Identifying and Eliminating Inconsistencies in Mappings across Hierarchical Ontologies

Bhavesh Sanghvi, Neeraj Koul, and Vasant Honavar

Iowa State University,
Ames, IA 50011-1041, USA

Abstract. Many applications require the establishment of mappings between ontologies. Such mappings are established by domain experts or automated tools. Errors in mappings can introduce inconsistencies in the resulting combined ontology. We consider the problem of identifying the largest consistent subset of mappings in hierarchical ontologies. We consider mappings that assert that a concept in one ontology is a subconcept, superconcept, or equivalent concept of a concept in another ontology and show that even in this simple setting, the task of identifying the largest consistent subset is NP-hard. We explore several polynomial time algorithms for finding suboptimal solutions including a heuristic algorithm to this problem. We experimentally compare the algorithms using several synthetic as well as real-world ontologies and mappings.

1 Introduction

Many semantic web applications (e.g., building predictive models from disparate data sources, assembling composite web services using components from multiple repositories) rely on mappings to bridge the semantic gaps between disparate ontologies. Such mappings may be established by domain experts, or automated tools designed to discover such mappings from data[1, 9]. This is inherently an error prone process. Errors in mappings can make the resulting ontology inconsistent. Consider for example an ontology with a single concept (say c_a) and another ontology with a single concept (say c_b). Consider the following set of mapping between the ontologies: (1) $c_a \prec c_b$ and (2) $c_b \prec c_a$ where $\prec$ denotes *strict* subconcept relationship. Obviously both these mappings cannot simultaneously hold. To ensure the consistency of the resulting ontology, one of the two mappings has to be discarded. While in this case the ontologies were simple and mappings were straightforward, in general, because of the large number of concepts and mappings involved, it is necessary to automate the process of identifying and eliminating inconsistencies.

In this paper we consider the problem of identifying the maximum subset of consistent mappings between hierarchical ontologies. We show that this problem is NP-hardeven in the restricted setting of mappings that assert that a concept in one ontology is a subconcept, superconcept, or equivalent concept of a concept in another ontology. We introduce several polynomial time algorithms for finding suboptimal solutions to the problem. We experimentally compare the algorithms using several synthetic as well as real-world ontologies and mappings.

R. Meersman et al. (Eds.): OTM 2010, Part II, LNCS 6427, pp. 999–1008, 2010.

2 Problem Description

We first introduce notations and definitions that we use in formalizing the problem of identifying the maximum subset of consistent mappings between hierarchical ontologies.

Definition 1 (Concept and Relationships). *A* concept *(or a* class*) represents a collection of objects or individuals. We consider two types of relationships between any pair of concepts:(1) strict subconcept relation ($\prec$) and equivalence relation($\equiv$).*

Definition 2 (Ontology). *An* ontology $\mathcal{O}_{\mathsf{x}} \colon \langle \mathbb{C}_{\mathsf{x}},\ \mathbb{R}_{\mathsf{x}} \rangle$ *where* $\mathsf{x} \in \mathbb{N}$ *is a two-tuple of a non-empty finite set of concepts* $\mathbb{C}_{\mathsf{x}}$ *and a finite set of relationships* $\mathbb{R}_{\mathsf{x}}$ *between those concepts.*

In our setting each ontology can be represented as a directed graph (called the *ontology graph*) where the each vertex is a set of equivalent concepts in the transitive closure of the ontology and a directed edge exists between two nodes in the graph if and only if a relationship exists between the concepts associated with two nodes (Note the edge is directed from the subconcept towards the superconcept).

Definition 3 (Ontology Graph). *The ontology graph associated with a given ontology* $\mathcal{O}_{\mathsf{x}} \colon \langle \mathbb{C}_{\mathsf{x}},\ \mathbb{R}_{\mathsf{x}} \rangle$ *is the graph* $\mathcal{G}_{\mathcal{O}_{\mathsf{x}}} \colon \langle \mathbb{V}_{\mathcal{O}_{\mathsf{x}}},\ \mathbb{E}_{\mathcal{O}_{\mathsf{x}}} \rangle$ *where* $\mathbb{V}_{\mathcal{O}_{\mathsf{x}}} = \{v_1^x, v_2^x \ldots\}$ *is a non-empty finite set of vertices and each vertex* v_i^x *represents a non-empty finite set of* equivalent concepts *in* $\mathbb{C}_{\mathsf{x}}$ *such that exactly one of the following conditions is true: (1) either* v_i^x *is a singleton set and the only concept contained in this set is not related to any other concept in* $\mathbb{C}_{\mathsf{x}}$ *with the equivalence relation or (2) Cardinality of* v_i^x *is more than* 1 *and each concept in* v_i^x *is related to at least one other concept in* v_i^x *with the equivalence relation.* $\mathbb{E}_{\mathcal{O}_{\mathsf{x}}}$ *is a finite set of edges such that if* $\left(e_p^{\mathsf{x}} = v_i^{\mathsf{x}} \dashrightarrow v_j^{\mathsf{x}}\right)$ *then,* $\exists c_{i_m}^{\mathsf{x}} \in v_i^{\mathsf{x}}, \exists c_{j_n}^{\mathsf{x}} \in v_j^{\mathsf{x}} \colon c_{i_m}^{\mathsf{x}} \prec c_{j_n}^{\mathsf{x}} \in \mathbb{R}_{\mathsf{x}}$.

