98(4.08e−3)**	0.93(6.31e−2)
247	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**	**1.00(0.00e0)**
250	0.95(7.17e−3)	0.95(7.05e−3)	0.95(9.26e−3)	**0.95(4.94e−3)**	0.88(5.31e−2)
257	0.95(8.58e−3)	**0.95(5.34e−3)**	0.95(8.41e−3)	0.95(8.39e−3)	0.88(2.70e−2)
258	0.87(9.15e−3)	**0.88(6.78e−3)**	0.87(9.13e−3)	0.87(2.50e−3)	0.76(3.29e−2)
259	**0.91(3.30e−3)**	0.91(8.87e−3)	0.91(7.18e−3)	0.91(4.26e−3)	0.82(5.97e−2)
260	0.83(5.08e−3)	0.85(7.10e−3)	0.83(5.31e−3)	**0.85(8.62e−3)**	0.75(7.35e−2)
261	0.75(2.58e−3)	**0.75(2.23e−3)**	0.75(3.79e−3)	0.75(2.32e−3)	0.68(6.93e−2)
301	0.76(4.19e−3)	**0.78(5.02e−3)**	0.76(7.38e−3)	0.78(6.30e−3)	0.66(1.72e−2)
302	0.66(5.02e−3)	0.66(7.94e−3)	0.66(6.52e−3)	**0.66(3.23e−3)**	0.54(5.89e−2)
303	0.62(3.63e−3)	**0.64(3.62e−3)**	0.62(7.62e−3)	0.64(9.06e−3)	0.53(6.23e−2)
304	0.96(3.28e−3)	0.97(2.92e−3)	0.96(3.12e−3)	**0.97(2.47e−3)**	0.87(7.51e−2)

demanding for structure-based similarity measures, so most of the matching systems involved in the comparison are able to achieve high results on both objective function values. The proposed method achieves the best results on both *recall* and *precision* of most of cases. Another heterogeneous case is that all the above three features are different, and the test cases consisting of the ontologies with that heterogeneous feature are 250∼261, which are more demanding for the metric technique based on the above three features. Facing these cases, the method in this work can achieve satisfactory results. The final 301∼304 ontologies are belong to real-world, participants of OAEI do not test them.

In summary, the matching method proposed in this work integrates a variety of different similarity measures that can mine the potential correspondence between two heterogeneous ontologies from different perspectives. And a set of best weights and threshold are automatically assigned to them by intelligent computing techniques, which enables to find a larger number of more correct matching pairs. Therefore, the matching technique put forward in this work can attain more excellent results on the four different heterogeneous cases in benchmark, which fully proves the sophistication of the proposed matching model.

Table 15 Comparison on OAEI's participants and D^{2}CMOPSO in terms of *recall* in benchmark track

ID	edna	AML	CroMatch	LogMap	LogMapLt	XMap	LogMapBio	D^{2}CMOPSO
101	1.00	0.00	1.00	0.96	0.99	1.00	0.56	1.00
201	0.03	0.00	1.00	0.00	0.00	0.00	0.00	0.93
202	0.01	0.00	0.82	0.00	0.00	0.00	0.00	0.08
221	1.00	0.34	1.00	0.95	0.99	1.00	0.56	1.00
222	1.00	0.33	1.00	n/a	0.99	0.67	n/a	1.00
223	1.00	0.34	1.00	0.95	0.99	1.00	0.56	0.96
224	1.00	0.34	1.00	0.95	0.99	1.00	0.56	1.00
225	1.00	0.34	1.00	0.96	0.99	1.00	0.56	1.00
228	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
232	1.00	0.34	1.00	0.95	0.99	1.00	0.56	1.00
233	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
236	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
237	1.00	0.33	1.00	n/a	0.99	0.67	n/a	1.00
238	1.00	0.34	1.00	0.96	0.99	1.00	0.56	0.95
239	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
240	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
241	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
246	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
247	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
250	0.79	0.82	0.88	0.79	0.79	0.79	0.79	0.94
257	0.79	0.85	0.82	0.79	0.79	0.79	0.79	0.94
258	0.79	0.29	0.86	n/a	0.78	0.51	n/a	0.83
259	0.79	0.30	0.94	0.74	0.78	0.71	n/a	0.88
260	0.81	0.81	0.88	n/a	0.81	0.81	n/a	0.83
261	0.79	0.82	0.85	0.79	0.79	0.79	n/a	0.79
Ave	0.87	0.58	0.96	0.85	0.87	0.83	0.71	0.93

