

# Article

---

## Evolutionary coincidence-based ontology mapping extraction

Vahed Qazvinian, Hassan Abolhassani, Seyed H. Haeri (Hossein) and Babak Bagheri Hariri

Web Intelligent Lab, Department of Computer Engineering, Sharif University of Technology and School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran

E-mail: qazvinian@ce.sharif.edu, abolhassani@sharif.edu, shhaeri@ce.sharif.edu, hariri@ce.sharif.edu

**Abstract:** *Ontology matching is a process for selection of a good alignment across entities of two (or more) ontologies. This can be viewed as a two-phase process of (1) applying a similarity measure to find the correspondence of each pair of entities from two ontologies, and (2) extraction of an optimal or near optimal mapping. This paper is focused on the second phase and introduces our evolutionary approach for that. To be able to do so, we need a mechanism to score different possible mappings. Our solution is a weighting mechanism named coincidence-based weighting. A genetic algorithm is then introduced to create better mappings in successive iterations. We will explain how we code a mapping as well as our crossover and mutation functions. Evaluation of the algorithm is shown and discussed.*

**Keywords:** ontology alignment, mapping extraction, coincidence-based scoring, genetic algorithm

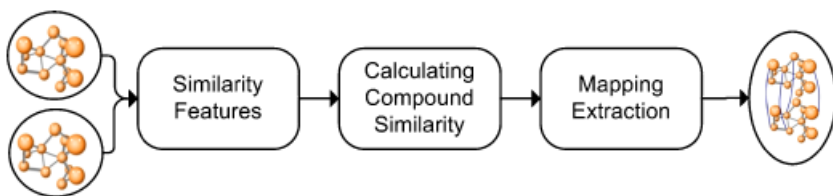
### 1. Introduction

The semantic web is rather a new concept, and is defined so that agents will be able to understand web content and communicate through it, as humans do now. Traditional knowledge-based systems are centralized, but the semantic web is distributed and heterogeneous. According to Mitra *et al.* (2003): 'Information sources, even those from the same domain, are heterogeneous in nature'. This heterogeneity has resulted in designing ontologies to lessen the difficulties of agents' understanding and communication. However, still another problem exists: ontologies themselves may have heterogeneity. This is when two ontologies are trying to express the same knowledge or concepts but use different languages or words (Euzenat & Valtchev, 2004).

Ontology alignment is a proposed solution to this problem by introducing a (proper) mapping of entities in two ontologies from two (different) domains. Bouquet *et al.* (2004a) defines ontology alignment as follows.

... given two ontologies which each describe a set of discrete entities (which can be classes, properties, rules, predicates etc.), find the relationships (e.g. equivalence or subsumption) holding between these entities.

Figure 1 shows a simplified ontology alignment framework. As shown in the figure, to extract an alignment it is customary to first apply some measures (simple or complex) to reach some initial similarity values. There is already a vast amount of research in the literature discussing lexical and structural measures suitable for



**Figure 1:** *A simplified alignment framework.*

ontology alignment. Having such similarity values, the next problem is how to form an ideal mapping. We refer to this problem as mapping extraction. The goal is to find correspondence of entities among two ontologies such that the overall similarity value is maximal. Therefore it can be viewed as an optimization problem in which evolutionary approaches could be a legitimate solution.

In this paper we introduce a genetic-based algorithm for the mapping extraction problem. First we give an explanation of related work in Section 2. Then in Section 3 explanations about the graph theoretical bases we use through the paper are given. To have a measure for calculation of how good a mapping is, we discuss coincidence-based weighting in Section 4. Our evolutionary algorithm is explained in Section 5 and in Section 6 evaluations are discussed. We also provide conclusions and explanations about future work in Section 7.

## 2. Related work

Unfortunately, and as stated by Bouquet *et al.* (2004b), work on ontology extractions is not common. However, current research on ontology mapping and its applications entails a large number of fields ranging from machine learning, concept lattices and formal theories to heuristics and linguistics. There is some similar work on matching graphs and trees (Hopcroft & Karp, 1973; Papadimitriou & Steiglitz, 1998), database schemas (Rahm & Bernstein, 2001) and even clustering compound objects with a machine learning technique (Bisson, 1992). Kalfoglou and Schorlemmer (2003) give a comprehensive review and presentation on the methods and approaches and the state of the art in ontology aligning.

Work related to our research is of the following two categories:

- ontology alignment weighting and similarity measures, in which the focus is on the introduction of new similarity measures between concepts of two ontologies, and a weight function to evaluate an alignment between two ontologies;
- ontology mapping extractions, in which the researchers try to address the problem of alignment extraction and propose methods to find a (proper) alignment among many different candidates.

There are also some works which address both problems simultaneously. We give a quick review on each category in the following sections.

### 2.1. Similarity measures and ontology alignment

There is a considerable amount of previous work on similarity measures and ontology alignment. Some standards of metrics are acknowledged and defined, as in the CommonKADS methodology (Schreiber *et al.*, 2000) or OntoWeb EU thematic network (Onto Web, 2002), which are partly endorsed by recognized bodies.

Also there has been some work on finding similarities of entities in two ontologies based on their structural standings. Valtchev (1999) computes the dissimilarity of elements in a hierarchy based on their distances from the closest common parent. The upward cotopic distance is introduced by Maedche and Zacharias (2002) who find dissimilarity of entities in hierarchies of ontologies. Resnik (1995) introduces a measure to calculate the similarity of WordNet<sup>1</sup>

<sup>1</sup>[wordnet.princeton.edu](http://wordnet.princeton.edu).

concepts, i.e. a single hierarchy. The similarity is computed based on the closest common parent and the distance of the two entities from the root. On the other hand, some methods tend toward a trade-off between different features such as efficiency and quality, as in QOM (Ehrig & Staab, 2003), and some have used approaches to integrate various similarity methods (Ehrig & Sure, 2004).

Also compound metrics use simple measures by combining them, in an attempt to improve the result of the mapping between two ontologies. One approach has been to define each measure as a dimension to find the Minkowski distance of two objects (Euzenat *et al.*, 2004). As introduced in Euzenat *et al.* (2004) another approach for this problem is to use a weighted average of features in which weights can be calculated by a machine learning technique. For example, GLUE (Doan *et al.*, 2003a) builds a similarity matrix by a machine learning approach. Also in APFEL (Ehrig & Staab, 2005) weights for each feature are calculated using decision trees. The user only has to provide some ontologies with known correct alignments. The learned decision tree is then used for aggregation and interpretation of the similarities. Abolhassani *et al.* (2006) introduces a new method for compound measure creation without any need for the mapping extraction phase.

## 2.2. Mapping extraction

Previous works do not specifically cover the problem of alignment extraction. A method for ontology alignment extraction proposed by Dieng and Hug (1998) examines linguistic features to compare two ontologies on the basis of an 'IS-A' relationship. To produce a reasonable extraction, Melnik *et al.* (2002) discuss the *stable marriage* problem (Gibbons, 1985).

There are also some other approaches, e.g. a machine learning approach to the problem is discussed by Doan *et al.* (2003b), and Mitra *et al.* (2003) describe a probabilistic based model.

Staab and Maedche (2002) have focused on structural and taxonomic comparison of two trees to extract an alignment, in which dissimilarity of the two concepts is calculated based on

their superclasses and subclasses. Stumme and Maedche (2001) use shared instances of two ontologies that are to be mapped; however, this work ignores the properties of classes.

Zhdanova and Shvaiko (2006) expand the notion of ontology matching to a community-driven approach to enable web communities to establish and reuse ontology mappings to achieve an adequate and timely domain representation.