Note that each concept in the ontology is associated with one and only one vertex in its associated ontology graph. In addition, if any two concepts are related through an equivalence relation they must have the same associated node in the ontology graph.

Remark 1 (Simplifying Assumption). Given an ontology $\mathcal{O}_{\mathsf{x}} \colon \langle \mathbb{C}_{\mathsf{x}},\ \mathbb{R}_{\mathsf{x}} \rangle$, it is fairly straightforward to generate the corresponding ontology graph $\mathcal{G}_{\mathcal{O}_{\mathsf{x}}} \colon \langle \mathbb{V}_{\mathcal{O}_{\mathsf{x}}},\ \mathbb{E}_{\mathcal{O}_{\mathsf{x}}} \rangle$. Note that we are actually merging all the equivalent classes in an ontology into a single vertex in the corresponding ontology graph. Since this merging is just the conceptual merging in the representation and not the actual merging of classes in the ontology, for simplicity, we will assume that the input ontologies do not contain any equivalence relationships. As a result, all the relationships contained in an ontology are subconcept relationships (hierarchical ontologies). This implies that all the vertices in the corresponding ontology graph have a single concept associated with them.

Definition 4 (Inconsistent). *An ontology* $\mathcal{O}_{\mathsf{x}} \colon \langle \mathbb{C}_{\mathsf{x}},\ \mathbb{R}_{\mathsf{x}} \rangle$ *is said to be* inconsistent*, if the transitive closure of* $\mathbb{R}_{\mathsf{x}}$ *contains* $c_i^{\mathsf{x}} \prec c_i^{\mathsf{x}}$.

Each subset of relationships in $\mathbb{R}_\mathsf{x}$ that lead to some relationship $c_i^\mathsf{x} \prec c_i^\mathsf{x}$ in the transitive closure of the ontology is said to be a set of **conflicting relationships**. For any inconsistent ontology there may be one or more sets of conflicting relationships and each set may contain two or more relationships that conflict as a whole.

Definition 5 (Consistent). *An ontology* $\mathcal{O}_\mathsf{x}\colon \langle \mathbb{C}_\mathsf{x},\ \mathbb{R}_\mathsf{x}\rangle$ *is said to be* consistent *if it is not inconsistent.*

Theorem 1. *An ontology* $\mathcal{O}_\mathsf{x}$ *is consistent if and only if its ontology graph* $\mathcal{G}_{\mathcal{O}_\mathsf{x}}$ *is a directed-acyclic graph (DAG).*

Proof. The proof (for the if direction) is by contradiction and follows from the fact that there is one-to-one mapping between concepts and relationships in ontology to vertices and edges in ontology graph. As such any concept associated with a vertex participating in a cycle will have a subconcept relationship with itself in the transitive closure of $\mathbb{R}_\mathsf{x}$. The reverse direction of the statement follows in a similar way.

Definition 6 (consistent**).** *We define function* consistent: $\mathbb{O} \longrightarrow \{\top, \bot\}$ *as:*

$$\text{consistent}\left(\mathcal{O}_\mathsf{x}\right) = \begin{cases} \top & \text{if } \mathcal{G}_{\mathcal{O}_\mathsf{x}} \text{ is DAG} \\ \bot & \text{otherwise} \end{cases}$$

Therefore, the function consistent *can be implemented using the topological ordering algorithm with a running time of* $O\left(|\mathbb{V}| + |\mathbb{E}|\right)$ *where* $|\mathbb{V}|$ *is the number of vertices and* $|\mathbb{E}|$ *is the number of edges in the graph [11]. Alternatively, we have a running time of* $O\left(|\mathbb{C}| + |\mathbb{R}|\right)$ *where* $|\mathbb{C}|$ *is the number of concepts and* $|\mathbb{R}|$ *is the number of relationships in the ontology.*

Given any two different ontologies $\mathcal{O}_\mathsf{x}\colon \langle \mathbb{C}_\mathsf{x},\ \mathbb{R}_\mathsf{x}\rangle$ and $\mathcal{O}_\mathsf{y}\colon \langle \mathbb{C}_\mathsf{y},\ \mathbb{R}_\mathsf{y}\rangle$ the Mappings define the relationships between any two concepts in the ontologies.

