More Than the Sum of Its Parts – Holistic Ontology Alignment by Population-Based Optimisation

Jürgen Bock[1], Sebastian Rudolph[2], and Michael Mutter[1]

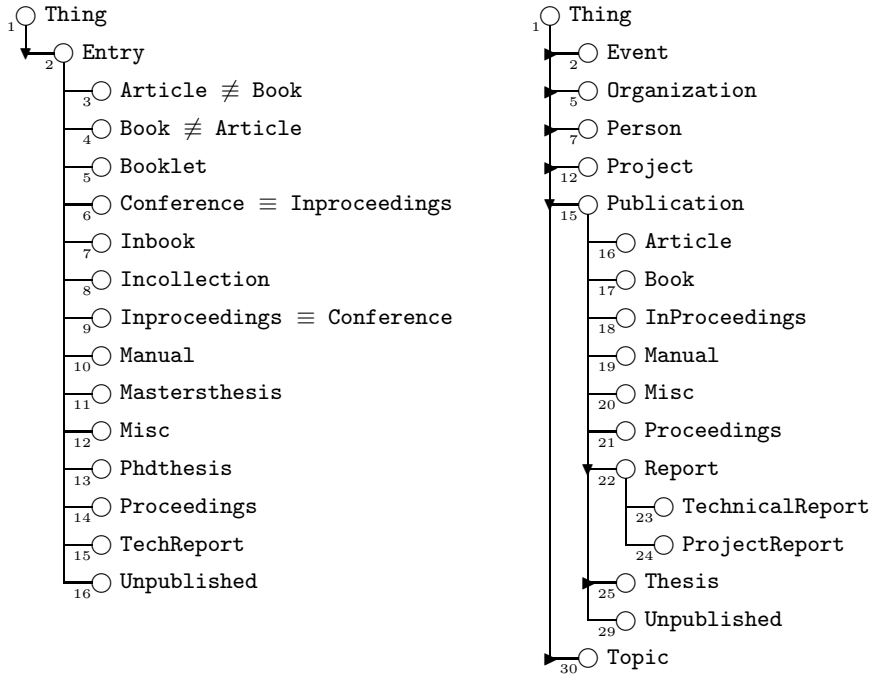
[1] FZI Research Center for Information Technology
Karlsruhe, Germany
{bock,mutter}@fzi.de
[2] Karlsruhe Institue of Technology
Karlsruhe, Germany
rudolph@kit.edu

Abstract. Ontology alignment is a key challenge to allow for interoperability between heterogeneous semantic data sources. Today, most algorithms extract an alignment from a matrix of the pairwise similarities of ontological entities of two ontologies. However, this standard approach has severe disadvantages regarding scalability and is incapable of accounting for global alignment quality criteria that go beyond the aggregation of independent pairwise correspondence evaluations. This paper considers the ontology alignment problem as an optimisation problem that can be addressed using nature-inspired population-based optimisation heuristics. This allows for the deployment of an objective function which can be freely defined to take into account individual correspondence evaluations as well as global alignment constraints. Moreover, such algorithms can easily be parallelised and show anytime behaviour due to their iterative nature. The paper generalises an existing approach to the alignment problem based on discrete particle swarm optimisation, and presents a novel implementation based on evolutionary programming. First experimental results demonstrate feasibility and scalability of the presented approaches.

1 Introduction

Ontology alignment is a key challenge to allow for interoperability between heterogeneous semantic data sources. In particular for more expressive ontologies, the optimal alignment is difficult to discover, since correspondences cannot be assessed in isolation as they can interfere with each other on the alignment level [18]. For instance, two classes that are in a subclass relationship in an ontology cannot correspond to two disjoint classes in another ontology as this causes an incoherency in case a correspondence is interpreted as semantic equality. Figure 1 shows two ontologies that might be candidates for being aligned. In this example, an alignment containing the correspondences a:`Article` ↔ b:`Article` and a:`Book` ↔ b:`Publication` would cause such an incoherency.

T. Lukasiewicz and A. Sali (Eds.): FoIKS 2012, LNCS 7153, pp. 71–90, 2012.



(a) MIT BibTEX ontology (modified excerpt) (b) Karlsruhe BibTEX ontology (modified excerpt)

Fig. 1. Example ontologies from the bibliography domain. Only class hierarchies and additional axioms for equivalence (≡) and disjointness (≢) are illustrated. Numbers at class nodes represent entity indices.

