

Exploration of Learning Service Discovery Algorithm based on Ontology

<https://doi.org/10.3991/ijet.v13i11.9607>

Xuhong Peng

South China Normal University, Guangzhou, China

evanina11@163.com

Abstract—To study the learning service discovery algorithm based on ontology, the introduction of bipartite graph theory is focused on. Based on the undergraduate technology, a learning service discovery algorithm based on the basic bipartite graph theory (eLSDA-BG) is proposed. The algorithm mainly converts the problem of learning service match into the optimal complete match problem of bipartite graphs. Through the optimal complete match problem, the learning service match is realized and the learning service discovery is finally reached. The eLSDA-BG algorithm is compared with the greedy algorithm, the UDDI (Universal Description, Discovery, and Integration) algorithm, and OWL-S (Web Ontology Language for Service) algorithm. The comparison of results shows that the eLSDA-BG algorithm proposed can improve recall ratio, accuracy rate and efficiency of learning service discovery in situation of certain sample size.

Keywords—bipartite graph, learning services, ontology

1 Introduction

With the application of modern educational technology in the field of modern distance education, the teaching mode and teaching research focus of modern distance education are changing. The continuous development of technology and related theories of computer supported collaborative/ cooperative learning (CSCL) further indicates that collaborative learning can enhance the interaction between learners, strengthen the awareness of learner autonomy and self-exploration, and promote the learners to analyze and solve problems. As a result, it is gradually becoming the main learning mode of learners in modern distance education. How to provide the necessary technical support for web-based learners has become a key factor affecting the quality of modern distance education. In modern distance education, education mode is changing, individualization, autonomy and collaborative learning have gradually become the goals pursued by network educators and learners, such as the automatic provision of learning resources, the automatic generation of personalized learning schemes, automatic discovery of the learning service (e-Learning Service), and self-evaluation of learning effects. How to find out the learning service needed in the teaching process quickly and accurately is the key to affect the teaching effect. Therefore, the ontology-based learning service discovery algorithm is studied.

The traditional discovery mechanism used in learning service discovery based on universal description, discovery and integration (UDDI) is only limited to keyword match, which is a static match method. Although the search speed is faster, the degree of automation is not high, and it cannot guarantee all the learning services that meet the requirements can be found. OWL-based ontology technology is applied to service description, which makes service description have semantic information. The Service Profile in OWL-S ontology describes the attributes of services: various functional attributes and non-functional attributes. Service discovery based on OWL-S can well overcome the weakness of UDDI match and improve the quality of service discovery.

The match of Web service is studied, but there are only 4 kinds of service match in the algorithm, and only the greedy algorithm is used in the match process so that the recall and precision of service discovery are not very high. The e-learning service discovery algorithm - bipartite graph (eLSDA-BG) is designed by combining the ontology technology and the bipartite graph.

2 Literature review

Low efficiency is a serious problem of learning service discovery in CSCL system at the present stage. The main manifestations are as follows: first of all, the existing discovery mechanism does not standardize the ontology description of learning services, neither supports the query of the CSCL tools attributes. It cannot describe the service ability properly, and the effect of learning service cannot meet the needs of the teachers. Moreover, the Web service discovery mechanism based on UDDI only provides a simple keyword-based search method, and lacks support for semantic reasoning.

Wu et al. (2016) studied the concept of user satisfaction, and proposed a e-Learning service discovery algorithm based on user satisfaction [1]. Mohammadi (2015) profoundly studied the collaborative problem with computer as a tool. The research results supported the learning of intelligent architecture design. And he developed an application system DomoSim-TPC, which was applied to various universities and teaching institutions in Spain, but the learning service discovery algorithm was not realized [2]. Chew and Rahim (2017) proposed the semantic discovery of Web service based on social learning, and realized the prototype of UDDI, and used the real data set of OWL-S technology diagram (OWLS-TC) to simulate. In the proposed structure, 67% noise responses from the syntactic search (N-gram string distance algorithm) were reduced. The results also illustrated the conceptual learning capability of the proposed architecture, and the significant improvement in service discovery after the contribution of some social concepts [3].