Results and analysis of sensor ontology matching

The validated D^{2}CMOPSO-based ontology matching system is used to solve the sensor ontology matching problem. The sensor ontology matching problem to be solved consists of five ontologies, which brief descriptions are given in Table 19, and they form seven test cases. The method in this work is compared with several classical matchers, namely Levenshtein distance-based [3], JaroCWinkler distance-based [39], WordNet similarity-based [2] and similarity flooding (SF)-based [6] ontology matcher. Table 20 shows the matching results of each system. The results show that in most of the test cases, the proposed approach achieved better or equal values for *recall* and *precision* than the other matchers, and the *recall* is lower than that of the Levenshtein distance-based, Jaro-Winkler distance-based and WordNet similarity-based matchers only on the SSN-OSSN case, because the matching results of the sensor ontology may not be all one-to-one, there are also one-to-many cases, while our method can only find one-to-one relationships. In addition, the Wordnet dictionary does not contain some specialized vocabulary related to the sensor field, resulting in inaccurate semantic similarity measures and reduced *precision* of matching results on SOSA-SN case.

Conclusion and future work

This work aims to solve the sensor ontology meta-matching problem. Because of the difference of sensor equipment and data processing in sensor networks of different fields and departments, the communication and interoperability between these sensors is difficult. Establishing linkages for sensors contribute to share and reuse sensor information, improve the connection between different departments, promote communication and cooperation, and enable better coordination of measurement, monitoring, detection and rescue efforts. For this purpose, sensor ontologies are constructed and relationships are established for different sensors by eliminating the ambiguity of heterogeneous ontology. Ontology matching is an effective method and technique for achieving semantic fusion, eliminating ambiguity, and

Table 16 Comparison on OAEI's participants and D^{2}CMOPSO in terms of *precision* in benchmark track

ID	edna	AML	CroMatch	LogMap	LogMapLt	XMap	LogMapBio	D^{2}CMOPSO
101	0.64	1.00	1.00	0.94	0.56	0.94	0.50	1.00
201	0.02	1.00	1.00	1.00	0.00	1.00	0.00	1.00
202	0.01	1.00	0.89	1.00	0.00	1.00	0.00	0.80
221	0.64	1.00	1.00	0.94	0.56	0.94	0.50	1.00
222	0.63	1.00	1.00	n/a	0.57	0.94	n/a	1.00
223	0.64	1.00	1.00	0.94	0.56	0.94	0.50	0.98
224	1.00	1.00	1.00	0.94	0.83	0.94	0.50	1.00
225	0.64	1.00	1.00	0.94	0.56	0.94	0.50	1.00
228	0.38	1.00	1.00	0.85	0.32	1.00	0.66	1.00
232	1.00	1.00	1.00	0.94	0.83	0.94	0.50	1.00
233	0.38	1.00	1.00	0.85	0.32	1.00	0.66	1.00
236	1.00	1.00	1.00	0.85	0.67	1.00	0.66	1.00
237	0.99	1.00	1.00	n/a	0.84	0.94	n/a	1.00
238	1.00	1.00	1.00	0.94	0.83	0.94	0.50	0.97
239	0.38	1.00	1.00	0.85	0.32	1.00	0.66	0.97
240	0.38	1.00	1.00	0.85	0.32	1.00	0.66	0.97
241	1.00	1.00	1.00	0.85	0.67	1.00	0.66	1.00
246	1.00	1.00	1.00	0.85	0.67	1.00	0.66	0.97
247	1.00	1.00	1.00	0.85	0.67	1.00	0.66	1.00
250	0.30	1.00	1.00	0.84	0.27	1.00	0.62	1.00
257	0.79	1.00	1.00	1.00	0.67	1.00	0.70	1.00
258	0.78	1.00	1.00	n/a	0.82	0.92	n/a	1.00
259	0.79	1.00	0.98	0.94	0.81	0.93	n/a	1.00
260	0.30	1.00	1.00	n/a	0.27	1.00	n/a	0.96
261	0.30	1.00	0.93	0.87	0.27	1.00	n/a	0.93
Ave	0.64	1.00	0.99	0.91	0.53	0.97	0.53	0.98