Definition 7 (Mapping Relationship). *Given* $\mathcal{O}_\mathsf{x}$ *and* $\mathcal{O}_\mathsf{y}$*, a* mapping relationship $r^{\mathsf{x:y}}$ *is a relationship* $c_i^\mathsf{x}\mathcal{R}c_j^\mathsf{y}$ *where* $c_i^\mathsf{x} \in \mathbb{C}_\mathsf{x}$*,* $c_j^\mathsf{y} \in \mathbb{C}_\mathsf{y}$*, and (1)* $c_i^\mathsf{x} \prec c_j^\mathsf{y}$ *represents that* c_i^x *is subconcept of* c_j^y*; (2)* $c_i^\mathsf{x} \succ c_j^\mathsf{y}$ *represents that* c_i^x *is superconcept of* c_j^y *and (3)* $c_i^\mathsf{x} \equiv c_j^\mathsf{y}$ *represents that* c_i^x *is equivalent to* c_j^y*.*

Definition 8 (Mapping Set). *Given* $\mathcal{O}_\mathsf{x}$ *and* $\mathcal{O}_\mathsf{y}$*, a* mapping set $\mathbb{M}_{\mathsf{x:y}}$ *is a finite set of mapping relationships.*

Given ontologies $\mathcal{O}_\mathsf{x}$, $\mathcal{O}_\mathsf{y}$, and mapping set $\mathbb{M}_{\mathsf{x:y}}$, we can generate a *merged ontology* by using the relationships specified in the mapping set $\mathbb{M}_{\mathsf{x:y}}$. We denote the *merged ontology* by $\mathcal{O}_\mathsf{z}\colon \langle \mathbb{C}_\mathsf{z},\ \mathbb{R}_\mathsf{z}\rangle$. In addition we use the following notations: $|\mathbb{C}| = |\mathbb{C}_\mathsf{x}| + |\mathbb{C}_\mathsf{y}|$, $|\mathbb{R}| = |\mathbb{R}_\mathsf{x}| + |\mathbb{R}_\mathsf{y}|$, and $|\mathbb{M}| = |\mathbb{M}_{\mathsf{x:y}}|$.

This merged ontology can be represented by a graph called *mapping graph*. Given a pair of ontologies $\mathcal{O}_\mathsf{x}$, $\mathcal{O}_\mathsf{y}$ and a mapping set $\mathbb{M}_{\mathsf{x:y}}$, the corresponding *mapping graph* is $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}\colon \left\langle \mathbb{V}_{\mathbb{M}_{\mathsf{x:y}}},\ \mathbb{E}_{\mathbb{M}_{\mathsf{x:y}}}\right\rangle$ where,

Table 1. Relationships in Mapping Graph

Relationship	In Mapping Graph	Edge Type
$c_i^{\mathsf{x}} \prec c_j^{\mathsf{x}}$	$v_i^{\mathsf{x}} \dashrightarrow v_j^{\mathsf{x}}$	Ontology edge
$c_i^{\mathsf{x}} \prec c_j^{\mathsf{y}}$	$v_i^{\mathsf{x}} \longrightarrow v_j^{\mathsf{y}}$	Mapping edge
$c_i^{\mathsf{x}} \succ c_j^{\mathsf{y}}$	$v_j^{\mathsf{y}} \longrightarrow v_i^{\mathsf{x}}$	Mapping edge
$c_i^{\mathsf{x}} \equiv c_j^{\mathsf{y}}$	$v_i^{\mathsf{x}} \longleftrightarrow v_j^{\mathsf{y}}$	Mapping edge

- $\mathbb{V}_{\mathbb{M}_{\mathsf{x:y}}}$ is finite non-empty set of vertices
- $\mathbb{E}_{\mathbb{M}_{\mathsf{x:y}}}$ is finite set of labeled directed edges such that each edge is one of the following:
  - an *ontology edge* corresponding to some subconcept relationship specified in either ontology.
  - an *mapping edge* corresponding to some subconcept or superconcept relationship specified in mapping set.
  - an *mapping edge* corresponding to some equivalence relationship specified in mapping set.

  Table 1 lists down how each relationship is being represented in the mapping graph.

The *mapping graph* intuitively differs from the *ontology graph* in the sense that it allows to distinguish between the edges corresponding to the relationships in the ontologies and the edges corresponding to the mapping relationships. In addition it can contain bidirectional edges corresponding to the equivalence relationships in the mapping set.