Suraci et al. (2015) studied the semantic-based Web service discovery algorithm, and proposed a fast service discovery algorithm combined with genetic algorithm. In the process of interaction between mobile agent and service node, the value of each service record was dynamically changed according to the results of the access. As a result, effective information could retain a longer time in the network, and outdated and erroneous service information would disappear quickly in the network. The simulation results show that the mobile agent could find the required service resources quickly in

the distributed network, but the algorithm was not achieved in the learning service discovery system [4]. Navimipour and Zareie (2015) explored the target Web service discovery ontology that used the tree data structure to describe the Web service and to give all nodes a weight value through a certain strategy, and then calculate the semantic similarity between the requested Web service and the registered service. In order to verify the feasibility and effectiveness of the algorithm, a prototype system was developed to demonstrate its working principle. The experiment results showed that the algorithm had high recall and precision [5]. Seghir et al. (2015) discussed the mapping Web service modeling ontology (WSMO) enhanced semantic Web service discovery based on machine learning. In order to make use of a large number of existing Web services libraries, the mapping from web service description language (WSDL) to OWL-S was created. However, OWL-S was only an extension of OWL, and it needed additional support to exert its best semantic capability. Another better ontology knowledge representation model is WSMO, which has a semantic framework for processing Web services, but lacks the extensibility of OWL-S. Therefore, a model that maps WSDL to WSMO was proposed [6]. Buvana and Suganthi (2015) developed a highly reliable and secure self-learning solution based on service. This kind of learning service is an interactive model between learners and learning services [7]. Lara et al. studied the learning service discovery model, constructed a cluster promoting Web service discovery model based on supervised item weighting and adaptive metric learning, and proved the feasibility of the model through the experimental results implemented on the real world Web service set [8].

To sum up, learning service is the key factor affecting the quality and effect of modern teaching. However, the traditional ontology-based learning service algorithm is still in a static match way, and it cannot guarantee all the learning services that meet the needs found. The new learning service discovery algorithm adopts the optimal match mode, which has high automation level and higher quality of service. It can quickly discover the required service resources, and also has high recall and precision. In order to solve the problems and shortcomings in the traditional learning service algorithm, through the study and analysis of existing related literature, a learning service discovery algorithm is proposed and designed based on the ontology technology by combining with bipartite graph theory.

3 Method

3.1 Basic concepts of ontology and learning services

The ancient Greek philosopher Aristotle gave the definition of ontology in philosophy: "ontology is a systematic description of objects in the world, that is, the theory of existence", the systematic explanation or illustration of objective existence, and the abstract essence of objective reality.

Learning service is a special semantic Web service. The domain covered by learning service ontology is mainly the field of modern distance education.

Learning service is a special semantic Web service. Besides the features of semantic Web service, there are some characteristics as follows: first, the class of learning services is limited. There are only 4 kinds of pop learning services: mail service, conference service, monitoring service and index search service. Then, the curriculum feature of the learning service is strong. Learning different contents (knowledge) needs different learning services, and even the same learning content has many learning services. There are many different ways of communication between learners with different subjects. For instance, there are lots of BBS of the same theme *Computer System Structure*, such as the communication garden of *Computer System Structure* of Zhang Chenxi, and the learning and discussion area of *Computer System Structure* for computer application. In addition, the granular diversity of learning services is also discussed. The granularity of learning services is not uniform. As far as BBS is concerned, some discussion areas are very pertinent, only for a certain point of knowledge, and some of the discussion areas are aimed at a certain course, even in a particular major and in a certain field. The long-term availability of learning services is explored. In terms of the cost of learning services, some learning services require the user to fill in the registration information, and the computer cannot complete it automatically. This requires the learner to register an account with the service provider and have its own independent username and password to use the service.

3.2 Bipartite graph theory

Definition 1: undirected graph $G = (V, E)$, where V is a set of vertices, and E is an edge set. If the graph meets $V = V_0 \cup V_1, V_0 \cap V_1 = \varnothing$ and in $\forall e = (x, y) \in E$, there are $x \in V_0$ and $y \in V_1$, then the graph G is called bipartite graph. If for any two points u and v , $u \in V_0$ and $v \in V_1$ have edges connected, then the bipartite graph $G = (V, E)$ is called complete bipartite graph, recorded as $K_{m,n}$, where $m = |V_0|$ and $n = |V_1|$, indicating the potential of vertex sets V_0 and V_1 , respectively.