Johnson *et al.* (2006) model inter-ontology relationship detection as an information retrieval task, where relationship is defined as any direct or indirect association between two ontological concepts.

Wang and Gasser (2002) present a specific formalization and algorithm for local interpretation of shared representations to build global semantic coherence for the distributed actions of individual agents, known as 'Mutual Online Ontology Alignment'.

LOM, as described in Li (2004), is a semi-automatic lexicon-based ontology mapping tool that supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped, based on their lexical similarity and simple heuristic methods.

## 3. Preliminaries and notation

In this section, we define some necessary mathematical concepts which are used throughout the paper.

### 3.1. Basic definitions

A graph  $G_i$ , by definition, consists of two sets,  $V(G_i)$ ,  $E(G_i)$ , where  $V(G_i)$  is the set of vertices and  $E(G_i)$  is the set of edges. The size of a graph is  $|V(G_i)|$ , which is denoted by  $|G_i|$ . Let us assume that labels assigned to nodes are chosen from a finite alphabet  $\Sigma$ . Let  $\lambda \notin \Sigma$  be a null character, and  $\Sigma_\lambda = \Sigma \cup \lambda$ .

### 3.2. Metric space

According to Rudin (1976), a set of points  $X$  along with a function is said to be a *metric space* if the function associates a real number with any pair of points  $p, q$ , denoted by  $d(p, q)$  and called

the distance  $p, q$ , such that

$$\begin{aligned} \forall x, y \in O \quad \delta(x, y) \geq 0 & \quad (\text{positiveness}) \\ \forall x \in O, \forall y, z \in O \quad \delta(x, x) \geq \delta(y, z) & \quad (\text{maximality}) \\ \forall x, y \in O \quad \delta(x, y) = \delta(y, x) & \quad (\text{symmetry}) \end{aligned}$$

Any function  $\delta$  satisfying the above conditions is said to be a distance function or a metric. In fact, the distance of two concepts belonging to two different ontologies is described as the distance of their labels in a metric space, and usually this metric distance is described by the *distance function* described above.

### 3.3. Typed graph

An ontology  $O_i$ , in this paper, is considered as a *typed graph*  $G_i$ . A typed graph, as defined in Haeri *et al.* (2006), is denoted by  $G_i(V, E, T)$ , for which  $E$  is of type  $E: V \times V \rightarrow T$ . Members of  $T$  are all from  $\Sigma_\lambda$ . In such a graph an edge  $e$  of type  $t$  between vertices  $v_{ij}$  and  $v_{ik}$  is denoted by  $e(v_{ij}, v_{ik}) : t$ . A homeomorphism from a typed graph  $G(V, E, T)$  to another typed graph  $G'(V', E', T)$  is a one-to-one correspondence between  $V$  and  $V'$ . In this paper, each ontology  $O_i$  is modelled using a typed graph  $G_i$  where concepts of  $O_i$  are nodes of  $G_i$ , and the relations/properties of  $O_i$  are *typed edges* of the graph.

### 3.4. Edge preservation

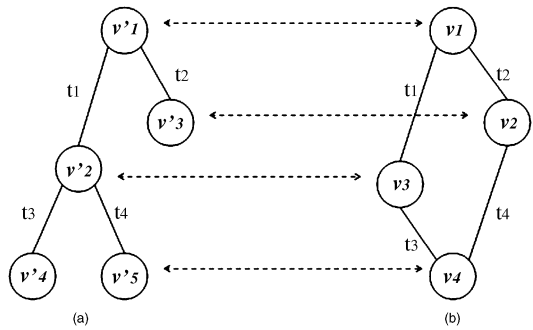
We will call an edge  $e(v_{ij}, v_{ik}) : t \in E(G_{i_1})$  *preserved under the mapping*  $\mathcal{M}$  if and only if there is an edge  $e(\mathcal{M}(v_{ij}), \mathcal{M}(v_{ik})) : t \in E(G_{i_2})$ . In other words, an edge  $e$  is preserved under mapping  $\mathcal{M}$  if and only if  $\exists e' \in E(G_{i_2}) : e' = (\mathcal{M}(v_{ij}), \mathcal{M}(v_{ik}))$ ,  $\mathcal{M}(e) = e'$ , and is not preserved otherwise.

The preservation of edges between corresponding nodes is the key point to find an ideal mapping. In fact in an ideal alignment most of the edges of one ontology are preserved in the second one.

### 3.5. Ontology alignment

In this section we will discuss our own understanding of a one-to-one alignment of two ontologies.

A one-to-one alignment of two ontologies  $O_{i_1}, O_{i_2}$  is denoted by  $\mathcal{M} : O_{i_1} \rightarrow O_{i_2}$  and is



**Figure 2:** A sample alignment of two graphs  $G, G'$ .

a one-to-one correspondence between nodes of the two graphs of  $O_{i_1}, O_{i_2} : (G_{i_1}, G_{i_2})$ .

Ehrig and Sure (2004) define the mapping function in the following way:

- $\mathcal{M} : O_{i_1} \rightarrow O_{i_2}$
- $\forall v \in G_{i_1} : \mathcal{M}(v) = v'$  if  $v' \in G_{i_2}$  and  $\delta(v, v') < t$ , for  $t$  a threshold

$v'$  is the *corresponding node* of  $v$  under the mapping  $\mathcal{M}$ .

Figure 2 illustrates a sample alignment for two example ontologies  $G, G'$ .

### 3.6. Our formulation of ontology alignment

We denote the correspondence from ontology  $O_i$  to  $O_j$  as described by the concept of *typed graphs*, i.e.  $G_i$  to  $G_j$  by  $\mathcal{M} : G_i \rightarrow G_j$ . It is defined as follows.

1.  $\forall v_i \in G_i, v_i$  corresponds to only one vertex  $v_j$  in  $G_j$  (denoted by  $\mathcal{M}(v_i) = v_j$ ), or does not correspond to any vertex in  $G_j$  (denoted by  $\mathcal{M}(v_i) = \text{null}$ ). And if  $v_1, v_2 \in V_i, v_1 \neq v_2, \mathcal{M}(v_1) \neq \text{null}, \mathcal{M}(v_2) \neq \text{null}$ , then  $\mathcal{M}(v_1) \neq \mathcal{M}(v_2)$ .
2. The correspondence of edges is determined by the correspondence of nodes:  $\forall e_i = (v_{i_1}, v_{i_2}) : t \in E(G_i)$ : if  $\mathcal{M}(v_{i_1}) = v_{j_1} \neq \text{null}, \mathcal{M}(v_{i_2}) = v_{j_2} \neq \text{null}$  and  $e_j = (v_{j_1}, v_{j_2}) : t \in E(G_j)$ , then  $e_i$  corresponds to  $e_j, \mathcal{M}(e(v_{i_1}, v_{i_2})) = e(v_{j_1}, v_{j_2})$ ; else  $e_i$  does not correspond to any edge in  $G_j$  (denoted by  $\mathcal{M}(e(v_{i_1}, v_{i_2})) = \text{null}$ ).
3. Let  $\mathcal{M}$  be a correspondence from  $G_i$  to  $G_j$ . We call  $\mathcal{M}$  a *map* from  $G_i$  to  $G_j$  if  $\forall v_i \in V_i, \mathcal{M}(v_i) \neq \text{null}$ , i.e.  $\mathcal{M}(v_i) \in V_j$ .

- Each map  $\mathcal{M}$  from  $G_i$  to  $G_j$  has a *weight* and this weight is defined by the coincidence-based technique described in Section 4.

The problem which is addressed in this paper is formulated as follows.