solving ontology heterogeneity problems, of which ontology meta-matching is a popular research area. Currently, the main challenges of ontology matching problem are how to optimize the ontology matching process with better no reference alignment method and how to find the most appropriate integration weights for a set of similarity measures. In this work, two approximate measures for the quality of ontology matching results are designed, and a multi-objective optimization model of ontology meta-matching problem is constructed. An MOPSO with density & distribution-based competitive mechanism is further suggested to work out the sensor ontology meta-matching problem and optimize its solution, denoted as D^{2}CMOPSO. D^{2}CMOPSO uses sparsity to measure the distributivity of PF's solutions and selects elite solutions from them, which have the characteristics of both low density and uniform distribution. In addition, each particle decides to prefer local or global search according to its own density, selects a variable number of candidate solutions among the elite solutions, and then chooses the leader solution among the candidate solutions based on the competitive mechanism to guide the update of its flight direction and position, which can effectively balance exploration and exploitation. Finally, three sets of experimentations prove the effectiveness and advanced nature of the proposed strategy and model, which is appropriate to solve different heterogeneous species of ontologies matching problems and has a positive effect on achieving sensor ontology matching and fusion.

This work also has some shortcomings, as the problems solved are small-scale matching problems, D^{2}CMOPSO can ensure the quality of matching results without consuming much time. However, the efficiency of D^{2}CMOPSO-based ontology meta-matching techniques may be reduced when faced with large-scale sensor ontology matching tasks. In next work, we will consider upgrading D^{2}CMOPSO to make it applicable to large-scale matching problems, and a refinement step of matching results is needed to further improve the accuracy of sensor ontology matching.

Table 17 Friedman test for *recall* among D^{2}CMOPSO and OAEI's participants in benchmark track

ID	edna	AML	CroMatch	LogMap	LogMapLt	XMap	LogMapBio	D^{2}CMOPSO
101	2	8	2	6	5	2	7	2
201	3	6	1	6	6	6	6	2
202	3	6	1	6	6	6	6	2
221	2.5	8	2.5	6	5	2.5	7	2.5
222	2	6	2	7.5	4	5	7.5	2
223	2	8	2	6	4	2	7	5
224	2.5	8	2.5	6	5	2.5	7	2.5
225	2.5	8	2.5	6	5	2.5	7	2.5
228	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
232	2.5	8	2.5	6	5	2.5	7	2.5
233	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
236	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
237	2	6	2	7.5	4	5	7.5	2
238	2	8	2	5	4	2	7	6
239	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
240	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
241	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
246	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
247	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
250	6	3	2	6	6	6	6	1
257	6	2	3	6	6	6	6	1
258	3	6	1	7.5	4	5	7.5	2
259	3	7	1	5	4	6	8	2
260	4.5	4.5	1	7.5	4.5	4.5	7.5	2
261	5	2	1	5	5	5	8	5
Sum rank	89.5	140.5	67	141	118.5	106.5	155	80

Table 18 Friedman test for *precision* between D^{2}CMOPSO and OAEI's participants in benchmark track