Remark 2 (Converting Mapping Graph to Ontology Graph). A mapping graph $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ can be easily converted to the corresponding ontology graph $\mathcal{G}_{\mathcal{O}_{\mathbb{M}_{\mathsf{x:y}}}}$ by first copying all the non-mapping edges and unidirectional mapping edges to the ontology graph and then merging the vertices that are combined by the bidirectional mapping edges.

Theorem 2. *A merged ontology $\mathcal{O}_{\mathbb{M}_{\mathsf{x:y}}}$ is consistent if its mapping graph $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ is a DAG.*

Proof. This is straightforward since the mapping graph $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ can be converted to ontology graph $\mathcal{G}_{\mathcal{O}_{\mathbb{M}_{\mathsf{x:y}}}}$ which in turn must be a DAG for the combined ontology $\mathcal{O}_{\mathbb{M}_{\mathsf{x:y}}}$ to be consistent as per Theorem 1.

Theorem 3. *Given consistent ontologies $\mathcal{O}_{\mathsf{x}}$, $\mathcal{O}_{\mathsf{y}}$, and a mapping set $\mathbb{M}_{\mathsf{x:y}}$, any cycle in $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ must contain at least two edges corresponding to the mapping relationships, that is, at least two edges in any cycle must belong to the set $\left(\mathbb{E}_{\mathbb{M}_{\mathsf{x:y}}} \setminus \left(\mathbb{E}_{\mathcal{O}_{\mathsf{x}}} \bigcup \mathbb{E}_{\mathcal{O}_{\mathsf{y}}}\right)\right)$. Moreover, those edges must be either ($v_i^{\mathsf{x}} \longrightarrow v_j^{\mathsf{y}}$ and $v_l^{\mathsf{y}} \longrightarrow v_k^{\mathsf{x}}$) or ($v_i^{\mathsf{x}} \longrightarrow v_j^{\mathsf{y}}$ and $v_k^{\mathsf{x}} \longleftrightarrow v_l^{\mathsf{y}}$) or ($v_i^{\mathsf{x}} \longleftrightarrow v_j^{\mathsf{y}}$ and $v_l^{\mathsf{y}} \longrightarrow v_k^{\mathsf{x}}$) or ($v_i^{\mathsf{x}} \longleftrightarrow v_j^{\mathsf{y}}$ and $v_k^{\mathsf{x}} \longleftrightarrow v_l^{\mathsf{y}}$.*

Proof outline. Since the given ontologies are consistent, they must be both DAGs by Theorem 1. Moreover, for the vertices corresponding to these DAGs to participate in a cycle in the mapping graph, there must be a path that goes from one DAG to the other DAG and back to the first DAG. Consequentially, two directed edges with opposite directions must exist between vertices in the two DAGs. Since the edges between the two DAGs are due to the mapping set, at least two of the edges in any cycle must be the mapping edges as specified above.

Remark 3. We can observe that each cycle can in turn be represented as a chain of relationships which will lead to inconsistency in the ontology. Hence, any such chain would contain at least two mapping relationships which must be either ($c_i^{\mathsf{x}} \prec c_j^{\mathsf{y}}$ and $c_k^{\mathsf{x}} \succ c_l^{\mathsf{y}}$) or ($c_i^{\mathsf{x}} \prec c_j^{\mathsf{y}}$ and $c_k^{\mathsf{x}} \equiv c_l^{\mathsf{y}}$) or ($c_i^{\mathsf{x}} \equiv c_j^{\mathsf{y}}$ and $c_k^{\mathsf{x}} \succ c_l^{\mathsf{y}}$) or ($c_i^{\mathsf{x}} \equiv c_j^{\mathsf{y}}$ and $c_k^{\mathsf{x}} \equiv c_l^{\mathsf{y}}$).

Definition 9 (Consistent Mapping Subset). *Given consistent ontologies $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, and a mapping set $\mathbb{M}_{\mathsf{x:y}}$, a subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ is said to be a* consistent mapping subset *if* consistent $\left(\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}\right) = \top$.

Definition 10 (Maximal Consistent Mapping Subset). *Given $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, and $\mathbb{M}_{\mathsf{x:y}}$, a consistent mapping subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ is said to be a* maximal consistent mapping subset *if* $\left(\text{consistent}\left(\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}\right) = \top\right)$ *and* $\forall r_p^{\mathsf{x:y}} \in \mathbb{M}_{\mathsf{x:y}} \setminus \mathbb{M}'_{\mathsf{x:y}}$, $\left(\text{consistent}\left(\mathcal{O}_{\mathbb{M}''_{\mathsf{x:y}}}\right) = \bot\right)$ *where* $\mathbb{M}''_{\mathsf{x:y}} = \left(\mathbb{M}'_{\mathsf{x:y}} \bigcup \left\{r_p^{\mathsf{x:y}}\right\}\right)$

Intuitively, a consistent mapping set is maximal if adding any more mapping to it will lead to inconsistencies. Note that for any given pair of consistent ontologies and a set of mappings between them there can be, in general, multiple maximal consistent mapping subsets. Furthermore, the user might have preferences over the mappings to be retained in the solution. A simple way to specify such preferences is by assigning weights to the individual mappings.

