# Ontology Optimization Algorithm for Similarity Measuring and Ontology Mapping Using Adjoint Graph Framework

Jiali Tang, Yu Pan, Zhihong Wang, Linli Zhu

***Abstract*—As a semantic analysis and calculation model, ontology has been applied to many subjects. Numerous of machine learning approaches have been employed to the ontology similarity calculation and ontology mapping. In these learning settings, all information for a concept is formulated as a vector, and the dimension of such vector may be very large in certain special applications. To deal with these circumstances, dimensionality reduction tricks and sparse learning technologies are introduced in ontology algorithms. In this paper, we raise a new ontology framework for ontology similarity measuring and ontology mapping. We construct the adjoint ontology graph by means of index set of ontology vector. The optimal ontology vector is obtained in terms of Lagrangian relaxation approach. Finally, four experiments are presented from various perspectives of different fields to verify the efficiency of the new ontology framework for ontology similarity measuring and ontology mapping applications.**

***Index Terms*—ontology, similarity measure, ontology mapping, adjoint ontology graph.**

## I. Introduction and Motivations

THE term "ontology" is first applied in the field of philosophy to describe the nature connection of things and the inherent hidden connections of their components. As to information and computer science, ontology has been widely applied in knowledge management, machine learning, information systems, image retrieval, information retrieval search extension, collaboration and intelligent information integration as a model for storing and representing knowledge. Owing to it is an effective concept semantic model and a powerful analysis tool, ontology has been used extensively in pharmacology, biology, medicine, geographic information system and social science in the past decade (see [1], [2], [3], [4], [5]).

The structure of ontology can be expressed as a simple graph. On an ontology graph, each vertex represents a concept, object or element in ontology and each (directed or undirected) edge refers to a relationship or hidden connection between two concepts (objects or elements). Let $O$ be an ontology and $G$ be a simple graph of $O$. Essentially, ontology engineer application is to get the similarity calculating function for computing the similarities between ontology vertices. These similarities illustrate the inherent association between vertices in ontology graph. As to ontology mapping, it is to obtain the ontology similarity measuring function by measuring the similarity between vertices from different ontologies. Such mapping connects different ontologies. Through mapping, a potential link between the objects or elements from different ontologies can be acquired. Specifically, the ontology similarity function $Sim : V \times V \to \mathbb{R}^+ \cup \{0\}$ is a semi-positive score function mapping each pair of vertices to a non-negative real number.

In recent years, ontology technologies have been applied in a variety of fields. [6] put forth a technology for stable semantic measurement based on graph derivation representation. [7] developed an ontology representation method for the knowledge in enterprise information of online shopping customers. By processing expert knowledge from external domain ontologies and using novel matching tricks, [8] presented an innovative ontology matching system that finds complex correspondences. [9] explained the main features of the food ontology and the application for traceability using examples. [10] put forward that ontologies can be utilized in designing an architecture for monitoring patients at home.

The advanced idea for handling the ontology similarity computation is using ontology learning algorithm which gets an ontology function $f : V \to \mathbb{R}$. By using the ontology function, the ontology graph is mapped into a line consisting of real numbers. The similarity between two concepts then can be measured by comparing the difference between their corresponding real numbers. Dimensionality reduction is the core of the idea. In order to associate the ontology function with ontology application, a vector is utilized to express all the information of a vertex, for instance $v$. To simplify the representation, the notations are slightly confused and $v$ is used to denote both the ontology vertex and its corresponding vector. The vector is mapped to a real number by ontology function $f : V \to \mathbb{R}$. The ontology function, as a dimensionality reduction operator, maps vectors of multi-dimension into one dimensional ones.

Many effective methods for getting efficient ontology similarity measure or ontology mapping algorithm have been studied in terms of ontology function. By ranking learning technology, [11] discussed the ontology similarity calculation. [12] presented a ontology optimizing model such that the ontology function is determined by means of NDCG measure, and it is successfully applied in physics education. To cut the time complexity for ontology application, [13] presented the fast ontology algorithm. [14] deduced an

Manuscript received Janurary 23, 2018. This work was supported in part by the Industry-Academia Cooperation Innovation Fund Project of Jiangsu Province (BY2016030-06) , the Natural Science Foundation of Shandong Province (ZR2016FM34) and Six Talent Peaks Project in Jiangsu Province (2016-XYDXXJS-020).

Jiali Tang is with the Department of Computer Engineering, Jiangsu University of Technology, Changzhou, China.(email: tangjl@jsut.edu.cn).

Yu Pan is with the Department of Computer Engineering, Jiangsu University of Technology, Changzhou, China.

Zhihong Wang is with the Department of Information Engineering, Shandong Yingcai University, Jinan, China.

Linli Zhu is with the Department of Computer Engineering, Jiangsu University of Technology, Changzhou, China.

ontology algorithm through diffusion and harmonic analysis on hyper-graph framework.

Moreover, several scholars have contributed to the research of ontology algorithms theoretically. [15] put forth the uniform stability of multi-dividing ontology algorithm and the generalization bounds for stable multi-dividing ontology algorithms. [16] cooperated to propose a gradient learning model for ontology similarity measuring and ontology mapping in multi-dividing setting. In the setting, the sample error was determined in terms of the hypothesis space and the ontology dividing operator. More ontology learning algorithms and theoretical analysis can refer to Gao et al. [17], Lan et al. [18] and [19], and Zhu et al. [20].

In this article, a new ontology learning framework for ontology similarity measuring and ontology mapping is introduced by virtue of adjoint ontology graph construction. The rest of the paper is arranged as follows: in Section 2, detailed description of ontology vector learning problem is shown; in Section 3, tools and technologies for optimizing our ontology problem are presented; in Section 4, implementations of main algorithms are determined and corresponding computational complexities are discussed; in Section 5, four respective simulation experiments on plant science, humanoid robotics, biology and physics education are designed to test the efficiency of new ontology optimization method. Final data imply that new approach has higher precision ratio for these applications.

## II. The Notations and Setting of Ontology Problem

Suppose that $V$ is an instance space. For an arbitrary vertex in ontology graph $G$, all its related information is expressed by a $p$ dimensional vector. The information includes its instance, structure, name, attribute, and semantic information of the concept which is corresponding to the vertex and that is contained in its vector. Without loss of generality, assuming that $v = \{v_1, \cdots, v_p\}$ is a vector corresponding to a vertex $v$. To make it simpler, their notations are slightly confused and $v$ is adopted to represent both the ontology vertex and its corresponding vector. To obtain an optimal ontology function $f : V \to \mathbb{R}$, the authors use ontology learning algorithms, and the similarity between two vertices $v_i$ and $v_j$ is therefore determined by the value of $|f(v_i) - f(v_j)|$. Dimensionality reduction, i.e., using real number to represent $p$ dimension vector is the core of such kind of ontology algorithm. Therefore, an ontology function $f$ can be regarded as a dimensionality reduction operator $f : \mathbb{R}^p \to \mathbb{R}$.