**Input:** Two ontologies together with a matrix, rows and columns of which stand for concepts of ontologies; each cell shows the distance of the two concepts as given by a distance measure.

**Output:** A proper alignment.

In what follows, we explain a technique to score possible mappings of ontologies, so-called coincidence-based mapping, in the next section and then use this weight function to extract a proper alignment with evolutionary approaches in Section 5.

#### 4. Coincidence-based weighting

In this section we introduce and discuss a new weighting model for an alignment with which we will later design our genetic algorithm.

The coincidence-based alignment weight function is sufficiently discussed in Haeri *et al.* (2006); here we will give an overview. Before discussing the weight itself, let us discuss the background.

There is a set of properties that we believe any mapping should comply with. Consider a mapping  $\mathcal{M}$  between two ontologies with graphs  $G_{i_1}, G_{i_2}$ , and two nodes  $v_{1_j}, v_{1_k} \in V(G_{i_1})$  and their matches  $\mathcal{M}(v_{1_j}), \mathcal{M}(v_{1_k})$ ; the weighting system should result in a high weight if  $v_{1_j}$  is close to  $\mathcal{M}(v_{1_j})$  and also  $v_{1_k}$  is close to  $\mathcal{M}(v_{1_k})$  and when  $e = (v_{1_j}, v_{1_k}) : t \in E(G_{i_1})$  is preserved under  $\mathcal{M}$ . This case is considered to be the most desired one and should be given the highest value.

An alternative is when the edge is not preserved. Here, a negligible negative point should be given. The reason for the negative point is the fact that the edge is not preserved and the structural matching of the graphs is interrupted. In this case the nodes are very close but the edge is missing.

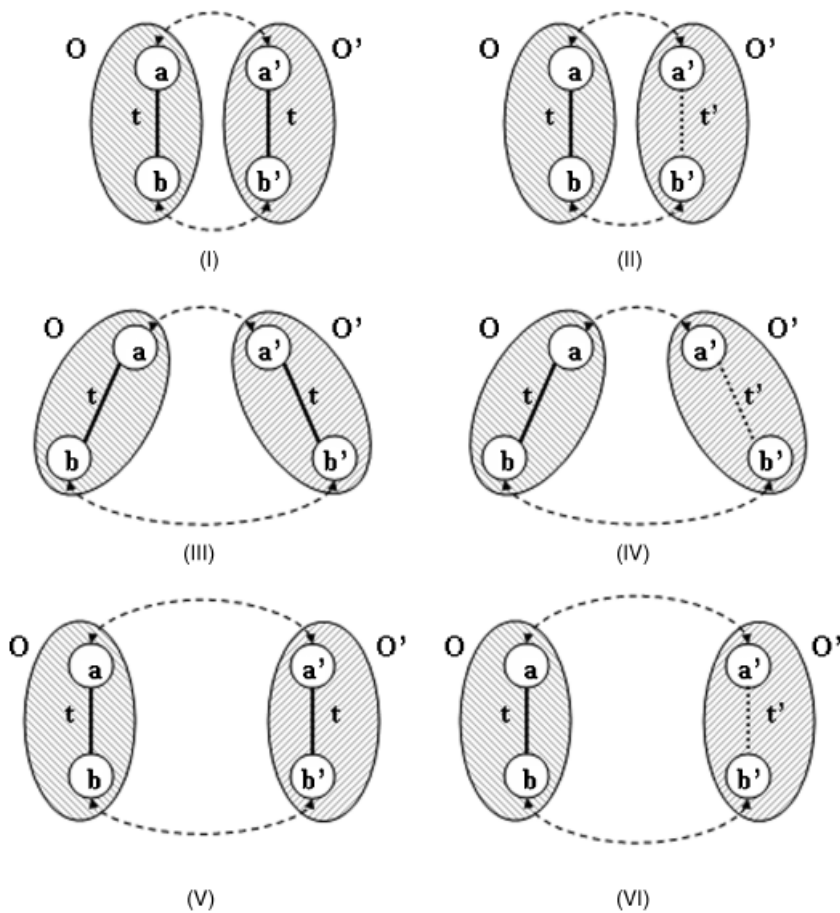
The farther any of the nodes is from its match, the lower should be the positive value of the

mapping. If the edge is preserved, we give this mapping a low positive value. But when the edge is not preserved, in fact it is an undesired mapping. So we give it a negative point. In this case not only are the nodes far from their matches, but also the edge is not preserved.

According to the above considerations there should be six different categories, which are graphically shown in Figure 3. In the following explanations we assume that  $G, G'$  are graphs of two ontologies  $O, O'$  to be aligned.  $a, b$  are concepts from  $G$ , and  $a', b'$  from  $G'$ .

- Category I:**  $a$  and  $a'$  are very close<sup>2</sup> and  $b, b'$  are close as well, and the edge between  $a, b$  is preserved under the matching process. This category is of much importance. This is because the two edges effectively coincide. To clarify the point suppose the case when  $a$  and  $b$  are 'means of communication' and 'mail' respectively, and  $a', b'$  are 'communication' and 'email'. The fact that there is an edge (i.e. *rdfs: type*) between both  $a, b$  and  $a', b'$  shows that the two edges coincide and that the two ontologies are describing the same world.
- Category II:** In this category, the two pairs of an edge are close to their matches, i.e.  $a$  is close to  $a'$  and  $b$  is close to  $b'$ . The only difference between this category and the previous one is that here the edge is not preserved. As described in Haeri *et al.* (2006), consider, for example, when  $O$  describes glazing technology whilst  $O'$  is the ontology of a simple glasses manufacturing studio. Let  $a, b$  be 'glass' and 'frame', and  $a', b'$  the same respectively. Although  $\delta(a, a'), \delta(b, b')$  are both small, the non-preservation of the edge ( $a, b$ ) is a negative point. The fact that the vertices coincide makes us not penalize this category much, because at least the concepts are close to their matches.
- Category III:** In this category the edge is preserved but only one of  $a$  or  $b$  is close to its match. This is good but not as good as the previous category. Consider two ontologies,

<sup>2</sup>In terms of the distance function described earlier.



**Figure 3:** Different properties of mapping in a metric space. Dotted edges with type  $t'$  show that there might be an edge of type  $t'$  or there might be no edge present. Dashed arrowed lines show the mapping elements. (I) shows the first category where vertices (concepts) are close and the edge is preserved. (II) is the same as category I except that the edge is not preserved. (III) shows the case when the edge is preserved but only one set of the endpoints of the two edges is close. (IV) is the same as category III except that the edge is not preserved. In (V) the edge is preserved but none of the endpoints is close to their match. (VI) is the same as category V except that the edge is not preserved.

describing two different worlds. Suppose  $O$  is describing high tech manufacturing while  $O'$  is describing a supermarket. Let  $a, b$  be 'laptop computer' and 'product' respectively and  $a', b'$  be 'laptops' and 'on sale item' respectively. The two ontologies are describing two totally different domains, whilst  $a$  and  $a'$  are close. So it seems as if such ontologies are getting close 'from the side of  $a'$ '. We would like to give such category a large weight, yet smaller than that of category I.

- Category IV:** As category II can be considered to be equivalent to category I, this category is equivalent to category III. The reason is that only one pair of the edge in  $O$  is close to what it is matched to, but the edge is not preserved under the matching. As is clear in Figure 3, in category IV the edge between  $a, b$  is not preserved, and  $b$  is far from  $b'$ . The only positive point of such a matching is that  $a$  and  $a'$  are close. As an example, just to make things clearer, con-

sider  $O$  to be describing a hotel's services, where  $a$  is 'egg' and  $b$  is 'omelette', and  $O'$  is describing a supermarket and  $a', b'$  are 'egg' and 'shampoo' respectively. This matching, which maps  $a$  to  $a'$  and  $b$  to  $b'$ , is not desired and is most probably a mistake. However, the mistake should not be penalized as much as the mistake in category VI.