While most alignment systems disregard such global constraints and compute all pairwise entity similarities to extract an alignment from that matrix [22,12], there have been recent approaches that consider global quality criteria during the alignment computation process [21,4]. These systems, however, have problems with scalability when the ontologies become large, due to their exhaustive search algorithm for solving the constraint satisfaction problem.

We propose an approach that considers the ontology alignment problem an optimisation problem, and applies nature-inspired, population-based optimisation techniques for solving it. The ontology alignment problem is considered in a holistic manner, *i.e.* by taking local (correspondence level) and global (alignment level) quality criteria into account. Apart from this holistic consideration of ontology alignment, such optimisers are straightforwardly parallelisable [24,3] and show anytime behaviour, which makes them highly feasible for large scale alignment problems.

In previous work the ontology alignment problem was tackled by a *Particle Swarm Optimisation* algorithm [2]. This paper abstracts from this particular

metaheuristic and demonstrates the general applicability of population-based optimisation techniques. Following this generalisation, an alternative metaheuristic, *Evolutionary Programming*, is introduced for solving the ontology alignment problem. Two prototype alignment systems are presented and evaluated for the two approaches, resp.

In the following Sect. 2 notations of ontology alignment and population-based optimisation are introduced, including an analysis of the search space for the ontology alignment problem. Section 3 discusses ontology alignment as a holistic optimisation problem and how it can be encoded in order to be tackled by optimisation algorithms. Two population-based optimisation algorithms are applied for the ontology alignment problem in Sect. 4, and evaluated in Sect. 5. Section 6 discusses related work before the paper concludes in Sect. 7.

2 Preliminaries

This work applies methods from the field of computational intelligence to a problem in the field of knowledge representation. This section introduces the basic notions of the problem domain *ontology alignment*, and of the *population-based optimisation* techniques used for solving it.

2.1 Ontology Alignment

The problem of ontology alignment is a rather generic mapping task and not necessarily restricted to a specific ontology language like OWL. Abstracting as much as possible from specificities of representation formalisms, we formalise the notions of ontology and ontology alignment.

Definition 1. *An* ontology $\mathcal{O}$ *is a set of statements*[1] *that are referring to a set of* ontology entities $\{e_1, \ldots, e_n\}$*, also called the* vocabulary *of the ontology and denoted by* $\mathsf{voc}(\mathcal{O})$*. Moreover, each* $e_i \in \mathsf{voc}(\mathcal{O})$ *is associated an* entity type $\tau(e_i)$ *with* $\tau : \mathsf{voc}(\mathcal{O}) \to \mathcal{T}$ *where* $\mathcal{T}$ *is a fixed finite set of types. The* size *of an ontology is the number of entities it contains:* $\sharp\mathcal{O} = |\mathsf{voc}(\mathcal{O})|$*. Moreover, for any* $t \in \mathcal{T}$ *we let* $\sharp_t\mathcal{O} = |\{e \mid e \in \mathsf{voc}(\mathcal{O}), \tau(e) = t\}|$ *denote the number of ontology entities of a specific type* t*.*

This definition accounts for the fact that many ontology formalisms support multi-sorted vocabularies. For instance, in the case of OWL, we have $\mathcal{T} = \{\texttt{class}, \texttt{object\_property}, \texttt{data\_property}, \texttt{individual}\}$. The subdivision of the ontology into these different categories is important since we assume that two entities of two ontologies can only be interlinked by a correspondence if they have the same type, as made more precise in the subsequent definition.

[1] Note that we do not further specify, how exactly the statements (also referred to as axioms) of an ontology look like.

Definition 2. *Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, a* correspondence *between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a pair of entities $C = \langle e, f \rangle$, where $e \in \mathsf{voc}(\mathcal{O}_1)$ and $f \in \mathsf{voc}(\mathcal{O}_2)$ and $\tau(e) = \tau(f)$, i.e., e and f are of the same entity type.*

Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, an alignment *$A \subseteq \mathsf{voc}(\mathcal{O}_1) \times \mathsf{voc}(\mathcal{O}_2)$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$, such that*

- *for each $e \in \mathsf{voc}(\mathcal{O}_1)$ there is at most one $f \in \mathsf{voc}(\mathcal{O}_2)$ with $\langle e, f \rangle \in A$ and*
- *for each $f \in \mathsf{voc}(\mathcal{O}_2)$ there is at most one $e \in \mathsf{voc}(\mathcal{O}_1)$ with $\langle e, f \rangle \in A$.*

The size *of an alignment, $|A|$, is the number of correspondences it contains.*

Note that the condition in definition 2 restricts the size of an alignment to be less or equal the size of the smaller ontology.