Definition 11 (Weighted Mapping Set). *Given a pair of ontologies $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, a* weighted mapping set $\mathbb{M}_{\mathsf{x:y}}$ *is a set of mapping relationships along with a weight function $\omega\colon \mathbb{M}_{\mathsf{x:y}} \longrightarrow (R_{\geq 0})$ where $(R_{\geq 0})$ is the set of positive reals.*

We will often use shorthand notation $\omega^{\Sigma}\left(\mathbb{M}_{\mathsf{x:y}}\right)$ to represent sum of weights of all relationships in $\mathbb{M}_{\mathsf{x:y}}$. In the rest of the paper, we will assume weighted mapping sets. If the ω function is not specified, we will assume each mapping has a unit weight.

Definition 12 (Maximum Consistent Mapping Subset). *Given a pair of ontologies $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, a weighted mapping set $\mathbb{M}_{\mathsf{x:y}}$, a* maximal *consistent mapping subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ is said to be a **maximum consistent mapping subset** if $\forall \mathbb{M}''_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}} : \omega^{\Sigma}\left(\mathbb{M}'_{\mathsf{x:y}}\right) \geq \omega^{\Sigma}\left(\mathbb{M}''_{\mathsf{x:y}}\right)$ where $\mathbb{M}''_{\mathsf{x:y}}$ is a maximal consistent mapping subset.*

So a maximum consistent mapping subset is a *maximal consistent mapping subset* with the most weight. Again, for any given pair of consistent ontologies and a weighted mapping set between those two ontologies, there can be multiple maximum consistent mapping subsets.

Definition 13 (Feedback Arc Set (FAS)). *Given a directed graph* $\mathcal{G}\colon \langle \mathbb{V},\ \mathbb{A}\rangle$ *where* $\mathbb{V}$ *is the set of vertices and* $\mathbb{A}$ *is the set of directed edges (arcs),* feedback arc set *of* $\mathcal{G}$ *is a subset of edges,* $\mathbb{A}' \subseteq \mathbb{A}$ *such that* $\mathcal{G}'\colon \langle \mathbb{V},\ \mathbb{A} \setminus \mathbb{A}'\rangle$ *is acyclic.*

We now introduce the definition of *Minimum Feedback Arc Set in Bipartite Tournament MFASBT* [8].

Definition 14 (MFASBT). *Given a bipartite tournament graph* $\mathcal{G}\colon \langle \mathbb{X}, \mathbb{A}, \mathbb{Y}\rangle$ *where* $\mathbb{X}$ *and* $\mathbb{Y}$ *are bipartite sets of vertices and* $\mathbb{A}$ *is set of directed arcs between the bipartite and a number* $k \in \mathbb{N}$*, is there a feedback arc set of size at most* k*, that is, is there a subset* $\mathbb{A}' \subseteq \mathbb{A}$ *such that* $|\mathbb{A}'| \leq k$ *and* $\mathcal{G} \setminus \mathbb{A}'$ *is acyclic?*

2.1 Problem Formulation

We now formally introduce the problem we are trying to solve in terms of the notation and definitions introduced above.

Decision Version (McM$_d$)**:** Given two consistent ontologies $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, some weighted mapping set $\mathbb{M}_{\mathsf{x:y}}$, and a number $k \in \mathbb{N}$, is there a maximal consistent mapping subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ of weight at least k? Formally, McM$_d$ can be stated as:

Given $\mathcal{O}_{\mathsf{x}}$, $\mathcal{O}_{\mathsf{y}}$, $\mathbb{M}_{\mathsf{x:y}}$, and some $k \in \mathbb{N}$, is there $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ such that all the following are true:

1. consistent $\left(\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}\right) = \top$.
2. $\forall r_p^{\mathsf{x:y}} \in \mathbb{M}_{\mathsf{x:y}} \backslash \mathbb{M}'_{\mathsf{x:y}}\colon \left(\text{consistent}\left(\mathcal{O}_{\mathbb{M}''_{\mathsf{x:y}}}\right) = \bot\right)$ where $\mathbb{M}''_{\mathsf{x:y}} = \left(\mathbb{M}'_{\mathsf{x:y}} \bigcup \left\{r_p^{\mathsf{x:y}}\right\}\right)$.
3. $\omega^{\Sigma}\left(\mathbb{M}'_{\mathsf{x:y}}\right) \geq k$.