- **Category V:** The last two categories describe the situation where none of the pairs of an edge is close to what it is matched to. Even though the edge might be preserved (as in category V) the two edges do not coincide at either endpoint. In other words, in these categories, both  $a, a'$  and  $b, b'$  are far from one another, and the difference is in the preservation of edges. The fact that the vertices are not close to their matches is quite enough to make us indifferent about the edge preservation. Because even if the edge is preserved, the two edges are not coincident. Both cases are not desired and should obtain low points. A clear example of category V would be when  $a$  is 'plant',  $b$  is 'water',  $a'$  is 'car' and  $b'$  is 'fuel'. Clearly this mapping is not a desired one.
- **Category VI:** This case is even worse than category V, where not even the edge is preserved. An example would be when  $a, b$  are 'mammal' and 'elephant' in  $O$  which is describing a zoo and  $a', b'$  are 'glasses' and 'frame' respectively in  $O'$  which is incidently describing a glasses manufacturing company. There is neither similarity between endpoints of the two edges, nor any preservation. The vertices in this category are mapped to vertices that have no similarities, semantically.

According to the above cases, the following weighting function is suggested:

$$\begin{aligned}
 w(\mathcal{M}) &= w_0(\mathcal{M}) - w_l(\mathcal{M}) - w_r(\mathcal{M}) \\
 w_0(\mathcal{M}) &= \sum_{(v_1, v_2): \exists E(G), (\mathcal{M}(v_1), \mathcal{M}(v_2)): \exists E(G')} f(v_1) + f(v_2) \\
 w_l(\mathcal{M}) &= \sum_{(v_1, v_2): \exists E(G), (\mathcal{M}(v_1), \mathcal{M}(v_2)): \exists E(G')} g(v_1) + g(v_2) \\
 w_r(\mathcal{M}) &= \sum_{(v_1, v_2): \nexists E(G), (\mathcal{M}(v_1), \mathcal{M}(v_2)): \exists E(G')} g(v_1) + g(v_2)
 \end{aligned}$$

The functions  $f$  and  $g$ , referred to as *normalization functions* (Haeri *et al.*, 2006), are in the form

$$\begin{aligned}
 f: R &\rightarrow R^+ \\
 g: R &\rightarrow R^+
 \end{aligned}$$

$f, g$  are related to the distance function. In fact,  $f$  should be a positive decreasing function, so that if  $\delta(v, \mathcal{M}(v))$  increases, it decreases to reduce the positive point. On the other hand,  $g$  should be a positive increasing function to grow with the growth of  $\delta(v, \mathcal{M}(v))$  to increase the negative point for that match. In any other cases, in one of the above six categories  $w$  will misbehave. Normalization functions are defined by tuning the system. This will be described later.

According to Haeri *et al.* (2006): 'no [not much] work is so far done on the problem of Ontology Alignment or Ontology Matching in which the graph theoretic backbone of a problem is scrutinized'. With the use of graph theory and such a modelling we believe that there is a vast area for new work on the problem of ontology aligning. The coincidence measure explained in this section is a step forward in this direction. We believe it can be used in various ways for the mapping extraction problem. The sole use of it in this paper is introduced in the next section.

## 5. Genetic algorithm

This section describes the developed genetic algorithm (GA).

Matching two general graphs in polynomial running time algorithms is impossible because the problem in its general case is MAX SNP-hard (Arora *et al.*, 1992). So a random search algorithm could be a good idea when designed carefully. This leads us to the idea of using GAs.

GA solutions are evolutionary algorithms which will approach the final state by gradually improving the solution. Any problem which is solved by a GA should first be coded in a way that it can be easily handled in different parts of the GA. Each coding of a solution will form an individual. Some individuals which are stored and processed in each iteration form the popula-

tion. The population improves in each step with the use of crossovers and mutations and the best individual will finally be reported as the answer of the GA. In the following sections we explain the different parts of our GA solution.

### 5.1. Coding a mapping

To code a mapping we use hashmaps (Cormen *et al.*, 2001) in which keys are concepts of one ontology and entries are concepts of another. This data structure helps us to easily manipulate a one-to-one alignment, with a search of concepts in  $O(1)$ . Entry for each key is actually the corresponding node of that key in the mapping. That is, if  $v_i \in O$  is mapped to  $v'_i \in O'$  then in the hashmap  $h$  we shall have the  $v_i$  as the key, and  $h(v_i) = v'_i$ .

**5.1.1. Pairs** According to the coincidence-based weighting, we define a *pair* as two concepts from one ontology between which there is a relation (so there is an edge in the graph of that ontology between them). Figure 2 shows the alignment of two ontologies in which  $(v_1, v_2), (v_1, v_3), (v_3, v_4), (v_2, v_4)$  are pairs of  $G$ .

A pair also has a weight according to the alignment in which it is involved. Thus, a pair is a function of the form

$$P : V \times V \times T \rightarrow R$$

where  $V$  is the set of vertices in ontology graph  $G$  and  $T$  is a set of labels in  $\Sigma_\lambda$ . So an ontology in a matching has a limited number of pairs.

The weight of a pair depends on the alignment in which the ontology is involved. Let  $G_{i_1}, G_{i_2}$  be two graphs of two aligned ontologies, and  $v_{1_j}, v_{1_k} \in V(G_{i_1})$ . Also assume an edge between  $v_{1_j}, v_{1_k}$  to be of type  $t$ ,  $e_{1_{jk}} = e(v_{1_j}, v_{1_k}) : t$ .

$P(v_{1_j}, v_{1_k}, t)$  in the alignment of two ontologies is given by

$$P(v_{1_j}, v_{1_k}, t) = \begin{cases} f(v_{1_j}) + f(v_{1_k}) & e_{1_{jk}} \text{ preserved} \\ -(g(v_{1_j}) + g(v_{1_k})) & e_{1_{jk}} \text{ not preserved} \\ -\infty & \text{if } e_{1_{jk}} \notin E(G_{i_1}) \end{cases}$$

For a couple of concepts which do not form a pair the value of function  $P$  is set to be  $-\infty$ . The definition of pairs is useful in the crossover function which will try to improve the structural matching.

In the alignment of two ontologies,  $O_{i_1}(G_{i_1}), O_{i_2}(G_{i_2})$ , say  $\mathcal{M} : O_{i_1} \rightarrow O_{i_2}$ , we also define the *weight* of a single concept from one ontology,  $W(v_{1_j})$  where  $v_{1_j} \in V(G_{i_1})$ , as follows:

$$\text{if } v_{1_j} \in V(G_{i_1}), \mathcal{M}(v_{1_j}) \in V(G_{i_2})$$

$$W(v_{1_j}) = \sum_{\forall v \in G_{i_2}, e(v_{1_j}, v) : t \in E(G_{i_1})} P(v_{1_j}, v, t)$$

**5.1.2. An example** To make the definition of pair and its corresponding weights described above clearer, we give an example of how to compute these weights.