As shown in Figure 1 (a), it is a bipartite graph $G = (V, E)$, $V = \{x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5\}$, and $E = \{(x_1, y_1), (x_1, y_5), (x_2, y_1), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_5)\}$. Because $V_0 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $V_1 = \{y_1, y_2, y_3, y_4, y_5\}$, obviously $V = V_0 \cup V_1$, $V_0 \cap V_1 = \varnothing$ and $\forall e = (x, y) \in E$ has $x \in V_0$ and $y \in V_1$.

Definition 2: if the graph $G' = (V, E')$ is a sub-graph of bipartite graph $G = (V, E)$, $E' \subseteq E$, and there is no common vertex for e_1 and e_2 in terms of $\forall e_1, e_2 \in E'$, then G' is called a match of G . The number of edges in the matching G' is called the cardinal number of the match G' . The match with the largest cardinality is called the largest cardinality match, referred to as maximal match.

As shown in Figure 1 (b), it is a match of the bipartite graph G shown in Figure 1 (a). The graph G'' shown in Figure 1 (c) is a maximal match of the bipartite graph G shown in Figure 1 (a). Obviously, the match and maximal matching of a bipartite graph are not necessarily unique. As shown in Figure 1 (a), (b) and (c), they are bipartite graph G_1 and their complete match G_1' and G_1'' , respectively.

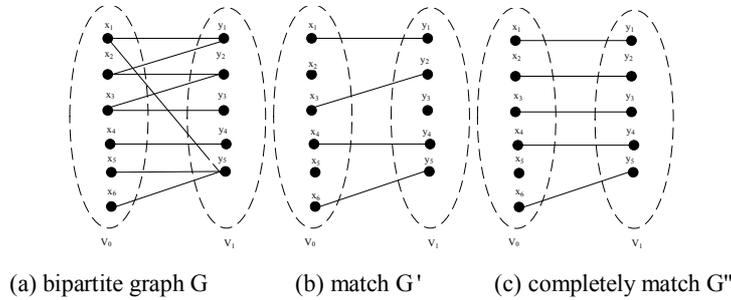


Fig. 1. Bipartite graph G , match and the maximal match

Definition 3: set G' to be a match of graph G . If G' covers all the vertexes in G , then G' is called a complete match of graph G .

Definition 4: set $G' = (V, E')$ to be a match of bipartite graph $G = (V, E)$, $v \in V$. If there are edges in G' with v as the endpoint, then v is called G' -saturated point; otherwise, v is called G' -unsaturated point.

Theorem 1: supposing that $G' = (V, E')$ is a match of bipartite graph $G = (V, E)$ and each node in G is G' -saturated point, then G' is complete match.

Proof: anti-evidence.

Assuming that G' is not complete match, then there must be a node $v_x \notin V$, that is, the node is G' -unsaturated point, then this is contradictory with that each node in G is G' -saturated point, so G' is complete match.

Definition 5: supposing that $G' = (V, E')$ is a match of bipartite $G = (V, E)$, if there is a simple path alternately composed of edges in G' and $G - G'$, then the simple path is called G' -alternating path of G . if the starting point and end point of G' -alternating path coincide and form the circuit, then the circuit is called G' -alternating circuit. If the starting point and end point of G' - alternating path are G' -unsaturated points, then this alternating path is called G' - augmenting path.

For the bipartite graph G_1 and its complete match G_1' in Figure 1 (a), $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$ in Figure 2 (a) is a G_1' -alternating path, and $(x_1, y_1, x_2, y_2, x_4, y_4, x_3, y_3, x_1)$ in Figure 2 (b) is G_1' -alternating circuit.