The vector corresponding to a vertex of ontology graph always shows high dimension. This is because it contains all the information of the vertex concept, the neighborhood structure and attribute in the ontology graph. Taking the biological ontology for an example, the information of all genes may be contained in a vector. In addition, the ontology structure becomes very complicated owing to there are large numbers of vertices in the ontology graph, and one of the typical examples is the GIS (Geographic Information System) ontology. These above factors may contribute to large similarity calculation in ontology application. But in fact, the similarity between the vertices is always determined by the small number of vector components. For instance, in the biological ontology, it is usually a small number of diseased genes that lead to a genetic disease, while most of the other genes can be ignored. Furthermore, in the application of geographic information system ontology, if an accident happens and causes casualties somewhere, it is better to find the nearest hospital without considering the schools and shops nearby. In other words, we only need to consider the neighborhood information that meets our specific requirements. This is also the reason why lots of academics and industries are interested in the research of sparse ontology algorithm.

In actual application, a sparse ontology function is expressed by

$$f_{\mathbf{w}}(v) = \sum_{i=1}^{p} v_i w_i + \delta. \qquad (1)$$

where $\mathbf{w} = (w_1, \cdots, w_p)$ and $\delta$ are a sparse vector and a noise term, respectively. The former is to reduce the irrelevant component to zero. In order to determine the ontology function $f$, sparse vector $\mathbf{w}$ has to be identified first.

One popular model with the penalization term via the $l_1$-norm of the unknown sparse vector $\mathbf{w} \in \mathbb{R}^p$ is

$$\min_{\mathbf{w} \in \mathbb{R}^p} Y(\mathbf{w}) = l(\mathbf{w}) + \lambda g(\mathbf{w}), \qquad (2)$$

where $\lambda > 0$ is a regularization parameter, which is also known as balance parameter and $l$ is the principal function (or loss function) to measure the quality of $\mathbf{w}$. The balance term measures the sparsity of sparse vector $\mathbf{w}$.

For example, some articles considered that ontology principal function $l : \mathbb{R}^p \to \mathbb{R}$ is a convex differentiable function and $\Omega : \mathbb{R}^p \to \mathbb{R}$ is a sparsity inducing typically non-smooth and non Euclidean norm.

## III. Tools and Technologies for Ontology Optimization

An ontology sparse vector $\mathbf{w} \in \mathbb{R}^p$ is called $s$-sparse if at most $s$ elements of $\mathbf{w}$ are nonzero. The support of $\mathbf{w}$ includes the indices corresponding to nonzero terms in $\mathbf{w}$, that is to say, $\text{supp}(\mathbf{w}) = \{i \in \{1, 2, \cdots, p\} | \mathbf{w}_i \neq 0\}$. By setting $\mathbf{w}_S$ as the restriction of $\mathbf{w}$ to indices in $S$ for fixed a index set $S \subseteq \{1, 2, \cdots, p\}$, we deduce $(\mathbf{w}_S)_i = \begin{cases} \mathbf{w}_i, & \text{if } i \in S \\ 0, & \text{Otherwise} \end{cases}$. The $l_2$-norm of ontology sparse vector $\mathbf{w}$ is denoted by $\|\mathbf{w}\| = \sqrt{\sum_{i \in \{1,2,\cdots,p\}} \mathbf{w}_i^2}$.

In some ontology applications, we have more information about a ontology sparse vector rather than $s$-sparsity. Here, we use the sparsity model raised by [21] for encoding the extra ontology sparse structure: let $\mathbb{S} = \{S_1, S_2, \cdots, S_L\}$ be a collection of supports, where index sets $S_i \subseteq \{1, 2, \cdots, p\}$. Hence, the corresponding ontology sparsity model $\mathcal{S}$ can be expressed as the set of vectors supported by one of the $S_i$:

$$\mathcal{S} = \{\mathbf{w} \in \mathbb{R}^p | \exists S \in \mathbb{S}, \text{supp}(\mathbf{w}) \subseteq S\}. \qquad (3)$$

From this point of view, the the ontology function (1) can be expressed as

$$\mathbf{y} = \mathbf{V}\mathbf{w} + \delta, \qquad (4)$$

where $\mathbf{V} \in \mathbb{R}^{n \times p}$ is the ontology matrix, $\mathbf{y} \in \mathbb{R}^n$ is the ontology response vector, and $\delta$ is the ontology noise and is often formulated as a vector.

The essence of our ontology framework is called the weighted adjoint ontology graph model (in short WAOGM) which is defined by the original ontology graph and its data representation. An potential graph $H = (V, E)$ (here, for convenience, we also use $V$ and $E$ to denote the vertex set and edge set of graph respectively if we know exactly which graph is) is used to define on the coefficients of the unknown vector $\mathbf{w} = \{1, 2, \cdots, p\}$. Moreover, the adjoint ontology graph is edge weighted with weight function $\omega : E \to \mathbb{N}$. We identify supports $S \subseteq \{1, 2, \cdots, p\}$ with subgraphs in $H$, and specially forests (unions of non-cycle graph structure with connected component more than 1). Theoretically, the WAOGM implicates ontology sparsity structures with a small number of connected components in adjoint ontology graph $H$. The WAOGM provides three parameters to grasp the ontology sparsity patterns: the total sparsity of $S$ is denoted by $s$; the maximum number of connected components coming from the forest $F$ corresponding to $S$ is defined by $\kappa$; and the total weight $\omega(F)$ of edges in the forest $F$ corresponding to $S$ is bounded by $M$.

Let $\gamma(H)$ be the number of connected components in a adjoint ontology graph $H$. Then, the $(H, s, g, M)$-WAOGM is defined as the collection of supports

$$\begin{aligned} \mathbb{S} = \{S \subseteq \{1, 2, \cdots, p\} ||S| = s \& \exists F \subseteq H : V_F = S, \\ \gamma(F) = \kappa, \omega(F) \leq M\}. \end{aligned} \tag{5}$$

Let $H = (V, E)$ be a weighted adjoint ontology graph with weight function defined on edge set: $\omega : E \to \mathbb{N}$. The weight-degree $d(v)$ of a vertex $v$ then defined by

$$d(v) = \max_{u \in \mathbb{N}} |\{(v, v') \in E | \omega(v, v') = u\}|, \tag{6}$$

which implies that the largest number of adjacent vertices are connected by edges with the same weight. Thus, the weight degree of $H$ is denoted to be the maximum weight degree of each $v \in V$, and the weight degree of $G$ is the same as the maximum vertex degree if adjoint ontology graph has constant weight function.