Optimization Version (McM)**:** Given two consistent ontologies $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$, and some weighted mapping set $\mathbb{M}_{\mathsf{x:y}}$, identify a maximum consistent mapping subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$. Formally, McM can be stated as:

Given $\mathcal{O}_{\mathsf{x}}$, $\mathcal{O}_{\mathsf{y}}$, and $\mathbb{M}_{\mathsf{x:y}}$, find a subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ such that all the following are true:

1. consistent $\left(\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}\right) = \top$.
2. $\forall r_p^{\mathsf{x:y}} \in \mathbb{M}_{\mathsf{x:y}} \backslash \mathbb{M}'_{\mathsf{x:y}}\colon \left(\text{consistent}\left(\mathcal{O}_{\mathbb{M}''_{\mathsf{x:y}}}\right) = \bot\right)$ where $\mathbb{M}''_{\mathsf{x:y}} = \left(\mathbb{M}'_{\mathsf{x:y}} \bigcup \left\{r_p^{\mathsf{x:y}}\right\}\right)$.
3. $\forall \mathbb{M}''_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}\colon \omega^{\Sigma}\left(\mathbb{M}'_{\mathsf{x:y}}\right) \geq \omega^{\Sigma}\left(\mathbb{M}''_{\mathsf{x:y}}\right)$ where $\mathbb{M}''_{\mathsf{x:y}}$ is a maximal consistent mapping subset.

3 Eliminating Inconsistent Mappings

We first show that the decision problem McM$_d$ is NP-complete. We then proceed to describe several efficient algorithms for finding sub-optimal solutions to the optimization version of the problem.

Theorem 4. McM$_d$ *is NP-complete.*

Proof Outline. The complete details of the proof are omitted for lack of space but main idea of the proof to reduce problem of Minimum Feedback Arc Set in Bipartite Tournament (MFASBT) [8], a known NP-completeproblem, to a special instance of the decision problem McM_d.

Theorem 5. McM *is NP-hard.*

Proof. The proof follows from Theorem 4.

Since McM is NP-hard, we consider several sub-optimal solutions to the problem. Specifically, we relax McM by dropping the 3^{rd} condition in its definition to obtain McM_m.

Definition 15 (McM_m). *Given a pair of consistent ontologies $\mathcal{O}_\mathsf{x}$ and $\mathcal{O}_\mathsf{y}$ and a mapping set $\mathbb{M}_{\mathsf{x:y}}$, find a subset $\mathbb{M}'_{\mathsf{x:y}} \subseteq \mathbb{M}_{\mathsf{x:y}}$ such that all the following are true:*

1. $\text{consistent}\left(\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}\right) = \top$
2. $\forall r_p^{\mathsf{x:y}} \in \mathbb{M}_{\mathsf{x:y}} \setminus \mathbb{M}'_{\mathsf{x:y}} : \left(\text{consistent}\left(\mathcal{O}_{\mathbb{M}''_{\mathsf{x:y}}}\right) = \bot\right)$ *where* $\mathbb{M}''_{\mathsf{x:y}} = \left(\mathbb{M}'_{\mathsf{x:y}} \bigcup \left\{r_p^{\mathsf{x:y}}\right\}\right)$

We consider three polynomial time algorithms for McM_m which provide sub-optimal solutions to McM.

3.1 Naïve Approach

A simple approach to identifying a maximal consistent subset of $\mathbb{M}_{\mathsf{x:y}}$ is to start with the set $\mathbb{M}'_{\mathsf{x:y}}$, initially empty, and add to it mappings from $\mathbb{M}_{\mathsf{x:y}} \setminus \mathbb{M}'_{\mathsf{x:y}}$ one at a time, in decreasing order of their weights, as long as the combined ontology $\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}$ remains consistent. The correctness of the algorithm follows by construction and a straightforward analysis shows that the running time of the algorithm is $O\big(|\mathbb{M}|\,(|\mathbb{C}| + |\mathbb{R}| + |\mathbb{M}|)\big)$ (Recall $|\mathbb{C}| = |\mathbb{C}_\mathsf{x}| + |\mathbb{C}_\mathsf{y}|$, $|\mathbb{R}| = |\mathbb{R}_\mathsf{x}| + |\mathbb{R}_\mathsf{y}|$, and $|\mathbb{M}| = |\mathbb{M}_{\mathsf{x:y}}|$).