Theorem 2: assuming that $G = (V_0 + V_1, E)$ is a bipartite graph, then there is complete match from V_0 to V_1 in G . When there is $|N(S)| \geq |S|$ for any $S \subseteq V_1$, $N(S)$ indicates all the nodes adjacent to nodes in S :

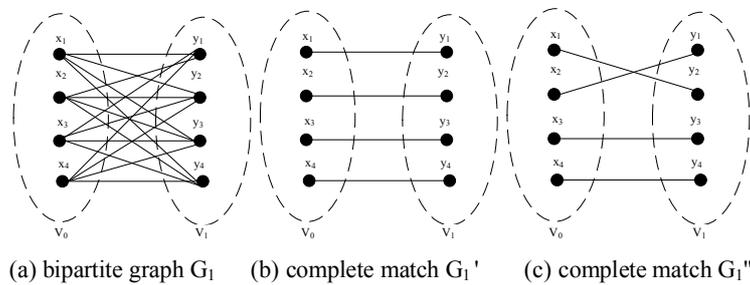


Fig. 2. Bipartite graph G_1 and its complete match G_1' and G_1''

3.3 Construction of learning service bipartite graph

Definition 6: supposing that there is bipartite graph $G = (V, E)$, if there is always a weight w for $\forall e \in E$, then this bipartite graph G is called weighting bipartite graph.

Definition 7: for learning service bipartite graph G_{eLS} and complete match G'_{eLS} , let $\max(w_i)$ to be the edge with the maximum weight in G'_{eLS} , then this edge indicates the poorest match degree between two vertex points in complete match G'_{eLS} , and $\max(w_i)$ is called entire match degree of G'_{eLS} , recorded as:

$$EMD(G'_{eLS}) = \max(w_i), i = 1, 2, 3, 4. \quad (1)$$

A complete match corresponds to the match of each attribute in the request learning service attribute set P_R with some attributes in the publish learning service attribute set P_A . If there is no complete match, it shows that the request learning service does not match with the publish learning service; if there are multiple complete matches, the optimal complete match is selected from them.

As shown in Figure 3, learning service bipartite graph G_{eLS} has two complete match G'_{eLS} and G''_{eLS} . The weights of three edges of G'_{eLS} are w_1, w_2 , and w_4 , respectively, so the entire match degree is $EMD(G'_{eLS}) = \max(w_1, w_2, w_4)$, that is w_4 . The weights of three edges of G''_{eLS} are w_3, w_3 , and w_3 , so the entire match degree is $EMD(G''_{eLS}) = \max(w_3, w_3, w_3)$, that is w_3 . $w_3 < w_4$, so G''_{eLS} is selected as the optimal match rather than selecting G'_{eLS} .

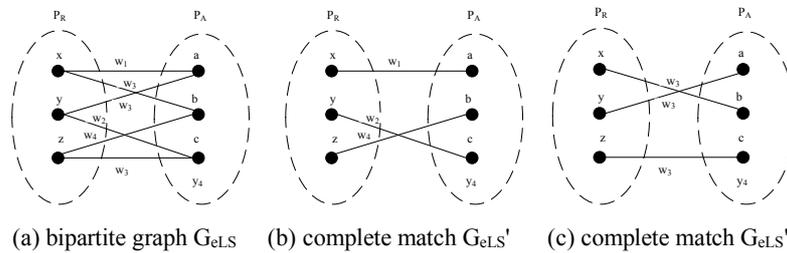


Fig. 3. Learning service bipartite graph G and its complete match G'_{eLS} and G''_{eLS}

3.4 Learning service match algorithm based on bipartite graph

The first is the design of bipartite graph weights.

The Hungarian algorithm calculates the complete matching of bipartite graph by calculating the minimum of the sum $\sum w_i$ of the weights of all sides in the bipartite graph, which can get the result in polynomial time. $|V|$ is recorded as the number of the vertices of the bipartite graph, then the time complexity of the Hungarian algorithm is $O(|V|^3)$. Unlike the Hungarian algorithm, the eLSDA-BG algorithm wants to compute the matching of $EMD(G'_{eLS})$ minimization. As shown in Figure 3, G'_{eLS} and G''_{eLS} are the two matches of G_{eLS} . If $w_1=1, w_2=2, w_3=3, w_4=4$, it can be calculated according to Table 1.

Table 1. eLSDA-BG and Hungarian algorithm

Complete match	EMD (G'_{eLS})	$\sum w_i$
G'	4	7
G''	3	9

G''_{eLS} is selected as the optimal match, and the Hungarian algorithm uses G'_{eLS} as the optimal match. As a result, the Hungary algorithm cannot be directly used to calculate the matching expected. And because the minimized match of $\sum w_i$ is equivalent to the match of $EMD(G'_{eLS})$ minimization, the Hungarian algorithm can be used to calculate the optimal matching wanted. In the eLSDA-BG algorithm, the computation of the edge weight of the bipartite graph of learning service is shown in Table 2.