Let $\mathbf{w} \in \mathbb{R}^p$ be in the $(H, s, \kappa, M)$-WAOGM. We infer that

$$n = O(s \log \frac{M d(v)}{s} + \kappa \log \frac{p}{\kappa}) \tag{7}$$

independently and identically distributed ontology samples are enough for estimating ontology sparse vector $\mathbf{w}$. Furthermore, for an arbitrary noise vector $\delta \in \mathbb{R}^n$, $y \in \mathbb{R}^n$ is defined as in (4), we can efficiently determine an estimate $\widehat{\mathbf{w}}$ satisfies

$$\|\mathbf{w} - \widehat{\mathbf{w}}\| \leq C\|\delta\|, \tag{8}$$

where $C > 0$ is a constant non-dependent on all parameters. The ontology sparse vector $\mathbf{w}$ is recovered exactly for non-noise situation ($\delta = 0$), and the estimate $\widehat{\mathbf{w}}$ is in a mild extensive WAOGM for any quantity of turbulence.

Specifically, the model projection problem for calculating efficiency course of our ontology sparsity framework is presented as follows: given a vector $\mathbf{b} \in \mathbb{R}^p$ and a WAOGM $\mathcal{S}$, search the optimal approximation to $\mathbf{b} \in \mathcal{S}$,

$$P_{\mathcal{S}}(\mathbf{b}) = \mathrm{argmin}_{\mathbf{b}' \in \mathcal{S}} \|\mathbf{b} - \mathbf{b}'\|. \tag{9}$$

According to Baraniuk et al., [21], we can instantiate the framework to get an algorithm for ontology sparse learning if such a projection is available. However, solving ontology problem (9) is exactly NP-hard for the WAOGM. To avoid such a hard procedure, we use the approximation tolerant framework (see [22], [23], [24], [25] , [26] ).

Instead of determining the solution of (9), our ontology frame can be divided into two implement parts: search an $S \in \mathbb{S}$ with

$$\|\mathbf{b} - \mathbf{b}_S\| \leq cr \min_{S' \in \mathcal{S}} \|\mathbf{b} - \mathbf{b}_{S'}\|, \tag{10}$$

and determine an $S \in \mathbb{S}$ meets

$$\|\mathbf{b}_S\| \geq c_H \cdot \max_{S' \in \mathcal{S}} \|\mathbf{b}_{S'}\|. \tag{11}$$

Here, $c_T > 1$ and $c_H < 1$ are given constants. In next section, Algorithm 1 and Algorithm 2 implicate two nearly-linear time algorithms for solving (10) and (11).

Our model projection ontology approaches are heavily relied on the prize-collecting Steiner tree problem (PCST) (see [27], [28], [29], [30] , [31] ). As the extension of the classical Steiner tree problem, it skips certain terminals from the solution and costs a specific price for each omitted vertex instead of searching the optimal way to connect all terminal vertices in a given weighted adjoint graph. Using PCST to WAOGM, we aim to find a subtree with the optimal balance between the cost paid for edges that are used to connect a subset of the vertices and the price of the rest of non-connected vertices.

## IV. Algorithm Description

In this section, we first introduce a problem of the PCST in which the purpose is to search a forest (called the prize-collecting Steiner forest (PCSF)) with $\kappa$ trees instead of a single tree. Hence, the projection problems (10) and (11) a small set of adaptively constructed PCSF instances are reduced. Then, a nearly-linear time ontology algorithm for the PCSF problem in ontology setting and hence the algorithm for desired ontology problem are presented.

Now, we build the relationship between the weighted adjoint ontology graph model and the prize-collecting Steiner tree (PCST) problem. Note that the PCST problem associated ontology setting can be expressed as: let $H = (V, E)$ be an undirected, weighted adjoint ontology graph with edge cost function $c : E \to \mathbb{R}_0^+$ and vertex prize function $\pi : V \to \mathbb{R}_0^+$. Denote $c(E') = \sum_{e \in E'} c(e)$ for a subset of edges $E' \subseteq E$ and $\overline{V'}$ as the complement $\overline{V'} = V - V'$ for a vertex subset $V' \subseteq V$. Then, the aim of the PCST problem is to determine a subtree $T = (V', E')$ with minimized $c(E') + \pi(\overline{V'})$.

Since PCST is also NP-hard, we should use efficient primal dual algorithm with the following condition

$$c(T) + 2\pi(\overline{T}) \leq 2 \min_{T' isatree} c(T') + \pi(\overline{T'}). \tag{12}$$

Let $\kappa \in \mathbb{N}$ be the objective number of connected components. Hence, we can search a subgraph $F = (V', E')$ with $\gamma(F) = \kappa$ that minimizes $c(E') + \pi(\overline{V'})$ to determine the prize-collecting Steiner forest (PCSF) problem.

As defined before, $\gamma(F)$ is the number of connected components in $F$. For convenience, we say a forest $F$ is a $\kappa$-forest if $\gamma(F) = \kappa$. Since deleting edges always decrease the objective value, an optimal solution for the PCSF problem containing $\kappa$ trees can always be found. It allows us to use the PCSF problem for determining supports in the WAOGM with several connected components.

It is easy to verify that there is an algorithm for the PCSF problem that returns a $\kappa$-forest $F$ satisfies

$$c(F)+2\pi(\overline{F})\leq \min_{F'\subseteq H,\gamma(P')\leq\kappa} 2c(F')+\pi(\overline{F'}). \quad (13)$$

There exists an algorithm obtaining the PCSF condition (13) in time $O(\alpha\cdot|E|\log|V|)$, where $\alpha$ is the number of bits used to specify a single edge cost or vertex prize.

By setting $c(\delta)=\omega(\delta)+1$ and $\pi(i)=u_i^2$, we get

$$c(F)=\omega(F)+(|F|-\kappa)$$

and

$$\pi(\overline{F})=\|\mathbf{b}-\mathbf{b}_F\|_2^2,$$

which connects the PCSF to the WAOGM. The PCSF objective $\lambda c(F)+\pi(\overline{F})$ essentially becomes a Lagrangian relaxation (see [32], [33], [34], [35] , [36] for the technologies of LR) of the ontology sub problem (10) after multiplying the edge cost function with a balance number $\lambda$. Algorithm 1 showed below the PCSF for ontology sub problem (10) which obtains the forest $F$.