In Figure 2 we have

$$P(v_1, v_2, t_2) = f(v_1) + f(v_2)$$

$$P(v_1, v_3, t_1) = f(v_1) + f(v_3)$$

$$P(v_3, v_4, t_3) = -g(v_3) - g(v_4)$$

$$P(v_2, v_4, t_4) = -g(v_2) - g(v_4)$$

$$P(v_1, v_4, t_i) = P(v_2, v_3, t_i) = -\infty$$

$$W(v_1) = (f(v_1) + f(v_2)) + (f(v_1) + f(v_3))$$

$$W(v_2) = (f(v_2) + f(v_1)) - (g(v_2) + g(v_4))$$

$$W(v_3) = (f(v_3) + f(v_1)) - (g(v_3) + g(v_4))$$

$$W(v_4) = -(g(v_4) + g(v_3)) - (g(v_4) + g(v_2))$$

Now, with these definitions, it is time to describe the steps of our GA.

### 5.2. Initialization

As for any GA, a primary population is needed. A population is made up of some individuals each of which is a solution to the problem (a mapping in this problem). The start population is initialized randomly, with an initial size of 1000 individuals. The ideal mapping can be reached more quickly if the initial individuals are made on the basis of the labels of concepts. That is, if  $v_{1_j}$  in  $G_{i_1}$  and  $v_{2_j}$  in  $G_{i_2}$  have the same

labels, then let  $v_{1j}$  correspond to  $v_{2j}$  in the initial mapping.

### 5.3. Selection

In each iteration, we sort the 1000 individuals according to their fitness described in Section 4 (coincidence-based weight function), and we select the 500 best individuals as parents of the next step. From these 500 individuals, with the use of the crossover and mutation functions (as we shall see later), 1000 new individuals are created. These 1000 individuals are sent to the next iteration as parents.

Two crossover functions are designed, one based on pairs and the other based on solitary vertices. In the following two sections we explain each one in detail.

### 5.4. Crossover I

In the first crossover, crossover I, the pairs are the primary concern and the best pairs from the parents are preserved in offspring.

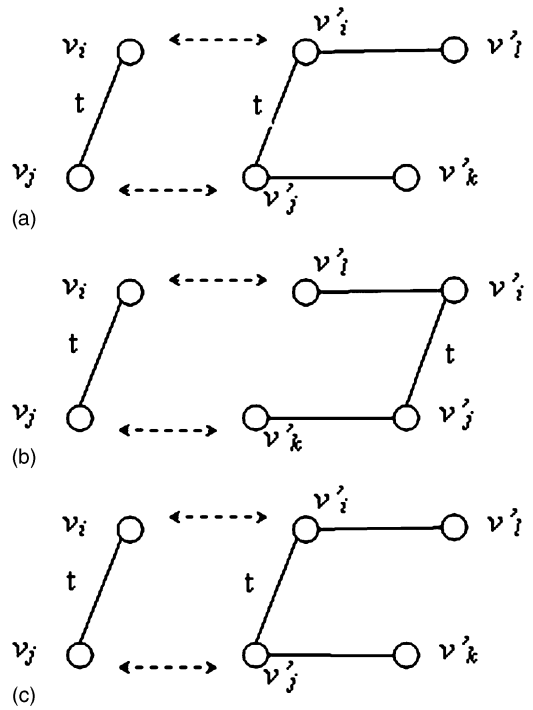
For every single node in the first ontology graph, the pairs of that node are examined in the two mappings (i.e. two parents), and the best pair, which has the highest value of weight, is copied in the offspring. If a matched node in ontology graph  $G'$  is already assigned to some other node of  $G$ , the assignment is done to a random unassigned node.

In Figure 4 two mappings as parents and the result of crossover I are shown. The pair  $(v_i, v_j, t)$  in parent 1 has greater weight than in parent 2. This results from computing  $P(v_i, v_j, t)$  in both alignments. In parent 1 we have  $P(v_i, v_j, t) = f(v_i) + f(v_j)$  because the edge  $e(v_i, v_j): t$  is preserved under  $\mathcal{M}$ , but in parent 2 we have  $P(v_i, v_j, t) = -(g(v_i) + g(v_j))$  because the edge  $e(v_i, v'_j): t$  is not preserved. So the pair in parent 1 is chosen for offspring. Since  $f, g$  are positive functions, in the offspring we have  $\mathcal{M}(v_i) = v'_i, \mathcal{M}(v_j) = v'_j$ .

It is clear that in this crossover the pairs in offspring are not worse than those of parents.

### 5.5. Crossover II

In this crossover function, single nodes are compared according to their weights. As we



**Figure 4:** Crossover I: (a) part of parent 1 mapping; (b) part of parent 2 mapping; (c) part of offspring.

described before, the weight of a single node in a mapping is the sum of weights of pairs in which that node is included.

Consider two parents in two ontology graphs  $G_{i_1}, G_{i_2}$ . To make an offspring from two parents, for every node in  $G_{i_1}$ , say  $v_{1j}$ , the mapping with larger  $W(v_{1j})$  is copied to the offspring. If  $\mathcal{M}(v_{1j})$  in  $G_{i_2}$  is already assigned with some other node of  $G_{i_1}$ , then  $v_{1j}$  is put in a reserved list. At the end of the complete iteration of nodes in  $G_{i_1}$ , the nodes in the reserved list are randomly mapped to the unassigned nodes of  $G_{i_2}$ . The random assignment is not done in the middle of an iteration to prevent nodes of  $G_{i_2}$  being assigned to some random nodes that can be assigned to better nodes later in the iteration. So this random assignment is postponed until all nodes of  $G_{i_1}$  are examined for mapping nodes in  $G_{i_2}$ .

As an example suppose  $v_{1_m} \in V(G_{i_1})$  should be mapped to  $v_{2_m} \in V(G_{i_2})$  but  $v_{2_m}$  is already

mapped by some node from  $G_{i_1}$ ; if at that time we assign  $v_{1_m}$  to some random node like  $v_{2_n} \in V(G_{i_2})$ , it will prevent a possible good mapping of  $v_{1_n}$  to  $v_{2_n}$  later in the iteration. So this random assignment is delayed until no more assignments are possible.

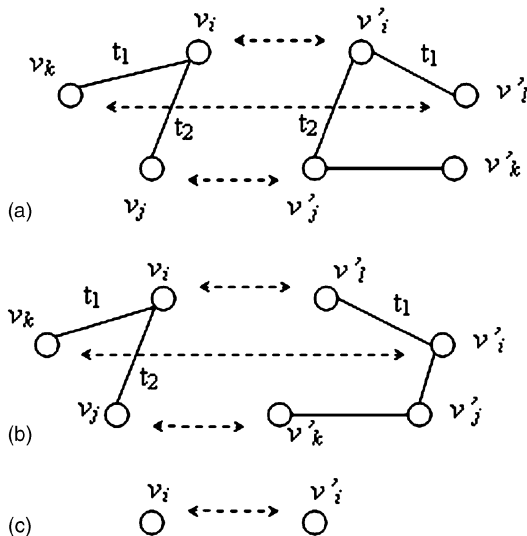
In Figure 5 two mappings between two ontologies  $O, O'$  are shown, and we want to decide the match node for  $v_i \in V(G)$  in the offspring. In parent 1 we have

$$\begin{aligned} W(v_i) &= P(v_i, v_j, t_2) + P(v_i, v_k, t_1) \\ &= (f(v_i) + f(v_j)) + (f(v_i) + f(v_k)) \end{aligned}$$

and in parent 2 we have

$$\begin{aligned} W(v_i) &= P(v_i, v_j, t_2) + P(v_i, v_k, t_1) \\ &= -(g(v_i) + g(v_j)) + (f(v_i) + f(v_k)) \end{aligned}$$

Again it should be noted that  $f$  and  $g$  are positive functions so the value of  $W(v_i)$  in parent 1 (Figure 5(a)) is greater than that in parent 2 (Figure 5(b)). So, as shown in Figure 5(c), the corresponding node of  $v_i \in V(G)$  in offspring is chosen by the mapping from parent 1, and therefore is  $v'_i \in V(G')$ .