To determine the size of the problem space, *i.e.* the number of possible alignments between two ontologies, we first consider the number of possible alignments $\mathcal{A}$ between two sets M and N of entities all having the same type. As already stated, such an alignment can contain between 0 and $\min(|M|, |N|)$ correspondences. Let us consider the number of alignments with exactly k correspondences. To fix one such alignment, we have to pick k elements from N and – independently – k elements from M. For each of these choices there are $k!$ different ways of arranging the picked sets into k non-overlapping pairs from $M \times N$. This leaves us with

$$\binom{|M|}{k} \cdot \binom{|N|}{k} \cdot k! = \frac{|M|!}{(|M| - k)!} \binom{|N|}{k} \tag{1}$$

alignments of size k. To obtain the count of all alignments of arbitrary size we have to sum over the alignment numbers for all possible k.

Finally, to determine the total number of alignments between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, we have to calculate the number of possible alignments for each type and obtain the number of possible combinations by multiplying the respective single-type alignment numbers. Thus we obtain for the number of total alignments:

$$\prod_{t \in \mathcal{T}} \sum_{k=0}^{\min(\sharp_t \mathcal{O}_1, \sharp_t \mathcal{O}_2)} \frac{\sharp_t \mathcal{O}_1!}{(\sharp_t \mathcal{O}_1 - k)!} \binom{\sharp_t \mathcal{O}_2}{k} \tag{2}$$

Thereby we obtain that the size of the problem space grows exponentially with the size of the ontologies.[2] This clearly implies that exhaustively searching the problem space for the optimal solution is infeasible for real-life ontologies.

2.2 Population-Based Optimisation Heuristics

The discipline of *Computational Intelligence* comprises various research categories out of which *Evolutionary Computation* [24] and *Computational Swarm*

[2] To be precise, we have to exclude the pathological case in which for every $t \in \mathcal{T}$ one of $\sharp_t \mathcal{O}_1$ or $\sharp_t \mathcal{O}_2$ remains constant.

Intelligence [14] are two relatively young areas that have already gained notable impact in several application domains.

This class of algorithms can be regarded in terms of their characteristics of being population-based metaheuristics, which have shown to be efficient for complex optimisation problems. In that respect they share the following characteristics.

Definition 3. *Let $\mathbb{P}$ be the problem space associated to some problem. Then, a* population *is defined as a pair $\langle I, p\rangle$ where I is a finite set of* individuals *and $p : I \to \mathbb{P}$ is a function assigning to every individual of the population a position in the problem space.*

An update *operation is a function U mapping populations to populations.*

An optimisation run *with respect to U is a finite sequence $\langle I_1, p_1\rangle, \ldots, \langle I_n, p_n\rangle$ of populations, where $\langle I_{i+1}, p_{i+1}\rangle = U(\langle I_i, p_i\rangle)$ for all $i \in \{1, \ldots, n-1\}$. The numbers $1, \ldots, n$ are normally referred to as* generations.

The generic execution of a population-based optimisation algorithm starts with an initialisation step, where the population $\langle I_1, p_1\rangle$ is initialised. Following this, the algorithm iterates through its generations, where in each generation i the individuals are evaluated according to the objective function. For each individual this evaluation results in a fitness score in the problem space. Furthermore, in each iteration, the update operator U creates a new population for the next generation, based on the current population's fitness scores. This process is repeated until a termination criterion is fulfilled, *e.g.* a maximum number of generations n or a prescribed fitness score is reached.

The algorithm-specific update function is one of the most distinguishing features of each paradigm. *E.g.* algorithms might require the size of I_i to remain constant throughout generations, *i.e.* for all $i, j \in \{1, \ldots, n\}$ holds $|I_i| = |I_j|$.

The remainder of this section presents the two population-based optimisation heuristics *Particle Swarm Optimisation* and *Evolutionary Programming*.