Algorithm 1: Forest construction.
Step 1: Input: $H, c, \pi, \kappa$, cost parameter $C$, control parameters $\nu$ and $\epsilon$. Set $c_\lambda(\delta)=\lambda c(\delta)$.
Step 2: $\min_{\pi(i)>0}\pi(i)\to\pi_{\min}$, $\frac{\pi_{\min}}{2C}\to\lambda_0$.
Step 3: $PCSF-WAOGM(G,c_\lambda,\pi,\kappa)\to F$.
Step 4: If $c(F)\leq 2C$ and $\pi(\overline{F})=0$ then return $F$.
Step 5: $0\to\lambda_r$, $3\pi(G)\to\lambda_l$, $\frac{\pi\min\epsilon}{C}\to\varepsilon$.
Step 6: while $\lambda_l-\lambda_r>\varepsilon$ do $(\lambda_l+\lambda_r)/2\to\lambda_m$ and $PCSF-FC(H,c\lambda_m,\pi,\kappa)\to F$.
Step 7: if $c(F)\geq C$ and $c(F)\leq\nu C$ then return $F$; if $c(F)>\gamma C$ then $\lambda_m\to\lambda_r$ else $\lambda_m\to\lambda_l$.
Step 8: end while
Step 9: return $PCSF-FC(H,c_\lambda,\pi,\kappa)\to F$

In Algorithm 1, $FC$ is a certain classical forest contracture approach (see [38], [37] ). Let $\mathcal{S}$ be a $(H,s,\kappa,B)$-WAOGM, $\mathbf{u}\in\mathbb{R}^p$, and $\nu>2$. There exsits an algorithm that return a support $S\subseteq\{1,2,\cdots,p\}$ in the $(H,2\nu,s+\kappa,\kappa,2\nu,B)$-WAOGM meeting (10) with $c_T=\sqrt{1+\frac{3}{\nu-2}}$. Furthermore, the time complexity of Algorithm 1 is $O(|E|\log p)$.

By multiply the vertex prize function instead of the edge cost function with a parameter $\lambda$, we present the algorithm for ontology sub problem (11) and determine the optimal ontology sparse vector $\mathbf{w}$.

Algorithm 2: Ontology sparse vector learning
Step 1. Input: $y,V,H,s,\kappa,B$, iterative number $t$. Set $0\to\widehat{\mathbf{w}}$.
Step 2: for $i\leftarrow 1,\cdots,t$ do
Step 3: $\mathbf{V}^T(\mathbf{y}-\mathbf{V}\widehat{\mathbf{w}}_{i-1})\to\mathbf{b}$, $\mathrm{supp}(\widehat{\mathbf{w}}_{i-1})\cup$ $[\mathrm{Algorithm1for}(\mathbf{b},H,s,\kappa,B)\to S'$, $\mathbf{V}_{S'}^{\dagger}\mathbf{y}\to z_{S'}$, and $0\to z_{S'^C}$.
Step 4: Algorithm 1 for $(z,H,s,\kappa,B)\to S$, $z_S\to\widehat{\mathbf{w}}_i$.
Step 5: end for
Step 6: return $\widehat{\mathbf{w}}_i\to\widehat{\mathbf{w}}$

Here, for matrix $\mathbf{A}$, $\mathbf{A}^{\dagger}$ is denoted as its pseudo inverse. There exists a subtree $T'\subseteq T$ with $\pi(T')\geq\frac{C'\pi(T)}{6c(T)}$ and $c(T')\leq C'$, where $T$ is a tree with $C'\leq c(T)$ and a subtree $T'$ can be determined in linear time. Let $\mathcal{S}$ be a $(H,s,\kappa,B)$-WAOGM and let $b\in\mathbb{R}^p$. Then, there is an algorithm that returns a support $S\subseteq\{1,2,\cdots,p\}$ with the $(H,2s+\kappa,\kappa,2B)$-WAOGM satisfying (11) with $c_H\approx 0.267$. The Algorithm 2 runs in time $O(|E|\log p)$.

Finally, the procedures of ontology similarity measuring and ontology mapping based on Algorithm 1 and Algorithm 2 are presented in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3. Ontology similarity measuring with adjoint ontology graph framework:
Step 1. Mathematizing ontology information. For each vertex in ontology graph, we use a vector to express all of its information.
Step 2. By Algorithm 1 and iteration Algorithm 2, we get the ontology sparse vector, and then map the ontology graph to the real line and map the vertices of ontology graph to real numbers.
Step 3. For each $v\in V(G)$, we use one strategy to obtain the similar vertices and return the outcomes to the users.

Algorithm 4. Ontology mapping with adjoint ontology graph framework:
Let $G_1,G_2,\cdots,G_m$ be ontology graphs corresponding to ontologies $O_1,O_2,\cdots,O_m$ (here $m$ is the number of ontologies).
Step 1. Mathematizing ontology information. For each vertex in ontology graph, we use a vector to express all of its information.
Step 2. By Algorithm 1 and iteration Algorithm 2, we get the ontology sparse vector, and then map the ontology graph to the real line and map the vertices of ontology graph to real numbers.
Step 3. For $v\in V(G_i)$, where $1\leq i\leq m$, we use one strategy to obtain the similar vertices and return the outcome to users.

Note that in step 3 of Algorithm 3 and Algorithm 4, the strategy for our experiment will be presented in the next section.

## V. EXPERIMENTS

Four simulation experiments relating to the ontology similarity measure and ontology mapping are designed below. In order to adjacent to the setting of ontology algorithm, a $p$-dimensional vector is applied to express the information of each vertex. The information includes the name, instance, attribute and structure of a vertex. Here the instance of vertex is used to express the set of the reachable vertex in the directed ontology graph.

The effectiveness of main ontology algorithm is verified in the following four experiments. After obtaining the sparse vector $\mathbf{w}$, then the $f_{\mathbf{w}}(v)=\sum_{i=1}^{p}v_i\mathbf{w}_i$ causes the ontology function.

### *A. Ontology similarity measure experiment on plant data*

In the first experiment, "PO" ontology built in http://www.plantontology.org. was adopted to test the efficiency of the proposed new algorithm for ontology similarity measuring. The basic structure of "PO" is shown in Fig. 1. $P@N$ (see [39] for more details) standard was also used for this experiment. Furthermore, ontology methods discussed in [11], [13] and [12] for the "PO" ontology were considered as well. The accuracies of these three algorithms were calculated to compare with that obtained through the algorithm proposed in the paper. Table 1 presents part of the data.