**Figure 5:** Crossover II: (a) part of parent 1 mapping; (b) part of parent 2 mapping; (c) part of offspring.

This kind of crossover seems reasonable because the mapping of a single node in the offspring is not worse than that of the two parents. So by this assumption, little by little, mappings of nodes will converge to ideal ones.

### 5.6. Mutation

A proportion of the population are mutated with some probability, different in various iterations. In a mutation of a mapping of two ontologies with graphs  $G_{i_1}, G_{i_2}$ , two random nodes from  $G_{i_1}$  are chosen, and their matches in  $G_{i_2}$  are substituted with each other. Let  $v_{1_j}, v_{1_k} \in V(G_{i_1})$  be chosen randomly; also let  $\mathcal{M}(v_{1_j}) = v_{2_j} \in V(G_{i_2}), \mathcal{M}(v_{1_k}) = v_{2_k} \in V(G_{i_2})$ . In the mutation process we just substitute the matched nodes of the selected nodes. So the new mapping will be  $\mathcal{M}(v_{1_j}) = v_{2_k} \in V(G_{i_2}), \mathcal{M}(v_{1_k}) = v_{2_j} \in V(G_{i_2})$ .

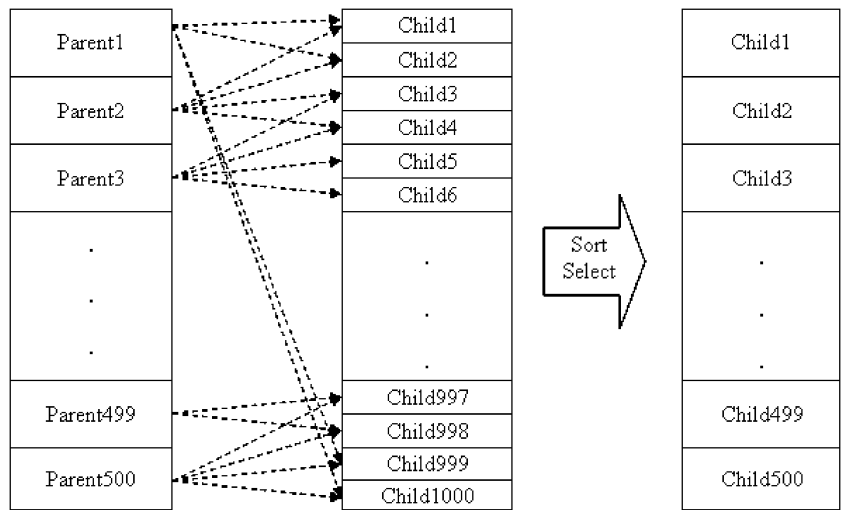
### 5.7. Continuation

One important issue with any GA is how to use the crossover and mutation functions, and how to create the new population based on the old one. In our solution, the two previously explained crossover functions are invoked on the  $i$ th and  $(i+1)$ th parents to create two offspring. The last parent is mixed with the first one to produce the last two offspring.

Our population, as described before, contains 500 individuals. These individuals are sorted decreasingly, and the sorted array forms the parents of the current step. In each iteration, these 500 parents, with the help of a series of crossovers and mutations (as explained earlier), produce 1000 new individuals. The resultant array of individuals is then sorted and the best 500 of them are selected as parents of the next step. Figure 6 shows this process.

### 5.8. End condition

Basically there is no end for the execution of any evolutionary algorithm and specifically a GA, where a user must pause the run process if she/he thinks the results are reasonable. However, to end the iteration of our GA, we used a threshold for convergence. The sequential GA is continued until the best mapping among all



**Figure 6:** Population generation in each step of the GA.

individuals in the population does not improve for more than 15 steps. Such mapping is reported as the answer for the problem of a proper alignment. The alignment process of two ontologies is then finalized.

## 6. Evaluation

To evaluate our GA, we designed three kinds of experiments. In the first experiment, we tested the GA with diverse mutation probabilities. In the second experiment, we tried to align two identical ontologies (actually we aligned one ontology with itself). This experiment helped us examine the efficiency and accuracy of the algorithm, when two ontologies are similar to each other. To verify our contribution, we did a third experiment in which we used a naive local search alignment method.

We have already discussed similarity measures in Section 2. It is an important issue to pay attention to in the ontology alignment and extraction process. Some similarity measures have been proposed earlier. For example, Euzenat and Altchev (2004) developed a similarity metric between concepts in OWL ontologies, which is a weighted combination of similarities of various features in OWL concept definitions: labels, domains, ranges of properties, restrictions on properties, types, IS-A relationships. Mitra *et al.* (2001) and Noy and Musen (2003)

use the combination of interactive specifications of mappings and heuristics to propose proper mappings. In this paper we used the Levenshtein (1966) string-based similarity measure for the concepts, where the dissimilarity of any two concepts (from two ontologies) is calculated by the Levenshtein distance. This measure is suitable for our experiments since most of the heterogeneity in the ontologies in our test collection comes from lexical differences. However, it should be noted that our coincidence-based weighting and hence our GA solution is independent of any similarity measure. To apply it to any other alignment, one can select another suitable measure for that domain and find similarity values to be used in our algorithm.<sup>3</sup>

### 6.1. Limitations

Coincidence-based weighting is innovative but there are essential practical limitations to applying this method. The most important limitation is the available ontologies and test collections. Most of them do not have a large taxonomic structure and so the method does not have enough merit for them. However, in our search for a suitable test collection we found the ‘Tourism’ ontologies good (Tourism Ontology

<sup>3</sup>As cited in Section 2, there are some good papers on selection of an appropriate measure for a domain which can be applied here.

FOAM, n.d.) with approximately 340 classes and concepts.

## 6.2. Various experiments

For the tourism ontologies, an ideal alignment is included in the test collection. We use such information to calculate the precision (Baeza-Yates & Ribeiro-Neto, 1999) for each experiment. Consider  $\mathcal{M}$  to be a mapping  $\mathcal{M}O \rightarrow O'$ . To find the accuracy of the method and calculate the precision, we need a *model* alignment  $\mathcal{M}'$  which is formed and extracted by some expert. Let  $G, G'$  be the graphs for  $O, O'$  respectively. Also suppose that  $S$  is the set of vertices  $v_i$  in  $V(G)$  where  $\mathcal{M}(v_i) = v'_i = \mathcal{M}'(v_i)$ . In other words

$$\forall v_i \in V(G) : v_i \in S \Leftrightarrow \mathcal{M}(v_i) = \mathcal{M}'(v_i)$$

Now, the precision of the alignment  $\mathcal{M}$  is given by

$$\text{precision}(\mathcal{M}) = \frac{|S|}{|V(G)|}$$

**6.2.1. Experiment 1** As discussed previously, in this experiment we aligned ‘TourismA’ with ‘TourismB’. This is the main experiment to check the efficiency of our coincidence-based GA.

In the experiment, from each two parents we made two offspring, one with the use of crossover I and the other with crossover II. From the four different individuals (parents and offspring) we chose the two best to introduce as children of this amalgamation.

The normalization functions are as follows:

$$f(v) = \frac{1}{e^{\delta(v, \mathcal{M}(v))}}$$

$$g(v) = \frac{1}{e^{\max(5, 15 - \delta(v, \mathcal{M}(v)))}}$$

These functions actually satisfy the characteristics expected from  $f, g$  (explained in Section 4).  $f$  is a decreasing function and decreases with the growth of  $\delta$ , and  $g$  is increasing. Exponential functions were chosen for  $f, g$  so that  $f, g$  would have close and comparable values. In fact, these functions match the discussions on positive and

negative points for different categories of a coincidence-based weight.