ID	edna	AML	CroMatch	LogMap	LogMapLt	XMap	LogMapBio	D^{2}CMOPSO
101	6	2	2	4.5	7	4.5	8	2
201	6	3	3	3	7.5	3	7.5	3
202	6	2	4	2	7.5	2	7.5	5
221	6	2	2	4.5	7	4.5	8	2
222	5	2	2	7.5	6	4	7.5	2
223	6	1.5	1.5	4.5	7	4.5	8	3
224	2.5	2.5	2.5	5.5	7	5.5	8	2.5
225	6	2	2	4.5	7	4.5	8	2
228	7	2.5	2.5	5	8	2.5	6	2.5
232	2.5	2.5	2.5	5.5	7	5.5	8	2.5
233	7	2.5	2.5	5	8	2.5	6	2.5
236	3	3	3	6	7	3	8	3
237	4	2	2	7.5	6	5	7.5	2
238	2	2	2	5.5	7	5.5	8	4
239	7	2	2	5	8	2	6	4
240	7	2	2	5	8	2	6	4
241	3	3	3	6	7	3	8	3
246	2.5	2.5	2.5	6	7	2.5	8	5
247	3	3	3	6	7	3	8	3
250	7	2.5	2.5	5	8	2.5	6	2.5
257	6	3	3	3	8	3	7	3
258	6	2	2	7.5	5	4	7.5	2
259	7	1.5	3	4	6	5	8	1.5
260	5	2	2	7.5	6	2	7.5	4
261	6	1.5	3.5	5	7	1.5	8	3.5
Sum rank	128.5	56.5	62	130.5	176	87	186	73.5

Table 19 Brief description of the five sensor ontologies

Sensor ontology	Description	Ontology scale
SN	SensorOntology2009 ontology	152 entities
OSSN	Original semantic sensor network ontology	107 entities
SSN	Semantic sensor network ontology	55 entities
IoT	IoT-lite ontology	40 entities
SOSA	Sensor, observation, sample, and actuator ontology	42 entities

Table 20 Comparison on the proposed method and other classical matchers in terms of *recall* on sensor ontology test cases

Sensor ontology matching task	Levenshtein distance-based matcher	Jaro-Winkler distance-based matcher	WordNet similarity-based matcher	SF-based matcher	Our way
SSN-SN	1.00	1.00	1.00	0.56	1.00
SSN-OSSN	1.00	1.00	0.97	0.35	0.93
SOSA-SN	1.00	1.00	1.00	1.00	1.00
SOSA-OSSN	1.00	1.00	1.00	0.50	1.00
SSN-IoT	1.00	1.00	1.00	1.00	1.00

Table 21 Comparison on the proposed method and other classical matchers in terms of *precision* on sensor ontology test cases

Sensor ontology matching task	Levenshtein distance-based matcher	Jaro-Winkler distance-based matcher	WordNet similarity-based matcher	SF-based matcher	Our way
SSN-SN	1.00	0.90	0.52	0.02	1.00
SSN-OSSN	1.00	0.94	0.80	0.06	1.00
SOSA-SN	0.75	0.75	0.33	0.07	0.92
SOSA-OSSN	1.00	1.00	0.67	0.20	1.00
SSN-IoT	1.00	1.00	0.33	0.01	1.00