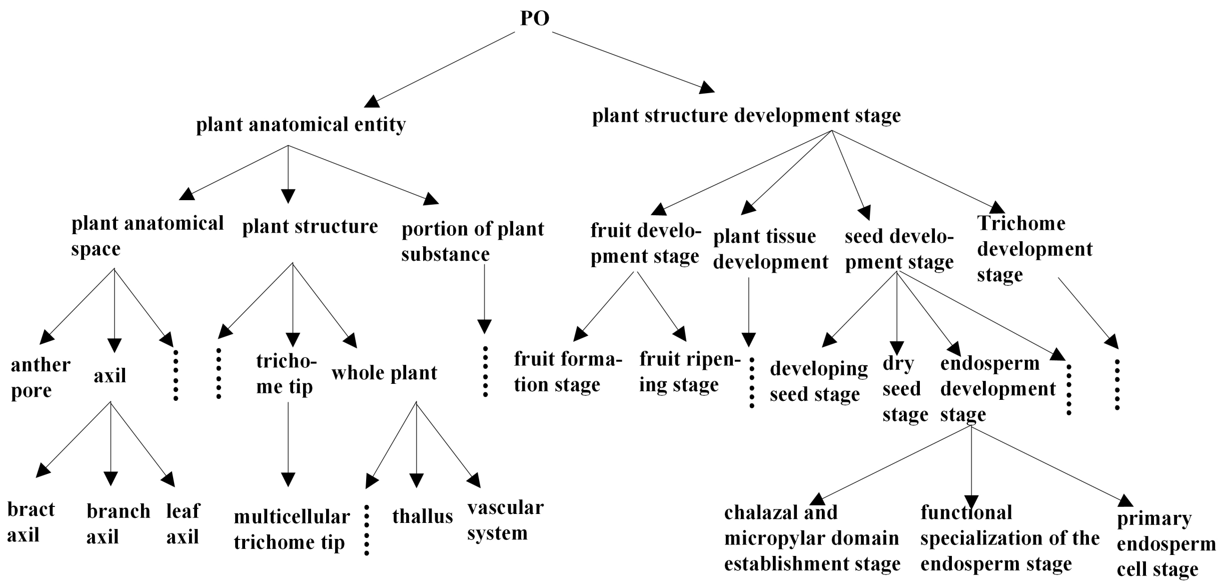


Fig. 1. The Structure of "PO" Ontology

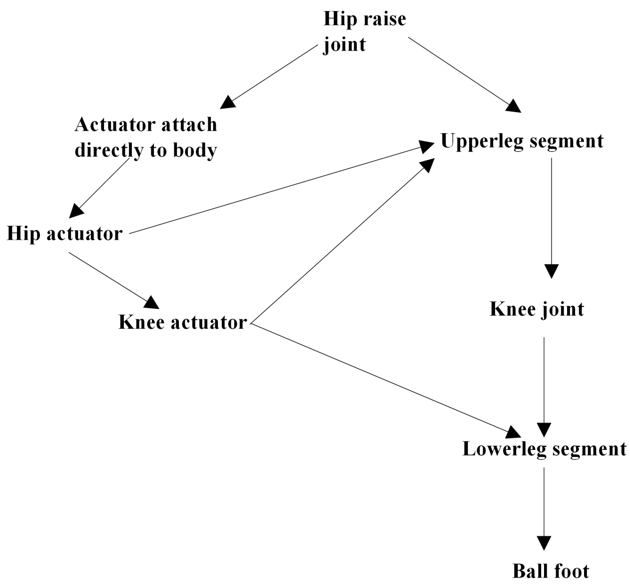


Fig. 2 "humanoid robotics" ontology $O_2$

We first give the closest $N$ concepts for every vertex on the ontology graph by experts in plant field, and then we obtain the first $N$ concepts for every vertex on ontology graph by the Algorithm 3, and compute the precision ratio. Specifically, for vertex $v$ and given integer $N > 0$. Let $Sim_v^{N,\text{expert}}$ be the set of vertices determined by experts and it contains $N$ vertices having the most similarity of $v$. Let

$$v_v^1 = \underset{v' \in V(G)-v}{\text{argmin}} \{|f(v) - f(v')|\},$$

$$v_v^2 = \underset{v' \in V(G)-\{v,v_v^1\}}{\text{argmin}} \{|f(v) - f(v')|\},$$

$$\cdots$$

$$v_v^N = \underset{v' \in V(G)-\{v,v_v^1,\cdots,v_v^{N-1}\}}{\text{argmin}} \{|f(v) - f(v')|\},$$

and

$$Sim_v^{N,\text{algorithm}} = \{v_v^1, v_v^2, \cdots, v_v^N\}.$$

Then the precision ratio for vertex $v$ is denoted by

$$\text{Pre}_v^N = \frac{|Sim_v^{N,\text{algorithm}} \cap Sim_v^{N,\text{expert}}|}{N}.$$

The $P@N$ average precision ratio for ontology graph $G$ is then stated as

$$\text{Pre}_G^N = \frac{\sum_{v \in V(G)} \text{Pre}_v^N}{|V(G)|}.$$

As we can see from what depicted above, the precision ratio which is got from our newly raised algorithm proves to be much higher than those from the previous raised ones in [11], [13] and [12], when $N$=3, 5, or 10. To be detailed, all the ratios from algorithm 3, the new one, are above 0.5, which shows good efficiency and accuracy. In other word, it overshadows the previous algorithms in [11], [13] and [12] in efficiency. Its obvious that when the number of $N$ is becoming larger, the ratios increase larger and

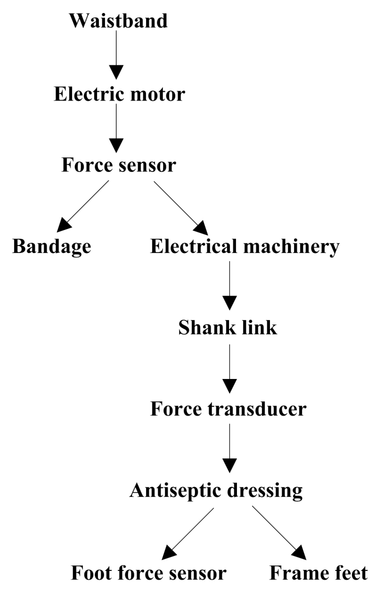


Fig. 3 “humanoid robotics” ontology $O_3$

TABLE I
THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

| | $P$@3 average precision ratio | $P$@5 average precision ratio | $P$@10 average precision ratio |
|---|---|---|---|
| Algorithm 3 in our paper | 0.5047 | 0.5983 | 0.7538 |
| Algorithm in [11] | 0.4549 | 0.5117 | 0.5859 |
| Algorithm in [13] | 0.4282 | 0.4849 | 0.5632 |
| Algorithm in [12] | 0.4831 | 0.5635 | 0.6871 |

more apparently with higher scores. That is, the advantage of our new algorithm becomes overwhelming, when $N$ is large enough and sufficiently. Therefore, the new algorithm proposed in this paper is superior to those proposed in [11], [13] and [12].

### B. Ontology mapping experiment on humanoid robotics data

Humanoid robotics ontologies $O_2$ and $O_3$ which were constructed in [16] were applied. Fig. 2 and Fig. 3 exhibit the structures of $O_2$ and $O_3$, respectively. This experiment is to determine ontology mapping between $O_2$ and $O_3$ by Algorithm 4. Likewise, the authors applied $P@N$ criterion to measure the equality of the experiment.