Table 2. The edge weight calculation of learning service bipartite graph in eLSDA-BG algorithm

Match Degree (a, b)	Weight
Accurate match	$w_1 = 1$
Pluggable match	$w_2 = (w_1 * P_A) + 1$
Inclusion match	$w_3 = (w_2 * P_A) + 1$
Compatibility match	$w_4 = (w_3 * P_A) + 1$

There are 5 kinds of matching results between the request service attribute of the learning service and the attribute of the publishing service, which are complete match, pluggable match, inclusion match, compatibility match and mismatch. Also because, according to definition 6, if the two attribute does not match, then, in the learning service bipartite graph $G_{eLS} = (P_R + P_A, E)$, there is no edge between the corresponding two points, and the learner acceptance for the rest 4 kinds of matching effects is: accurate match > pluggable match > inclusion match > compatibility match. Therefore, the calculation of the weight of the edges can be designed as shown in Table 2.

Then, a learning service matching algorithm based on bipartite graph is designed.

Assuming that the number of publish learning service is 1, then the attribute set of learner's request learning service u_R is P_R , $P_R = P_R^{out} \cup P_R^{in}$. The function of algorithm eLSDA-BG is to compare all the input attributes and output attributes of the request learning service with the related attributes of the publish learning service in the library one by one. If the match degree is not Fail, the publish learning service is added to the result set. Finally, the result set after sorting is returned to the learning service request.

Algorithm name: eLSDA-BG.

Input: attribute set P_R of request learning service U_R .

Output: sub-set U_{sub} of publish learning service.

The detailed algorithm flow is shown in Figure 4.

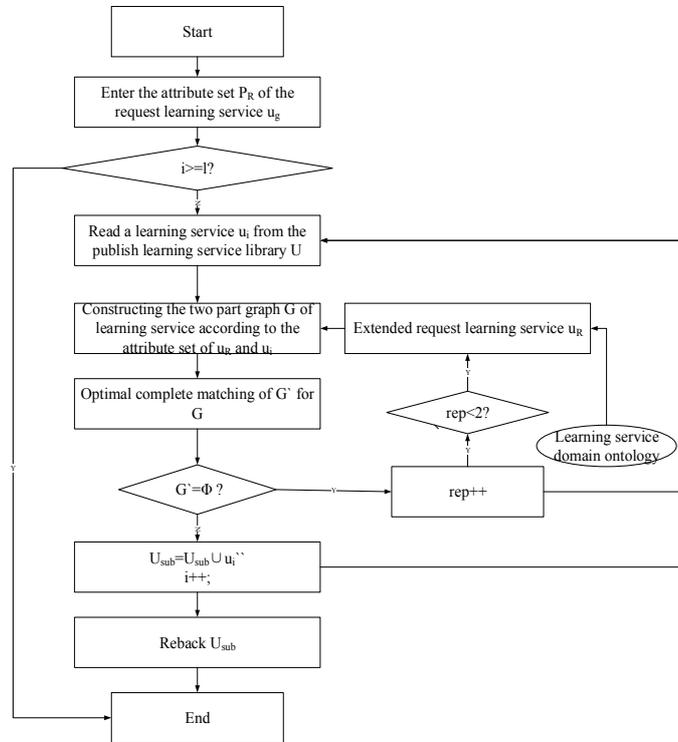


Fig. 4. Learning service match algorithm based on bipartite graph

The time complexity of the algorithm is the basic statement - the if-else statement. This basic statement is nested in a for cycle, and the number of cycles of the for cycle is the number l of publish learning services. The cycle needs to build learning service bipartite graph, to find the optimal complete match. According to the analysis of complete match calculation, let m and n to be the average attribute number of a request learning service and the average attribute number of a publish learning service, so the time complexity of the algorithm eLSDA-BG is $O(lmn)$. The main storage variables of the algorithm are an attribute set of a request learning service and an attribute set of a learning service and a matching result. The storage space occupied by the three is a constant level, so the spatial complexity of the algorithm is $O(1)$.

Finally, it is the calculation of complete match.