Acknowledgements This work was supported by the Taiyuan University of Technology, CN.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Asada G, Dong M, Lin TS, Newberg F et al (1998) Wireless integrated network sensors: Low power systems on a chip. Proceeding of the 24th European Solid-State Circuits Conference (ESSCIRC 1998). Hague, Netherlands, IEEE pp 9–16
2. Beckwith R, Fellbaum C, Gross D, Miller GA (2021) WordNet: A lexical database organized on psycholinguistic principles. Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon. Psychology Press, Hove, pp 211–232
3. Behara KN, Bhaskar A, Chung E (2020) A novel approach for the structural comparison of origin-destination matrices: Levenshtein distance. Trans Res 111:513–530. https://doi.org/10.1016/j.trc.2020.01.005
4. Biniz M, El Ayachi R (2018) Optimizing ontology alignments by using Neural NSGA-II. J Electron Commer Org 16(1):29–42
5. Bock J, Hettenhausen J (2012) Discrete particle swarm optimisation for ontology alignment. Inf Sci 192(6):152–173. https://doi.org/10.1016/j.ins.2010.08.013
6. Boukhadra A, Benatchba K, Balla A (2015) Similarity Flooding for Efficient Distributed Discovery of OWL-S Process Model in P2P Networks. Procedia Comput Sci 56:317–324. https://doi.org/10.1016/j.procs.2015.07.214
7. Chen Z, Zhan Z, Lin Y, Gong Y et al (2018) Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach. IEEE trans cyber 49(8):2912–2926
8. Coello C, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279. https://doi.org/10.1109/TEVC.2004.826067
9. Deb K (2014) Multi-objective optimization. In: Search methodologies, Boston, MA, Springer, pp 403-449. https://doi.org/10.1007/978-1-4614-6940-7_15
10. Doan A, Domingos P, Halevy AY (2001) Reconciling schemas of disparate data sources: a machine-learning approach. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp 509–520. https://doi.org/10.1145/375663.375731
11. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE, pp 39–43
12. Eddine-Djedd W, Tarek-Khadir M, Ben-Yahia S (2016) XMap: results for OAEI 2016. In: Proceedings of the 11th International Workshop on Ontology Matching Co-located with the 15th International Semantic Web Conference, Kobe, Japan
13. Faria D, Pesquita C, Balasubramani BS, Martins C et al. (2016) OAEI 2016 results of AML. In: Proceedings of the 11th International Workshop on Ontology Matching, Kobe, Japan
14. Gulić M, Vrdoljak B, Banek M (2016) CroMatcher-Results for OAEI 2016. In: Proceedings of the 11th International Workshop on Ontology Matching Co-located with the 15th International Semantic Web Conference, Kobe, Japan
15. Hill J, Culler D (2002) A wireless embedded sensor architecture for system-level optimization. UC Berkeley Technical Report: 1-2
16. Jiang C, Xue X (2021) A uniform compact genetic algorithm for matching bibliographic ontologies. Appl Intel 7:1–16. https://doi.org/10.1007/s10489-021-02208-6
17. Jimenez-Ruiz E, Cuenca Grau B, Cross V (2016) LogMap family participation in the OAEI 2016. In: Proceedings of the 11th International Workshop on Ontology Matching Co-located with the 15th International Semantic Web Conference, Kobe, Japan
18. Kahn JM, Katz RH, Pister KS (1999) Next century challenges: mobile networking for "Smart Dust". In: Proceeding of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, WA, USA, pp 271–278. https://doi.org/10.1145/313451.313558
19. Kureychik V, Semenova A (2017) Combined method for integration of heterogeneous ontology models for big data processing and analysis. Computer Science on-line Conference. Springer, Cham, pp 302–311
20. Lambrix P, Liu Q (2009) Using partial reference alignments to align ontologies. In: European Semantic Web Conference, Springer, Berlin, Heidelberg, pp 188–202. https://doi.org/10.1007/978-3-642-02121-3_17
21. Li D, Guo W, Lerch A, Li Y et al (2021) An adaptive particle swarm optimizer with decoupled exploration and exploitation for

large scale optimization. Swarm and Evolutionary Computation 60(7):100789. https://doi.org/10.1016/j.swevo.2020.100789