Specifically, for vertex $v \in G_i$ ($i$=2 or 3) and given integer $N > 0$. Let $Sim_v^{N,\text{expert}}$ be the set of vertices determined by experts and it contains $N$ vertices having the most similarity of $v$ in different ontolog. Let

$$v_v^1 = \underset{v' \notin V(G_i)}{\text{argmin}} \{|f(v) - f(v')|\},$$

$$v_v^2 = \underset{v' \notin V(G_i), v' \neq v_1}{\text{argmin}} \{|f(v) - f(v')|\},$$

$$v_v^3 = \underset{v' \notin V(G_i), v' \neq v_1, v' \neq v_2}{\text{argmin}} \{|f(v) - f(v')|\},$$

$$\cdots$$

$$v_v^N = \underset{v' \in V(G_i), v' \neq v_1, \cdots, v' \neq v_{N-1}}{\text{argmin}} \{|f(v) - f(v')|\},$$

and

$$map(v) = \{v_v^1, v_v^2, \cdots, v_v^N\}.$$

Then the precision ratio for vertex $v$ is denoted by

$$\text{Pre}_v^N = \frac{|map(v) \cap Sim_v^{N,\text{expert}}|}{N}.$$

The $P@N$ average precision ratio for ontology graphs $G_2$ and $G_3$ is then stated as

$$\text{Pre}_{V(G_2) \cup V(G_3)}^N = \frac{\sum_{v \in V(G_2) \cup V(G_3)} \text{Pre}_v^N}{|V(G_2) \cup V(G_3)|}.$$

The ontology algorithms used in [40], [13] and [12] to humanoid robotics ontologies were employed. Then the precision ratios of these three methods were compared with that of the proposed algorithm. The experimental results are demonstrated in Table 2.

The better performance of our new algorithm can be fully illustrated by the results in Table 2, compared with those algorithms in [40], [13] and [12]. The advantage becomes more and more obvious when $N$ is increasing.

### C. Ontology similarity measure experiment on biology data

“GO” ontology $O_4$ constructed in http: //www. geneontology.org. was used in the experiment. Fig. 4 illustrates the basic structure of $O_4$. $P@N$ was employed to measure the equality of the experiment. Ontology algorithms were employed to “GO” ontology in [13], [12] and [41]. Then the authors compared the precision ratios obtained from four methods. Several experiment results are displayed in Table 3.

Clearly seen from Table 2, when $N$= 3, 5, 10 or 20, the precision ratio computed by adopting algorithm 3, the newly

TABLE II
THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

| | $P$@1 average precision ratio | $P$@3 average precision ratio | $P$@5 average precision ratio |
|---|---|---|---|
| Algorithm 4 in our paper | 0.2778 | 0.5185 | 0.7444 |
| Algorithm in [40] | 0.2778 | 0.4815 | 0.5444 |
| Algorithm in [13] | 0.2222 | 0.4074 | 0.4889 |
| Algorithm in [12] | 0.2778 | 0.4630 | 0.5333 |

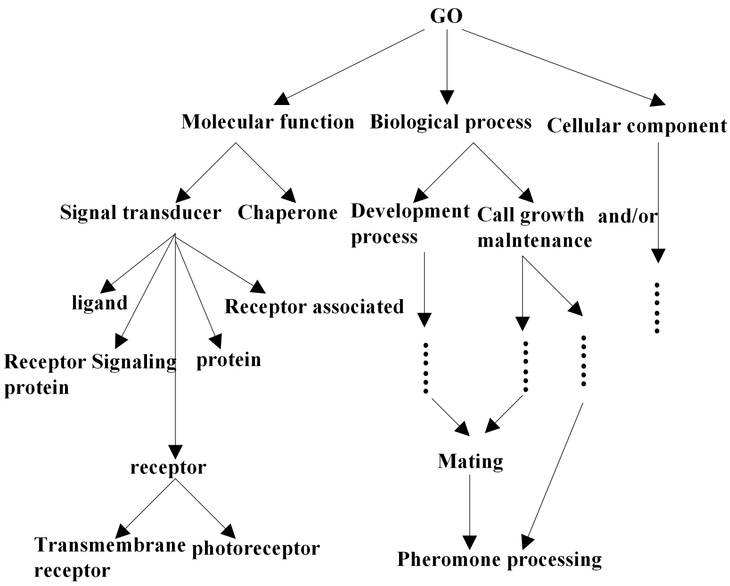


Fig. 4 The structure of "GO" ontology

TABLE III
THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

| | $P$@3 average precision ratio | $P$@5 average precision ratio | $P$@10 average precision ratio | $P$@20 average precision ratio |
|---|---|---|---|---|
| Algorithm 3 in our paper | 0.5071 | 0.6267 | 0.7643 | 0.8577 |
| Algorithm in [13] | 0.4638 | 0.5348 | 0.6234 | 0.7459 |
| Algorithm in [12] | 0.4356 | 0.4938 | 0.5647 | 0.7194 |
| Algorithm in [41] | 0.4213 | 0.5183 | 0.6019 | 0.7239 |

raised one, is higher than those from algorithms in [13], [12] and [41]. To be specific, all the ratios from algorithm 3, the new one, are above 0.5, which shows good efficiency and accuracy. In other word, it overshadows the previous algorithms in [13], [12] and [41] in efficiency.

### D. Ontology mapping experiment on physics education data

Physical education ontologies $O_5$ and $O_6$ were studied. The structures of $O_5$ and $O_6$ are displayed in Fig. 5 and Fig. 6 respectively. The experiment is to conduct ontology mapping between $O_5$ and $O_6$ by Algorithm 4. The equality of the experiment was tested according to $P@N$ criterion. After determining the closest $N$ concepts for each vertex on the ontology graph by experts, the first $N$ concepts for every vertex on ontology graph were obtained by utilizing the algorithm and computing the precision ratio. Ontology algorithms were also used in physical education ontology in [13], [12] and [14]. So, the authors compared the precision ratios computed by four methods. The experiment results are illustrated in Table 4.

Its obvious that when the number of $N$ is becoming larger, the ratios increase larger and more apparently with higher scores. That is, the advantage of our new algorithm becomes overwhelming, when $N$ is large enough and sufficiently. Therefore, the new algorithm proposed in this paper is superior to those proposed in [13], [12] and [14].

## VI. CONCLUSIONS

Ontology, as a widely used data structure representation and storage model, has drawn attention from the researchers. One popular ontology learning trick is mapping each vertex to a real number, and the similarity is judged by the difference between the real number which the vertices correspond to. With the background of big data, the complexity of ontology data is often large and the dimension of ontology vector is high. It inspires us to use the dimensionality reduction technologies in ontology similarity measuring and ontology mapping. In this paper, we raise an ontology learning model for ontology application. The basic idea of our ontology