3.2 Biased Approach

It follows from Remark 3 that given a pair of strict partial order ontologies $\mathcal{O}_\mathsf{x}$, $\mathcal{O}_\mathsf{y}$ and a mapping set $\mathbb{M}_{\mathsf{x:y}}$ the resulting combined ontology is guaranteed to be consistent if $\mathbb{M}_{\mathsf{x:y}}$ contains only mappings relations of type $\prec$ or of type $\succ$. This is because in such a case the associated mapping graph is guaranteed to be acyclic. This observation leads to the following *biased* algorithm: Let $M_\prec$ and $M_\succ$ be the subsets of mapping relations (respectively) of type $\prec$ and $\succ$ in $\mathbb{M}_{\mathsf{x:y}}$. We start with the subset $\mathbb{M}'_{\mathsf{x:y}}$ initialized to one of the mapping subset that has the larger weight among $M_\prec$ and $M_\succ$. We add to $\mathbb{M}'_{\mathsf{x:y}}$, mappings one at a time, from $\mathbb{M}_{\mathsf{x:y}} \setminus \mathbb{M}'_{\mathsf{x:y}}$, in decreasing order of their weights, as long as the combined ontology $\mathcal{O}_{\mathbb{M}'_{\mathsf{x:y}}}$ remains consistent. The correctness of the algorithm follows by construction and a straightforward analysis shows that the running time of the algorithm is also $O\big(|\mathbb{M}|\,(|\mathbb{C}| + |\mathbb{R}| + |\mathbb{M}|)\big)$.

3.3 Graph-Based Approach

We proceed to describe a solution for McM_m that is inspired by an algorithm introduced by Demetrescu and Finocchi for finding a feedback arc set in a weighted directed graph [2]. We start with the mapping graph $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ constructed from a given pair of ontologies $\mathcal{O}_{\mathsf{x}}$, $\mathcal{O}_{\mathsf{y}}$ and a set of mappings $\mathbb{M}_{\mathsf{x:y}}$. Recall that the edges in $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ are of two types: (1) those corresponding to relations in $\mathcal{O}_{\mathsf{x}}$ and $\mathcal{O}_{\mathsf{y}}$; and (2) the edges corresponding to mappings in $\mathbb{M}_{\mathsf{x:y}}$. The basic idea is to identify and remove a minimal feedback arc set in $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ while preserving all the edges of type (1). We construct $G_{\mathcal{W}}$, a weighted version of $\mathcal{G}_{\mathbb{M}_{\mathsf{x:y}}}$ by assigning *infinite* weights to edges of type (1) and weights that reflect user preferences for mappings in $\mathbb{M}_{\mathsf{x:y}}$ to edges of type (2). The edges in a minimal feedback arc set of $G_{\mathcal{W}}$ correspond precisely to the mappings that need to be removed from $\mathbb{M}_{\mathsf{x:y}}$ in order to obtain a maximal consistent mapping subset of $\mathbb{M}_{\mathsf{x:y}}$. As computed by Demetrescu and Finocchi, the worst runtime complexity of this algorithm is $O\left(|\mathbb{V}|\,|\mathbb{E}|\right)$ which in our case turns out to be $O\Big(|\mathbb{C}|\,\left(|\mathbb{R}|+|\mathbb{M}|\right)\Big)$. Moreover this algorithm guarantees an approximation ratio bounded by the length (in terms of number of edges) of the longest simple cycle in the graph. We note that Demetrescu and Finocchi's algorithm for MFASBT works by removing edges with the minimum weight. Consequently, we need a procedure to assign weights to the edges in the Ontology Graph. To ensure that only the mapping edges are removed in our solution we assign an *infinite* weight to all the non-mapping edges. Approaches to assign weights to non-mapping edges include assigning unit weight to all non-mapping edges. The user preference of mappings can be incorporated by assigning user defined weights to the mappings.

4 Evaluation and Results

Since there are no benchmarks available for evaluating the performance of algorithms for identifying maximal consistent subsets of mappings between ontologies, we used randomly generated ontologies and mapping sets. We chose parameters of the distribution used to generate the ontologies based on a survey of real world ontologies published by Wang et. al. [15]. We say that an algorithm outperforms another if the cardinality of the solution (maximal consistent mapping subset) returned by it exceeds that of the solution returned by the other.

Our first set of experiments were designed to compare the sub-optimal solutions computed by the Naïve, Biased and Graph based approaches with each other. The graph-based approach yielded the best result on 88% of the cases; The Biased approach provided the best result in 7% of the cases; and the Naïve approach yielded the best result 5% of the cases.