22. Liu X, Zhan Z, Gao Y, Zhang J et al (2018) Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. IEEE Trans Evol Comput 23(4):587–602. https://doi.org/10.1109/TEVC.2018.2875430
23. Liu Y, Yen GG, Gong D (2018) A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. IEEE Trans Evol Comput 23(4):660–674. https://doi.org/10.1109/TEVC.2018.2879406
24. Lv Q, Jiang C, Li H (2020) Solving ontology meta-matching problem through an evolutionary algorithm with approximate evaluation indicators and adaptive selection pressure. IEEE Access 9:3046–3064. https://doi.org/10.1109/ACCESS.2020.3047875
25. Marjit U (2015) Aggregated similarity optimization in ontology alignment through multiobjective particle swarm optimization. Int J Adv Res 4(4):258–263. https://doi.org/10.17148/IJARCCE.2015.4257
26. Martinez-Gil J, Alba E, Aldana-Montes JF (2008) Optimizing ontology alignments by using genetic algorithms. In: Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web, Karlsruhe, Germany, pp 1–15
27. Martinez-Gil J, Aldana-Montes JF (2011) Evaluation of two heuristic approaches to solve the ontology meta-matching problem. Knowl Inf Syst 26(2):225–247. https://doi.org/10.1007/s10115-009-0277-0
28. Mascardi V, Locoro A, Rosso P (2009) Automatic ontology matching via upper ontologies: a systematic evaluation. IEEE Trans Knowl Data Eng 22(5):609–623. https://doi.org/10.1109/TKDE.2009.154
29. Moore J (1999) Application of particle swarm to multiobjective optimization. Technical report
30. Neri F, Mininno E, Iacca G (2013) Compact particle swarm optimization. Inf Sci 239:96–121. https://doi.org/10.1016/j.ins.2013.03.026
31. Qu B, Li C, Liang J, Yan L et al (2020) A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems. Appl Soft Comput. 86: https://doi.org/10.1016/j.asoc.2019.105886
32. Rabaey J, Ammer J, Da Silva JL, Patel D (2000) PicoRadio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. In: Proceedings IEEE Computer Society Workshop on VLSI 2000. System Design for a System-on-Chip Era, FL, USA, USA, IEEE, pp 9–12. https://doi.org/10.1109/IWV.2000.844522
33. Ritze D, Paulheim H (2011) Towards an automatic parameterization of ontology matching tools based on example mappings. In: Proc. 6th ISWC Ontology Matching Workshop, Bonn, pp 37–48
34. Semenova A, Kureychik V (2016) Application of swarm intelligence for domain ontology alignment. In: Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI’16), Springer, Cham, pp 261–270. https://doi.org/10.1007/978-3-319-33609-1_23
35. Semenova A, Kureychik V (2016) Multi-objective particle swarm optimization for ontology alignment. In: 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, IEEE, pp 1–7. https://doi.org/10.1109/ICAICT.2016.7991672
36. Shvaiko P, Euzenat J, Jimnez-Ruiz E, Cheatham M et al. (2016) Proceedings of the 11th International Workshop on Ontology Matching (OM-2016). Ontology matching workshop. Kobe, Japan, pp 1–252
37. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’06), Vienna, Austria, IEEE, pp 695-701. https://doi.org/10.1109/CIMCA.2005.1631345
38. Wang H, Wu Z, Rahnamayan S, Liu Y et al (2011) Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci 181(20):4699–4714. https://doi.org/10.1016/j.ins.2011.03.016
39. Wang Y, Qin J, Wang W (2017) Efficient approximate entity matching using jaro-winkler distance. International Conference on Web Information Systems Engineering. Springer, Cham, pp 231–239
40. Wang Y, Yao H, Wan L, Li H et al (2020) Optimizing hydrography ontology alignment through compact particle swarm optimization algorithm. In: International Conference on Swarm Intelligence, Springer, Cham, pp 151-162. https://doi.org/10.1007/978-3-030-53956-6_14