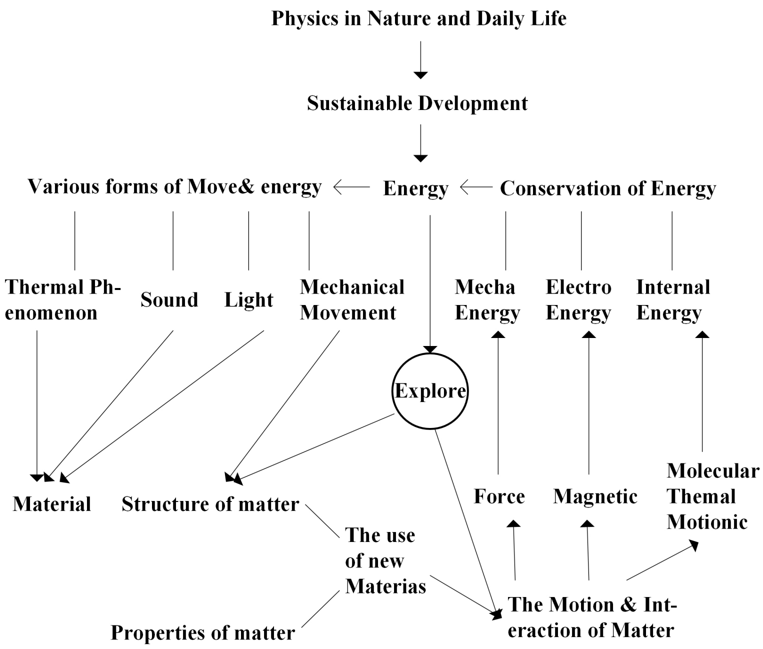


Fig. 5 "physics education"ontology $O_5$

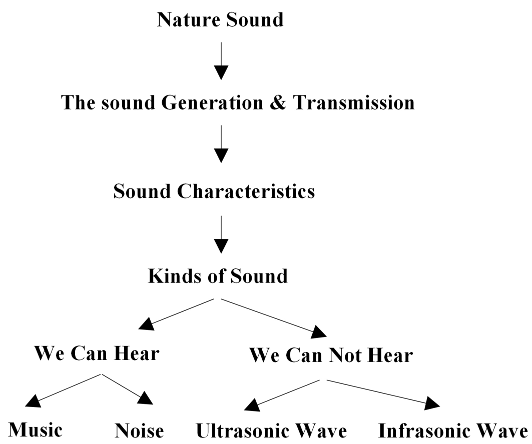


Fig. 6 "physics education"ontology $O_6$

TABLE IV
THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

| | $P@1$ average precision ratio | $P@3$ average precision ratio | $P@5$ average precision ratio |
|---|---|---|---|
| Algorithm 4 in our paper | 0.6774 | 0.8172 | 0.9161 |
| Algorithm in [13] | 0.6129 | 0.7312 | 0.7935 |
| Algorithm in [12] | 0.6913 | 0.7556 | 0.8452 |
| Algorithm in [14] | 0.6774 | 0.7742 | 0.8968 |

framework is based on the adjoint ontology graph technologies in which constructers form the index set of ontology vector. The tolerant framework, prize-collecting Steiner tree technologies and the Lagrangian relaxation approach are used as tools in order to get the solution of ontology optimization problem. The experiments show the effectiveness of adjoint ontology graph based ontology framework. The new technology contributes to the state of art for applications. The result gotten from experiments in our paper illustrates the promising application prospect.

## VII. CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests regarding the publication of this paper.

## ACKNOWLEDGMENT

We thank the reviewers for their constructive comments in improving the quality of this paper.

## References


[1] J. M. Przydzial, B. Bhhatarai, and A. Koleti, " GPCR ontology: development and application of a *G* protein-coupled receptor pharmacology knowledge framework," *Bioinformatics*, vol. 29, no. 24, pp. 3211-3219, 2013.

[2] S. Koehler, S. C. Doelken, and C. J. Mungall, " The human phenotype ontology project: linking molecular biology and disease through phenotype data," *Nucleic Acids Research*, vol. 42, no. D1, pp. 966-974, 2014.

[3] M. Ivanovic and Z. Budimac, " An overview of ontologies and data resources in medical domains," *Expert Systerms and Applications*, vol. 41, no. 11, pp. 5158-15166, 2014.

[4] A. Hristoskova, V. Sakkalis, and G. Zacharioudakis, " Ontology-driven monitoring of patient's vital signs enabling personalized medical detection and alert," *Sensors"*, vol. 14, no. 1, pp. 1598–1628, 2014.

[5] M. A. Kabir, J. Han, and J. Yu, " User-centric social context information management: an ontology-based approach and platform," *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1061–1083, 2014.

[6] Y. L. Ma, L. Liu, K. Lu, B. H. Jin, and X. J. Liu, " A graph derivation based approach for measuring and comparing structural semantics of ontologies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1039-1052, 2014.

[7] Z. Li, H. S. Guo, Y. S. Yuan, and L. B. Sun, " Ontology representation of online shopping customers knowledge in enterprise information," *Applied Mechanics and Materials*, vol. 483, pp. 603-606, 2014.

[8] R. Santodomingo, S. Rohjans, M. Uslar, J. A. Rodriguez-Mondejar, and M.A. Sanz-Bobi, " Ontology matching system for future energy smart grids," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 242-257, 2014.

[9] T. Pizzuti, G. Mirabelli, M. A. Sanz-Bobi, and F. Gomez-Gonzalez, " Food Track & Trace ontology for helping the food traceability control," *Journal of Food Engineering*, vol. 120, no. 1, pp. 17-30, 2014.

[10] N. Lasierra, A. Alesanco, and J. Garcia, " Designing an architecture for monitoring patients at home: Ontologies and web services for clinical and technical management integration," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 896-906, 2014.

[11] Y. Y. Wang, W. Gao, Y. G. Zhang and Y. Gao, " Ontology similarity computation use ranking learning method," *The 3rd International Conference on Computational Intelligence and Industrial Application*, Wuhan, China, pp. 20-22, 2010.

[12] W. Gao and L. Liang, " Ontology similarity measure by optimizing NDCG measure and application in physics education," *Future Communication, Computing, Control and Management*, vol. 142, pp. 415-421, 2011.

[13] X. Huang, T. W. Xu, W. Gao and Z. Y. Jia, " Ontology similarity measure and ontology mapping via fast ranking method," *International Journal of Applied Physics and Mathematics*, vol. 1, no. 1, pp. 54-59, 2011.

[14] W. Gao, Y. Gao, and L. Liang, " Diffusion and harmonic analysis on hypergraph and application in ontology similarity measure and ontology mapping," *Journal of Chemical and Pharmaceutical Research*, vol. 5, no. 9, pp. 592-598, 2013.

[15] W. Gao and T. W. Xu, " Stability analysis of learning algorithms for ontology similarity computation," *Abstr. Appl. Anal., Art. ID 174802, 9pages*, http://dx.doi.org/10.1155/2013/174802, 2013.

[16] W. Gao, L. L. Zhu, " Gradient learning algorithms for ontology computing," *Computational Intelligence and Neuroscience*, Article ID 438291, 12 pages, http://dx.doi.org/10.1155/2014/438291, 2014.

[17] W. Gao, L. L. Zhu and Y. Guo, "Multi-dividing infinite push ontology algorithm," Engineering Letters, vol. 23, no. 3, pp. 132-139, 2015.