The algorithm eLSDA-BG calls a function Complete Match that calculates the complete match of bipartite graph. In accordance with the attribute set P_R of request learning service and the attribute set P_A of publish learning service, the function constructs a bipartite graph, and calls the Hungarian algorithm to find the optimal complete match of the bipartite graph.

The number of request learning service attribute is $|P_R| = m$, the number of publish learning service attribute is $|P_A| = n$, the basic statement of the algorithm Complete Match is $\text{switch}(\text{Degree}(P_{A_i}, P_{R_j}))$, and the time complexity of the algorithm is obviously $O(mn)$.

3.5 Experiment environment

The basic hardware environment of the computer used in this experiment is: PIII 2.6G CPU, 512M memory computer. The system is loaded with 9 ontologies, a total of 3697 attributes, and 415 publish learning services. In the above environment, two groups of tests are carried out on the greedy matching algorithm proposed in the eLSDA-BG algorithm, and the calculation of the match degree of the two algorithms both uses the algorithm Degree. The learning service discovery model is shown in Figure 5.

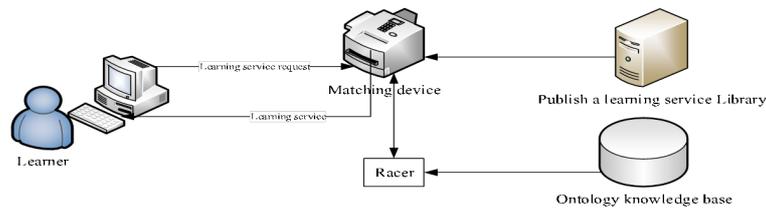


Fig. 5. Learning service discovery model

4 Results and discussion

4.1 Experiment 1 and result analysis

In the learning service discovery model shown in Figure 5, the greedy algorithm and the eLSDA-BG two algorithms are used, respectively. Table 3 is the comparison of the recall and precision of the experimental results.

Table 3. The recall and precision of two learning service discovery algorithms

Algorithms	Match results	Match degree						Recall	Precision
		Accurate	Plug-gable	Inclusion	Compatibility	Not match	Entire match		
The first group	Greedy algorithm	1	0	0	4	1375	1380	100%	20%
	eLSDA-BG	1	0	0	0	1379	1380	100%	100%
The second group	Greedy algorithm	0	0	3	0	2487	2490	50%	100%
	eLSDA-BG	0	0	6	0	2484	2490	100%	100%

The first group: the test is to find 1 exact match in 1380 learning services. The experimental data show that the recall rate of both the greedy algorithm and the eLSDA-BG algorithm is 100%. The greedy algorithm produces 4 false affirmative, resulting in the precision reduced greatly, only 20%. The reason is that the greedy algorithm of this group does not delete the attributes from the candidate list.

The second group: the greedy algorithm in this group deletes attributes from the candidate list. 6 request learning services are constructed by ontology, and since 6 request learning services will be searched in 415 publish learning services, there will be $6 \times 415 = 2490$ matches. Ideally, all 6 request learning services can find the corresponding publish learning services. The actual results are shown as follows: the precision of the greedy algorithm and the eLSDA-BG algorithm is 100%, and the eLSDA-BG algorithm matches all 6 request learning services. The greedy method produces 3 error negation, resulting in the recall rate of only 50%.

Figure 6 shows the time cost of the learning service match of the two algorithms in this experiment. The match time of eLSDA-BG algorithm is about 10% higher than that of greedy algorithm, which is within acceptable range. The match time of the two is linear, because the time complexity of the two algorithms is $O(|V_0|)$.