Particle Swarm Optimisation. Commonly classified under the category of computational swarm intelligence, *Particle Swarm Optimisation* [15,23] has a strong emphasis on the social aspects of individuals (particles) in the population (swarm). After initialisation, the population remains constant, *i.e.* no particle leaves or joins the swarm and thus $\forall i \in \{1, \ldots, n-1\} : I_{i+1} = I_i$. In its classical implementation, every particle remembers the best position in the problem space it has ever visited (*personal best*), and the swarm remembers the best position any particle has ever visited (*global best*). In each iteration i (generation[3]) the update operation applies a velocity vector $v_i(x_j)$ to the position $p_i(x_j)$ of each particle $x_j \in I_i, 1 \leq j \leq |I_i|$, changing its position in the problem space. The velocity vector composes of the personal and global best positions, as well as an inertia component [23]. The new population in the $(i+1)$th iteration thus

[3] Since the swarm consists of the same set of particles throughout the execution of the algorithm, the term "generation" is slightly misplaced in this context.

is $\langle I_{i+1}, p_{i+1}\rangle = U(\langle I_i, p_i\rangle) = \langle I_i, q_i\rangle$, where $\forall x_j \in I_i, 1 \leq j \leq |I_i| : q_i(x_j) = p_i(x_j) + v_i(x_j)$.

Evolutionary Programming. As one paradigm classified under the category of Evolutionary Algorithms, *Evolutionary Programming* [10,24] simulates species (individuals) that compete with each other in the problem space. As opposed to Genetic Algorithms, the update function in Evolutionary Programming does not involve a recombination operation for exchanging information between individuals in the population. However, each species creates an offspring by mutation, temporarily doubling the size of the population.

Formally, the update operation U is defined as follows: In a temporary population $\langle I_{i'}, p_{i'}\rangle$, for each $j \in \{1, \ldots, |I_i|\}$ let $x_{|I_i|+j} \in I_{i'}$ be the mutated species created by x_j. Thus, for all $k \in \{1, \ldots, |I_{i'}|\}$

$$p_{i'}(x_k) = \begin{cases} p_i(x_k) & \text{if } 1 \leq k \leq |I_i|, \\ q_i(x_k) & \text{if } (|I_i| + 1) \leq k \leq |I_{i'}| \end{cases}$$

where q_i maps a species to the new position after mutation.

In a subsequent selection step, half of that population becomes extinct, returning to the original size of the population. Survivors are typically determined by some sort of tournament selection, where species pairwise compete with other species, which results in a ranking that is used to select survivors of that generation. Formally, using a particular selection principle, the population in the $(i+1)$th generation is $\langle I_{i+1}, p_{i+1}\rangle$, such that $I_{i+1} \subseteq I_{i'}$ with $|I_{i+1}| = |I_i|$.

3 Ontology Alignment: A Holistic Optimisation Problem

Typical state-of-the-art ontology alignment systems follow the approach of pairwise entity comparison in order to generate alignments, as shown in Fig. 2. After some preprocessing operations, various basic matchers are applied to all $\sharp_t\mathcal{O}_1 \cdot \sharp_t\mathcal{O}_2$ pairs of entities from $\mathcal{O}_1$ and $\mathcal{O}_2$, for every $t \in \mathcal{T}$. The measures delivered by the basic matchers are combined and from the resulting similarity matrix an alignment is extracted by solving the resulting assignment problem. Typically this is done by variations of the Hungarian method [16]. In that approach correspondences are considered independently and the validity of the alignment with respect to any global constraints, such as inter-correspondence dependencies, can only be assessed after an actual alignment is extracted. Some systems deal with this situation by revisiting the similarity matrix when encountering the violation of such global constraints. The *ASMOV* system, for instance, applies a "semantic verification" by iteratively correcting the alignment [12].

The computation of the similarity matrix, alignment extraction, and iterative correction can become very costly when the ontologies are large. This section presents a holistic view on the alignment problem in terms of assessing the quality of an alignment in a way it cannot be assessed by considering its correspondences in isolation.