[18] M. H. Lan, J. Xu, and W. Gao, "Ontology feature extraction via vector learning algorithm and applied to similarity measuring and ontology mapping," IAENG International Journal of Computer Science, vol. 43, no.1, pp. 10-19, 2016.

[19] M. H. Lan, J. Xu, and W. Gao, "Ontology similarity computation and ontology mapping using distance matrix learning approach" IAENG International Journal of Computer Science, vol. 45, no.1, pp. 164-176, 2018.

[20] L. L. Zhu, W. G. Tao, X. Z. Min and W. Gao, "Theoretical characteristics of ontology learning algorithm in multi-dividing setting," IAENG International Journal of Computer Science, vol. 43, no.2, pp. 184-191, 2016.

[21] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, " Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982-2001, 2010.

[22] M. A. Simplicio, L. H. Iwaya, B. M. Barros, T. C. M. B. Carvalho, M. Naslund, " SecourHealth: a delay-tolerant security framework for mobile health data collection," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 761-772, 2015.

[23] F. P. Guimaraes, P. Celestin, D. M. Batista, G. N. Rodrigues, A. C. M. A. de Melo, " A framework for adaptive fault-tolerant execution of workflows in the grid: empirical and theoretical analysis," *Journal of Grid Computing*, vol. 12, no. 1, pp. 127-151, 2014.

[24] D. Portugal, R. P. Rocha, " Distributed multi-robot patrol: a scalable and fault-tolerant framework," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1572-1587, 2013.

[25] F. M. Al-Turjman, A. E. Al-Fagih, W. M. Alsalih, H. S. Hassanein, " A delay-tolerant framework for integrated RSNs in IoT," *Computer Communications*, vol. 36, no. 9, pp. 998-1010, 2013.

[26] A. Elwhishi, P. H. Ho, K. Naik, B. Shihada, " A novel message scheduling framework for delay tolerant networks routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 871-880, 2013.

[27] M. Leitner, I. Ljubic, M. Sinnl, " A computational study of exact approaches for the bi-objective prize-collecting steiner tree problem," *Infroms Journal on Computing*, vol. 27, no. 1, pp. 118-134, 2015.

[28] E. Alvarez-Miranda, A. Candia-Vejar, X. J. Chen, X. D. Hu, B. Li, " Risk models for the prize collecting steiner tree problems with interval data," *Acta Mathematicae Applicatae Sinica-English Series*, vol. 30, no. 1, pp. 1-26, 2014.

[29] E. Alvarez-Miranda, I. Ljubic, P. Toth, " Exact approaches for solving robust prize-collecting Steiner tree problems," *European Journal of Operational Research*, vol. 229, no. 3, pp. 599-612, 2013.

[30] M. Haouari, S. B. Layeb, H. D. Sherali, " Algorithmic expedients for the prize collecting steiner tree problem," *Discrete Optimization*, vol. 7, no. 1-2, pp. 32-47, 2010.

[31] M. Haouari, S. B. Layeb, H. D. Sherali, " The prize collecting steiner tree problem: models and lagrangian dual optimization approaches," *Computational Optimization and Applications*, vol. 40, no. 1, pp. 13-39, 2008.

[32] A. Diabat, O. Battia, D. Nazzal, " An improved Lagrangian relaxation-based heuristic for a joint location-inventory problem," *Computers & Operations Research*, vol. 61, pp. 170-178, 2015.

[33] M. E. Keskin, I. K. Altinel, N. Aras, " Combining simulated annealing with Lagrangian relaxation and weighted Dantzig-Wolfe decomposition for integrated design decisions in wireless sensor networks," *Computers & Operations Research*, vol. 59, pp. 132-143, 2015.

[34] A. Diabat, P. J. Richard, " An integrated supply chain problem: a nested lagrangian relaxation approach," *Annals of Operations Research*, vol. 229, no. 1, pp. 303-323, 2015.

[35] M. A. Bragin, P. B. Luh, J. H. Yan, N. P. Yu, G. A. Stern, " Convergence of the surrogate Lagrangian relaxation method," *Journal of Optimization Theory and Applications*, vol. 164, no. 1, pp. 173-201, 2015.

[36] T. Logenthiran, W. L. Woo,T. P. Van, " Lagrangian relaxation hybrid with evolutionary algorithm for short-term generation scheduling," *International Journal of Electrical Power & Energy Sgystems*, vol. 64, pp. 356-364, 2015.

[37] X. Liu, M. L. Song, D. C. Tao, Z. C. Liu, L. M. Zhang, C. Chen, J. J. Bu, " Random forest construction with robust semisupervised Node splitting," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 471-483, 2015.

[38] M. X. Goemans, D. P. Williamson, " A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 296-317, 1995.

[39] N. Craswell and D. Hawking, " Overview of the TREC 2003 web track," *In proceedings of the Twelfth Text Retrieval Conference, Gaithersburg, Maryland, NIST Special Publication*, pp. 78-92, 2003.

[40] W. Gao and M. H. Lan, " Ontology mapping algorithm based on ranking learning method," *Microelectronics and Computer*, vol. 28, no. 9, pp. 59-61, 2011.

[41] Y. Gao and W. Gao, " Ontology similarity measure and ontology mapping via learning optimization similarity function," *International Journal of Machine Learning and Computing*, vol. 2, no. 2, pp. 107-112, 2012.



**Jiali Tang** , male, was born in the city of Luo Yang, Henan Province, China on May 10, 1980. He got Bachelors Degree on Material Science and Technology from Nanjing University of Aeronautics and Astronautics in 2002. He got PhD Degree on Mechatronic Engineering from Jiangsu University in 2016. Now, he acts as vice professor in the School of computer engineering, Jiangsu University of Technology. As a researcher in computer science, his interests are covered two disciplines: Computer Automatic Control and Machine Vision.

**Yu Pan**, male, was born in the city of Huang Shan, Anhui Province, China on March 31, 1963. He got Bachelor's Degree on Computer and

Application from China University of Mining and Technology in 1985. He got Masters Degree on Mining Electrification and Automation from Beijing Graduate School of China University of Mining and Technology in 1990. Now, he acts as professor in the School of computer engineering, Jiangsu University of Technology. As a researcher in computer science, his interests are covered two disciplines: Computer Network and Information Security.

**Zhihong Wang**, Female, was born in the city of Tai An, Shandong Province, China on Oct 8,1978.She got Bachelors Degree on Computer Science and Technology from Shandong University in 2007.Now,she acts as vice professor in Shandong Ying Cai University. As a researcher in computer science, her interests: machine learning and information hiding.

**Linli Zhu**, male, was born in the city of Badong, Hubei Province, China on Sep. 20, 1975. He got Master degrees on computer software and theory from Yunnan normal university in 2007. Now, he acts as associate professor in the department of computer engineering, Jiangsu University of Technology. As a researcher in computer science, his interests are covered two disciplines: computer network and artificial intelligence.