- *Experiment 1.1* After the 1000 individuals are created, we mutate the **lower half** of them (with the mutation function described before) with a probability of **0.7**.
- *Experiment 1.2* After the 1000 individuals are created, we mutate the **lower half** of them (with the mutation function described before) with a probability of **0.3**.
- *Experiment 1.3* Mutation was done on each of the 1000 individuals in **all** of the 1000 children with a probability of **0.5**.

**6.2.2. Experiment 2** In this experiment we are aligning TourismA with itself. This actually is a verification that shows how efficiently the GA will work if two ontologies are very similar and even coincide.

The generation summary is similar to the previous experiment, and mutation was done on the **lower half** of the individuals, with a probability of 0.5.

The normalization functions are similar to the previous experiment:

$$f(v) = \frac{1}{e^{\delta(v, \mathcal{M}(v))}}$$

$$g(v) = \frac{1}{e^{\max(5, 15 - \delta(v, \mathcal{M}(v)))}}$$

**6.2.3. Experiment 3** This experiment provides a baseline comparison of the GA method with a naive local search method. In this part, we implemented a naive hill-climbing local search method. For the start point, we made an initial alignment. In this initial alignment, all concepts in TourismA are matched with concepts in TourismB. For a node  $v_j$  in TourismA if there were a node  $v'_j$  with label  $label(v_j)$  in TourismB, we matched  $v_j$  with  $v'_j$ . Otherwise we mapped  $v_j$  to a random node of TourismB.

After that, in each iteration, the best single change (mutation) was performed to improve the weight value of the alignment. We iterated the method until almost 1000 steps, where the results did not improve for more than 15 steps.

### 6.3. Results

Figure 7 shows the results of the above experiments according to the precision measure. As shown, with identical graphs the GA finds the best mapping and precision is 1. With the experiments other, however, the result is a little inaccurate in comparison with the ideal alignment and precision is approximately 0.8.

In our experiments, the distance threshold, which we discussed in Section 3.5, is set to be 4. We chose this number by experience; however, there could be other ways of determining it, such as machine learning techniques etc. It is also possible to add one level of iteration to our GA to test different values for the distance threshold and select the threshold that results in the best total weight for the mapping. Since the labels for concepts in a typical ontology normally do not exceed the length of an English (or other natural language) word, the upper limit of the distance threshold can be said to be less than 20. So by execution of our GA with different distance threshold values (e.g. 1 to 20) we may find the best threshold value for an alignment task.

We also did an investigation on iteration number and convergence of the result in this GA for Experiment 1. The results are shown in Figure 8.

Figure 8 shows the weight of the best alignment, i.e. the best individual in the population, reached in each iteration of the running process in the GA. The figure shows the best alignment weight for all three experiments. It is clear that in all experiments the GA converges before 35 iterations. The precision average for convergence is 79%. This means that the GA will not find a better solution if it runs more than this number of iterations. Usually in such circumstances all individuals in the population converge to a single individual and are equal, so that the combination of them in the form of crossovers will not improve the solution and will result in the same individuals. In addition, the mutation in most cases will result in a worse individual rather than a better one. The reason is quite clear. The optimal states (individuals) are much fewer than worse states, and the mutation is absolutely random. Mutation is mostly useful when the population is trapped in

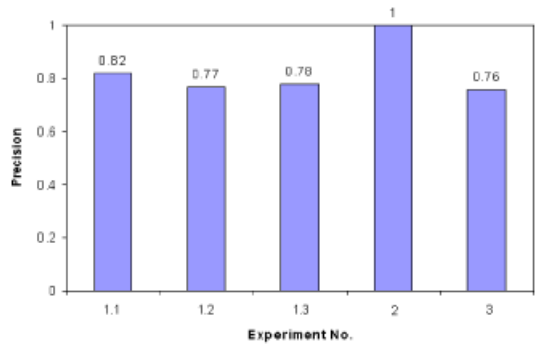


Figure 7: Precision results of the experiments.

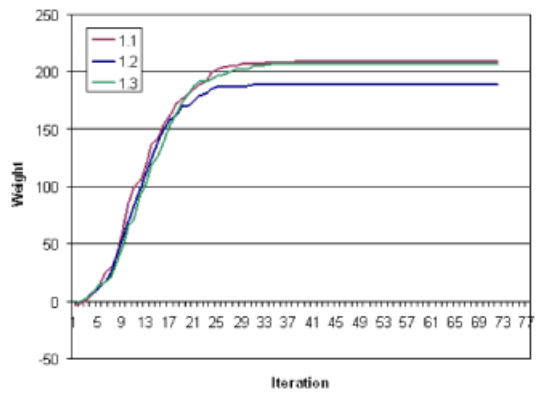


Figure 8: Convergence of results in the GA.

a local maximum, and the mutation will move it to another point in the GA.

It should be noted that in this area the main criterion of the goodness of an approach, which is the precision of the extracted mapping and an optimal time solution, is beyond the state of the research.

## 7. Conclusion and future work

In this paper, we discussed our method for mapping extraction in an ontology alignment framework. First, we introduced our modelling of an ontology with the concept of typed graphs. Then our coincidence-based measure for weighting of different candidate alignments was discussed. Our solution with help from graph theory (and which was contrived for ontology

alignment only) can be further extended to a wider range of problems. Graph matching and similarity measures of graphs have many applications in machine vision, pattern recognition, bio-informatics etc.

We defined two crossover functions based on the coincidence-based measure. They are crafted in a way that ensures that new generations are not worse than previous ones. In our experiments our GA solution converges very rapidly, e.g. after approximately 40 iterations, which, in order of magnitude, is considerably better than a naive 'candidate mapping generation and test' approach. This number can be reduced even more by choosing a biased initial population, where labels can be involved to choose better initial mappings.

There are also some weaknesses with GAs. One of them is the dependence of the results on the initial population. The more significant weakness is when the two ontologies are sparse graphs or, even worse, when they are like forests. In these cases the domain for crossover is not a soft one, and small changes in an individual in the crossover or mutation might take it to a very far point. The reason for this anomaly is that in sparse graphs or jungles, the number of disconnected vertices is greater and therefore a small change in the alignment will map one node to another vertex which, probably, will not form a coincident mapping. Roughly speaking, the more connected and taxonomic the two ontologies are, the better the results of the GA coincidence-based extraction will be.

In continuation of our research, work is now being done on tree-like ontologies (which seem a very common form). Once we can align tree ontologies, we can model ontologies as trees and align the resulting trees. We are also interested in extending our theory and mechanisms for matching ontologies based on their various graphical shapes, the properties of subgraphs etc.

### Acknowledgements

This research was in part supported by a grant from IPM (No. CS1385-4-01).