41. Wu Z, Palmer M (1994) Verb semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Las Cruces, NM, USA
42. Xu B, Mei Y, Wang Y, Ji Z et al (2021) Genetic Programming with Delayed Routing for Multi-Objective Dynamic Flexible Job Shop Scheduling. Evol Comput 29(1):75–105. https://doi.org/10.1162/evco_a_00273
43. Xue X, Jiang C, Wang H, Tsai PW et al (2021) An improved multi-objective evolutionary optimization algorithm with inverse model for matching sensor ontologies. Soft Computing 2:1–14. https://doi.org/10.1007/s00500-021-05895-y
44. Xue X, Jiang C, Yang C, Zhu H et al (2021) Artificial Neural Network Based Sensor Ontology Matching Technique. In: Companion Proceedings of the Web Conference 2021, Ljubljana, Slovenia, pp 44-51. https://doi.org/10.1145/3442442.3451138
45. Xue X, Wang Y (2017) Improving the efficiency of NSGA-II based ontology aligning technology. Data Knowl Eng 108:1–14. https://doi.org/10.1016/j.datak.2016.12.002
46. Xue X, Wang Y, Hao W (2013) Using MOEA/D for optimizing ontology alignments. Soft Computing 18(8):1589–1601. https://doi.org/10.1007/s00500-013-1165-9
47. Xue X, Wang Y, Ren A (2014) Optimizing ontology alignment through memetic algorithm based on partial reference alignment. Expert Syst Appl 41(7):3213–3222. https://doi.org/10.1016/j.eswa.2013.11.021
48. Xue X, Wu X, Chen J (2020) Optimizing biomedical ontology alignment through a compact multiobjective particle swarm optimization algorithm driven by knee solution. Discrete Dynamics in Nature and Society 2020. https://doi.org/10.1155/2020/4716286
49. Xue X, Wu X, Jiang C, Mao G et al (2021) Integrating sensor ontologies with global and local alignment extractions. Wirel Commun Mob Comput 10:1–10. https://doi.org/10.1155/2021/6625184
50. Xue X, Yang C, Jiang C, Tsai PW et al (2021) Optimizing ontology alignment through linkage learning on entity correspondences. Complexity. https://doi.org/10.1155/2021/5574732
51. Xue X, Yao X (2018) Interactive ontology matching based on partial reference alignment. Applied Soft Computing 72:355–370. https://doi.org/10.1016/j.asoc.2018.08.003
52. Xue Y, Zhu H, Liang J, Slowik A (2021) Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. Knowl Based Syst 227(5):1–9. https://doi.org/10.1016/j.knosys.2021.107218
53. Yue C, Qu B, Liang J (2017) A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans Evol Comput 22(5):805–817. https://doi.org/10.1109/TEVC.2017.2754271
54. Yuen MC, Ng SC, Leung MF (2020) A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem. Cybe Syst 52(3):1–32. https://doi.org/10.1080/01969722.2020.1827795
55. Zhan Z, Li J, Cao J, Zhang J et al (2013) Multiple populations for multiple objectives: A coevolutionary technique for solving multi-

objective optimization problems. IEEE trans cyber 43(2):445–463. https://doi.org/10.1109/TSMCB.2012.2209115

56. Zhang X, Zhan Z, Fang W, Qian P et al. (2021) Multi population ant colony system with knowledge based local searches for multiobjective supply chain configuration. IEEE Trans Evol Comput
57. Zhang X, Zheng X, Cheng R, Qiu J et al (2017) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. Inf Sci 427:63–76. https://doi.org/10.1016/j.ins.2017.10.037
58. Zhang Y, Zuo T, Zhu M, Huang C et al (2021) Research on multi-train energy saving optimization based on cooperative multiobjective particle swarm optimization algorithm. Int J Energy Res 45(2):2644–2667. https://doi.org/10.1002/er.5958
59. Zhu H, Xue X, Jiang C, Ren H (2021) Multiobjective sensor ontology matching technique with user preference metrics. Wirel Commun Mob Comput 5:1–9. https://doi.org/10.1155/2021/5594553

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.