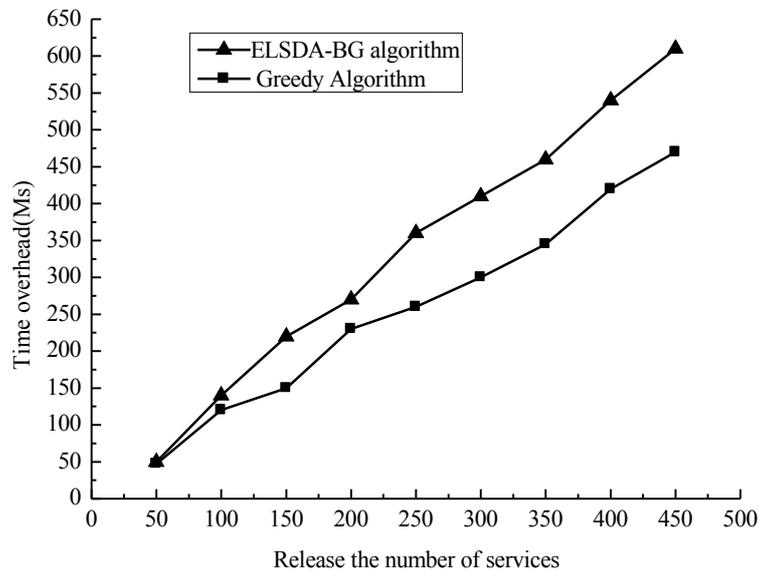


Fig. 6. Time cost of learning service match

4.2 Experiment 2 and result analysis

Compared with the greedy algorithm, the eLSDA-BG algorithm can maintain a high recall and precision, but compared with the UDDI algorithm and the OWL-S algorithm, the effect of eLSDA-BG algorithm is not known. Moreover, the efficiency compared in Experiment 1 is the efficiency of learning service matching, rather than the efficiency of learning service discovery. In the process of learning service discovery, there are steps of construction of learning service bipartite graph and the conversion of learning service results. As a result, experiment 2 is made. 15000 learning services are registered in the system and their learning service match results are compared with those of the UDDI algorithm and the OWL-S algorithm.

Table 4 shows the comparison of the precision and recall among UDDI algorithm, OWL-S algorithm and eLSDA-BG algorithm. It can be seen from the table that when the number of learning services is less than 100, the precision of the eLSDA-BG algorithm is the same as that of the OWL-S algorithm, both of which are 100%. This shows that the eLSDA-BG algorithm has a good effect when the number of learning services is small. When the number of learning services is more than 1000, the precision of the eLSDA-BG algorithm is reduced. Compared with the OWL-S algorithm, the efficiency is 2.4% to 12.8% lower, but still remains above 80%, far higher than 29.5% to 47.3% of the UDDI algorithm. As for the recall rate of learning services, the eLSDA-BG algorithm always keeps 100%, which shows that as long as there are related learning services in the publish learning service library, they can always be found.

Table 4. Assessment of precision and recall of learning service discovery

The number of publish learning service	UDDI algorithm		OWL-S algorithm		eLSDA-BG algorithm	
	Precision	Recall	Precision	Recall	Precision	Recall
50	54.2%	100%	100%	100%	100%	100%
100	62.8%	73.2%	100%	100%	100%	100%
1000	47.3%	54.8%	95.1%	100%	92.7%	100%
10000	34.4%	39.7%	94.7%	100%	86.6%	100%
20000	29.5%	35.6%	93.5%	100%	81.6%	100%

Figure 7 is the efficiency comparison of UDDI algorithm, OWL-S algorithm and eLSDA-BG algorithm.