### References

- ABOLHASSANI, H., S.H. HAERI and B. HARIRI (2006) On ontology alignment experiments, *Webology*, **3** (3), article 28.
- ARORA, A., C. LUN, R. MOTWANI, M. SUDAN and M. SZEGEDY (1992) Proof verification and hardness of approximation problems, in *33rd IEEE Symposium on the Foundations of Computer Science*, New York: IEEE Press, 14–23.
- BAEZA-YATES, R. and B. RIBEIRO-NETO (1999) *Modern Informational Retrieval*, Reading, MA: Addison-Wesley.
- BISSON, G. (1992) Learning in FOL with similarity measure, in *Proceedings of the 10th American Association for Artificial Intelligence Conference*, Palo Alto, CA, 82–87.
- BOUQUET, P., J. EUZENAT, E. FRANCONI, L. SERAFINI, G. STAMOU and S. TESSARIS (2004a) Specification of a common framework for characterizing alignment, Deliverable 2.2.1, Knowledge Web NoE.
- BOUQUET, P., J. EUZENAT, E. FRANCONI, L. SERAFINI, G. STAMOU and S. TESSARIS (2004b) Specification of common network framework for characterizing alignments, Knowledge Web NoE 2.2.1.
- CORMEN, T.H., C.E. LEISERSON, R.L. RIVEST and C. STEIN (2001) *Introduction to Algorithms*, 2nd edn, Boston, MA: MIT Press.
- DIENG, R. and S. HUG (1998) Comparison of personal ontologies represented through conceptual graphs, in *13th ECAI*, 341–345.
- DOAN, A., P. DOMINGOS and A. HALEVY (2003a) Learning to match the schemas of data sources: a multistrategy approach, *Machine Learning*, **50** (3), 279–301.
- DOAN, A., J. MADHAVAN, P. DOMINGOS and A. HALEVY (2003b) Ontology matching: a machine learning approach, in *Handbook on Ontologies in Information Systems*, Berlin: Springer.
- EHRRIG, M. and Y.S. STAAB (2003) QOM quick ontology mapping, in *Proceedings of ISWC-2003*.
- EHRRIG, M. and Y.S. STAAB (2005) Bootstrapping ontology alignment methods with APFEL, in *Proceedings of the 4th International Semantic Web Conference*. Lecture Notes in Computer Science, Berlin: Springer, 186–200.
- EHRRIG, M. and Y. SURE (2004) Ontology mapping – an integrated approach, in *1st European Semantic Web Symposium*, Berlin: Springer, 76–91.
- EUZENAT, J. and P. ALTSCHEV (2004) Similarity-based ontology alignment in OWL-Lite, in *16th European Conference on Artificial Intelligence*.
- EUZENAT, J. and P. VALTCHEV (2004) An integrative proximity measure for ontology alignment, in *Proceedings of the 3rd International Semantic Web Conference*.

- EUZENAT, J., J. BARRASA, P. BOUQUET and J. BO (2004) State of the art on ontology alignment, *Knowledge Web*, Statistical Research Division.
- GIBBONS, A. (1985) *Algorithmic Graph Theory*, Cambridge: Cambridge University Press.
- HAERI, H., B. HARIRI and H. ABOLHASSANI (2006) Coincidence-based refinement of ontology alignment, in *3rd International Conference on Soft Computing and Intelligent Systems*.
- HOPCROFT, J. and R. KARP (1973) An  $n^5/2$  algorithm for maximum matchings in bipartite graphs, *SIAM Journal of Computing*, **2** (4), 225–231.
- JOHNSON, H.L., K.B. COHEN, W.A. BAUMGARTNER, Z. LU, M. BADA, T. KESTER, H. KIM and L. HUNTER (2006) Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies, *Pacific Symposium on Biocomputing*, 28–39.
- KALFOGLOU, Y. and M. SCHORLEMMER (2003) Ontology mapping: the state of the art, *Knowledge Engineering Review*, **18**, 1–31.
- LEVENSHEIN, V.I. (1966) Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics – Doklady*, **6**, 707–710.
- LI, J. (2004) LOM: a lexicon-based ontology mapping tool, in *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS04) and Information Interpretation and Integration Conference (I3CON)*.
- MAEDCHE, A. and V. ZACHARIAS (2002) Clustering ontology based metadata in the semantic web, in *Proceedings of the 13th ECML and 6th PKDD*.
- MELNIK, S., H. GARCIA-MOLINA and E. RAHM (2002) Similarity flooding: a versatile graph matching algorithm, in *Proceedings of the 18th International Conference on Data Engineering*, 117–128.
- MITRA, P., G. WIEDERHOLD and S. DECKER (2001) A scalable framework for interoperation of information sources, in *1st International Semantic Web Working Symposium, Stanford, CA*.
- MITRA, P., N.F. NOY and A.R. JAISWAL (2003) Omen: a probabilistic ontology mapping tool, in *4th International Semantic Web Conference*, Lecture Notes in Computer Science 3729, Berlin: Springer, 537–547.
- NOY, N.F. and M.A. MUSEN (2003) The PROMPT suite: interactive tools for ontology merging and mapping, *International Journal of Human-Computer Studies*, **59** (6), 983–1024.
- ONTOWEB (2002) *OntoWeb: A Survey on Ontology Tools*; available from [www.ontoweb.org/deliverable.htm](http://www.ontoweb.org/deliverable.htm).
- PAPADIMITRIOU, C.H. and K. STEIGLITZ (1998) *Combinatorial Optimization Algorithms and Complexity*, Upper Saddle River, NJ: Prentice Hall.
- RAHM, E. and P. BERNSTEIN (2001) A survey of approaches to automatic schema matching, *VLDB Journal*, **10** (4), 334–350.
- RESNIK, P. (1995) Using information content to evaluate semantic similarity in a taxonomy, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.
- RUDIN, W. (1976) *Principles of Mathematical Analysis*, 3rd edn, New York: McGraw-Hill.
- SCHREIBER, G., R. DE HOOG, H. AKKERMANS, A. ANJEWIERDEN, N. SHADBOLT and W.V. DE VELDE (2000) *Knowledge Engineering and Management*, Boston, MA: MIT Press.
- STAAB, S. and A. MAEDCHE (2002) Measuring similarity between ontologies, *Lecture Notes in Artificial Intelligence*, **2473**, 251–263.
- STUMME, G. and A. MAEDCHE (2001) FCA-merge: bottom-up merging of ontologies, in *Proceedings of the 17th IJCAI*, 225–230.
- TOURISM ONTOLOGY FOAM (n.d.) Available from <http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies/>.
- VALTCHEV, P. (1999) Construction automatique de taxonomies pour l'aide la representation de connaissances par objets, PhD Dissertation, Université Grenoble.
- WANG, J. and L. GASSER (2002) Mutual online ontology alignment, in *Proceedings of the AAMAS 2002 Workshop*.
- ZHDANOVA, A.V. and P. SHVAIKO (2006) Community-driven ontology matching, in *Proceedings of ESWC*, 34–49.

## The authors

### Vahed Qazvinian

Vahed Qazvinian joined Sharif University of Technology, Tehran, as a BSc student in computer engineering in 2003. In 2005 he joined the Web Intelligence Laboratory under the supervision of Dr Abolhassani, as a senior undergraduate research assistant. His main interest is in ontology alignment scoring and extraction.

### Hassan Abolhassani

Hassan Abolhassani received his BSc, MSc and PhD all in software engineering. He joined the Computer Engineering Department of Sharif University of Technology in September 2003 and is now an Assistant Professor lecturing courses in the web intelligence area and directing the Web Intelligence Laboratory. His main

interests are ontology alignment, search engine result clustering, semantic search engines and trust in social networks. He is a member of the Computer Society of Iran.

### **Seyed H. Haeri (Hossein)**

From 2006 Hossein Haeri has been a PhD student in theoretical computer science at Heriot-Watt University, Edinburgh. He has published in the *Journal of Advanced Computational Intelligence and Intelligent Informatics* and in the *Joint 3rd International Conference on Soft Com-*

*puting and Intelligent Systems and 7th International Symposium on Advanced Intelligent Systems*. He is a member of the Association of C/C++ Users.

### **Babak Bagheri Hariri**

Babak Hariri has graduated with an MSc in software engineering from Sharif University of Technology. His current areas of academic research include the semantic web and his main interest is ontology alignment.