Our second set of experiments focused on the execution time of the algorithms on some real world ontologies and mappings taken from [4]. The results of these experiments are shown in Table 2. Interestingly, in this case, the graph based algorithm executes significantly faster than the other two algorithms. This can

Table 2. Avg. Execution Time (in *ms*) for Real World Ontologies and Mappings

Ontology Name	$\|\mathbb{C}\|$	$\|\mathbb{R}\|$	$\|\mathbb{M}\|$	Simple Naïve	Biased	Graph-based
animals (A, B)	17	19	9	7	2	1
people+pets (A, B)	116	147	58	100	90	2
russia (C, D)	225	243	86	264	285	2
russia (1, 2)	314	327	70	336	286	3
russia (A, B)	254	275	103	374	379	3
Sport (Soccer, Event)	570	558	148	1326	1332	8
Tourism (A, B)	814	850	190	2827	2754	124

be explained by the fact that the mappings used were specified by experts and contained few inconsistencies. In such a setting the resulting mapping graph contains few cycles and the graph based algorithm is able to terminate quickly. However, the other algorithms need to check for consistency whenever they try to add a mapping to the solution set, and hence, they end up performing poorly.

Our third set of experiments were designed to compare the best of these three approaches with the brute force approach. The latter exhaustively enumerates all possible subsets of $\mathbb{M}_{x:y}$, checks each of them for consistency and outputs a maximum consistent subset. To permit exhaustive enumeration, we had to limit the cardinality of the mapping set to a maximum of 31. We found that in 97% of the cases, the heurstic solution was as good as the optimal solution. Morever, in the cases where the heuristic solution differed from the optimal, the cardinality of the set of mappings in the heuristic solution differed from the optimal solution by atmost 2.

5 Summary and Discussion

We showed that the problem of identifying the maximum subset of consistent mappings between hierarchical ontologies is NP-hard. We explored introduce several polynomial time algorithms for finding suboptimal solutions to the problem in the restricted setting of mappings that assert that a concept in one ontology is a subconcept, superconcept, or equivalent concept of a concept in another ontology. These algorithms can be extended in a natural way to deal with more expressive mappings between hierarchical ontologies e.g., those that assert the equivalence of a concept in a target ontology with a concept that is formed by *Union*, *Intersection* or *Difference* of two or more concepts in the source ontology. The task of specifying the mappings has been extensively studied in the semantic web, database and the ontology communities in various contexts including *ontology mapping*, *ontology matching*, *ontology alignment*, *ontology merging* and *ontology integration* (see survey papers [1], [3],[9], and the book [6]). The Ontology Alignment Evaluation Initiative is a coordinated international initiative that is aimed at developing a set of common metrics and benchmarks for evaluating ontology alignment methods [5]. A variety of techniques for automating the specification of mappings between ontologies have been explored. These include schema based approaches, instance based approaches or a combination of the two (see [6] for a detailed discussion of these methods). Of related interest is the work in learning from heterogeneous data sources in presence of mapping errors [12]. Ensuring consistency of mappings, specially in large ontologies, has been identified as one of the major challenges in ontology integration [14]. Barring a few exceptions ([13],[10],[7]) the problem of eliminating inconsistent mappings has received limited attention in the literature. To the best of our knowledge, the the problem of finding the maximum consistent subset of mappings has not been studied.

References

1. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Rec. 35(3), 34–41 (2006)
2. Demetrescu, C., Finocchi, I.: Combinatorial algorithms for feedback problems in directed graphs. Inf. Process. Lett. 86, 2003 (2003)
3. Doan, A., Halevy, A.Y.: Semantic-integration research in the database community. AI Mag. 26(1), 83–94 (2005)
4. Ehrig, M., Sure, Y.: Foam - a framework for ontology alignment and mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729. Springer, Heidelberg (2005)
5. Euzenat, J., et al.: Ontology alignment evaluation initiative (2010), `http://oaei.ontologymatching.org/`
6. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Berlin (2007)
7. Falconer, S.M., Storey, M.A.D.: A cognitive support framework for ontology mapping. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
8. Guo, J., Hüffner, F., Moser, H.: Feedback arc set in bipartite tournaments is np-complete. Inf. Process. Lett. 102(2-3), 62–65 (2007)
9. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowl. Eng. Rev. 18(1), 1–31 (2003)
10. Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C.: Repairing unsatisfiable concepts in owl ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006)
11. Kleinberg, J., Tardos, E.: Algorithm Design. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (2005)
12. Koul, N., Honavar, V.: Learning in presence of ontology mapping errors. In: Web Intelligence (2010)
13. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: AAAI, pp. 1408–1413 (2007)
14. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1164–1182. Springer, Heidelberg (2008)
15. Wang, T.D., Parsia, B., Hendler, J.: A survey of the web ontology landscape. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 682–694. Springer, Heidelberg (2006)