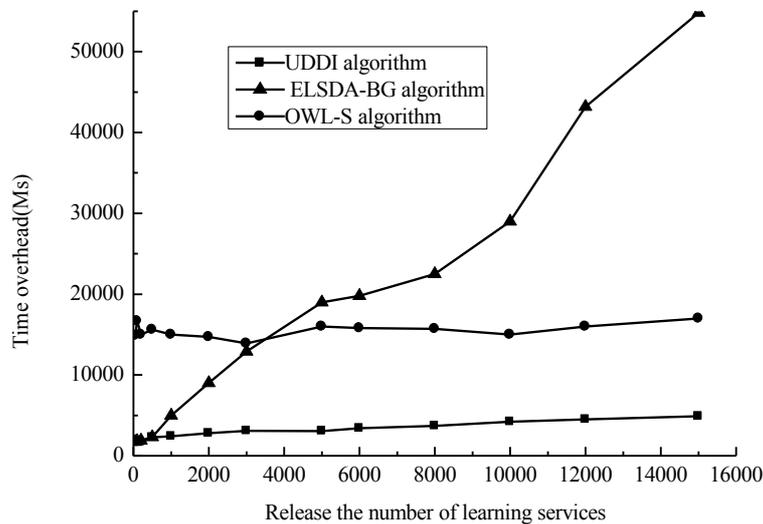


Fig. 7. Time cost of learning service discovery

It can be seen from Figure 7 that, the time cost of UDDI discovery algorithm has been maintained at a low level because UDDI is a static match based on the keyword and it has a higher match efficiency. The time cost of OWL-S algorithm has been kept at a moderate level, because the algorithm needs inference by inference engine. At the same time, it is found that with the increase of the number of learning services, the time cost of the eLSDA-BG algorithm increases rapidly. The reason is that the algorithm is matched based on bipartite graph. For each learning service match, a learning service bipartite graph is built. With the increase of the number of publish learning service, the learning service bipartite graph built is also increasing linearly. In the meanwhile, the number of learning service bipartite graph built is also multiplied increasing, resulting in a rapid increase in the time cost of the learning service discovery.

Compared with the traditional UDDI algorithm, the recall and precision of learning service discovery are greatly improved. When the number of learning services is less than 50, the efficiency of learning service discovery is at the same level as that of the UDDI algorithm. Compared with the OWL-S algorithm, the recall of the two algorithms is maintained at a high level of 100%. However, the precision of the algorithm is lower when the number of learning services is greater than 100, but it is still higher than that of the UDDI algorithm. When the number of learning services is less than 3000, the efficiency of the algorithm is higher than that of learning service discovery algorithm based on OWL-S.

5 Conclusion

In order to improve the recall, precision and efficiency of learning service discovery, a learning service discovery algorithm based on bipartite graph - eLSDA-BG is designed in accordance with the characteristics of the bipartite graph. This algorithm takes the attribute set of the request learning service and the publish learning service as the bipartite graph vertex sets. The line between all the match attributes is the edge, and the edge's weight is the attribute match degree. The problem of learning service matching is converted to the optimal complete match problem of the bipartite graph. Finally, two sets of experiments are made, and the algorithm is compared with greedy algorithm, UDDI algorithm, and OWL-S algorithm.

6 References

- [1] Wu, D., Yan, J., Wang, H., & Wang, R. (2016). Service demand discovery mechanism for mobile social networks. *Sensors*, 16(11): 1982. <https://doi.org/10.3390/s16111982>
- [2] Mohammadi, H. (2015). Investigating users' perspectives on e-learning: An integration of TAM and IS success model. *Computers in Human Behavior*, 45: 359-374. <https://doi.org/10.1016/j.chb.2014.07.044>
- [3] Chew, B. L., & Rahim, M. A. (2017). Integration of EBSCO Discovery Service widget into the learning spaces of LMS: A case study of Wawasan Open University. *Asian Association of Open Universities Journal*, 12(2): 137-153. <https://doi.org/10.1108/AAOUJ-01-2017-0013>

- [4] Suraci, V., Giorgi, C. G., Battilotti, S., & Facchinei, F. (2015). Distributed workload control for federated service discovery. *Procedia Computer Science*, 56: 233-241. <https://doi.org/10.1016/j.procs.2015.07.221>
- [5] Navimipour, N. J., & Zareie, B. (2015). A model for assessing the impact of e-learning systems on employees' satisfaction. *Computers in Human Behavior*, 53: 475-485. <https://doi.org/10.1016/j.chb.2015.07.026>
- [6] Seghir, N. B., Kazar, O., & Rezeg, K. (2015). A decentralized framework for semantic web services discovery using mobile agent. *International Journal of Information Technology and Web Engineering (IJITWE)*, 10(4): 20-43. <https://doi.org/10.4018/IJITWE.2015100102>
- [7] Buvana, M., & Suganthi, M. (2015). An efficient cluster based service discovery model for mobile ad hoc network. *KSI Transactions on Internet and Information Systems (TIIS)*, 9(2): 680-699. <https://doi.org/10.3837/tiis.2015.02.011>
- [8] Lara, J. A., Lizcano, D., Martínez, M. A., Pazos, J., & Riera, T. (2014). A system for knowledge discovery in e-learning environments within the European Higher Education Area—Application to student data from Open University of Madrid, UDIMA. *Computers & Education*, 72: 23-36. <https://doi.org/10.1016/j.compedu.2013.10.009>

7 Author

Xuhong Peng is with South China Normal University, Guangzhou, China (evaninall@163.com).

Article submitted 13 September 2018. Resubmitted 23 October 2018. Final acceptance 27 October 2018. Final version published as submitted by the author.