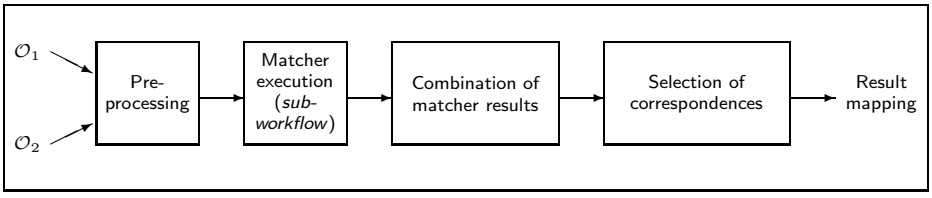


Fig. 2. General workflow for pairwise ontology matching (following Rahm [22])

3.1 Holistic Objective Function

A population-based optimisation algorithm as defined in Sect. 2.2 strives for finding the optimal argument for an objective function F, such that it becomes minimal (or maximal). In the case of ontology alignment, assume the objective function is minimal, iff an alignment is optimal. Thus the goal is to find

$$A^* = \operatorname{argmin}_{A \in \mathbb{P}} F(A) \tag{3}$$

for all possible candidate alignments A from the space $\mathbb{P}$ of possible alignments.

The objective function F evaluates an alignment as a whole, thus it goes beyond the pairwise evaluation of entity pairs, *i.e.* candidate correspondences. We can thus subdivide F into a *local* and a *global* part: $F(A) = F_{\text{loc}}(A) + F_{\text{glob}}(A)$ which are described in more detail in the following.

Local Aspects. Some criteria of whether correspondences should be contained in an alignment that can be evaluated for each correspondence in isolation, *i.e.* there is a function $f : \mathsf{voc}(\mathcal{O}_1) \times \mathsf{voc}(\mathcal{O}_2) \to \mathbb{R}$ assigning to every correspondence $\langle e_1, e_2 \rangle$ a local score $f(e_1, e_2)$. The total contribution F_{loc} of the local aspects to the objective function is then obtained by summing over the local scores of all entity pairs of the given alignment:

$$F_{\text{loc}}(A) = \sum_{\langle e_1, e_2 \rangle \in A} f(e_1, e_2)$$

Those local aspects comprise lexical and linguistic similarity measures of entity labels, comments, and other annotations that might be available depending on the vocabularies and ontology languages used. This is what most approaches aggregate into similarities collected in a similarity matrix of entity pairs, and what Niepert *et al.* [21] refer to as a-priori likelihoods.

Global Aspects. Some alignment quality criteria go beyond the superposition of independent evaluations on the correspondence level. We mention the most natural possibilities and for some examples show how we incorporate them into our objective function, for a comprehensive description of the objective function see our previous work [2].

Alignment Size. Typically there is not a complete overlap of ontologies. Thus it is important to determine the optimal size of an alignment. Since we naturally prefer larger alignments to smaller ones, we add a term $-\eta|A|$ to the objective function where η is a positive weighting factor.

Correspondence Correlations. Considering alignments from a structural point of view, correspondences can be evaluated differently depending on the presence of other correspondences in the alignment.

- Subsumption hierarchy: A correspondence between two classes is more likely to be correct, if the alignment also contains a correspondence of the classes' respective superclasses. The same holds for properties. If we denote the intra-ontological subclass/subproperty relationship with $\sqsubseteq$, the corresponding term of F_{glob} can be expressed by

$$\sum_{\langle e_1,e_2\rangle\in A} \sum_{\substack{\langle e_1',e_2'\rangle\in A \\ e_1\sqsubseteq e_1',e_2\sqsubseteq e_2'}} f(e_1',e_2').$$

  Note that although this term is represented as a sum over all correspondences it cannot be described locally, since the single summands again depend on the whole alignment A.
- Domain / range: A correspondence between two classes is more likely to be correct, if the alignment also contains correspondences between the properties of which these classes appear in the domain or range restriction. Likewise, a correspondence between two properties is more likely to be correct, if the alignment also contains correspondences between the classes that appear in their domain (or range) restrictions.
- Assertion axioms: A correspondence between two classes is more likely to be correct, if the alignment also contains correspondences between the individuals that are assigned to these classes. The same holds for property correspondences and correspondences of individuals asserted to the respective properties.

Alignment Coherence. Assuming a strict semantics of an alignment, it can cause incoherencies with respect to the ontologies it aligns [18]. The alignment criterion of coherency can only be assessed when considering the alignment as a whole.

3.2 Solution Representation

In the case of ontology alignment, each individual in the population represents a solution to the alignment problem, *i.e.* a candidate alignment. In order to efficiently converge towards the global optimum, an alignment must be represented such that it allows for

1. efficient manipulations of an alignment via the update operation U
2. efficient computation of a fitness score w.r.t. the objective function F.

In the following, two representations are presented, the *correspondence set* representation as the straightforward solution, and the *correspondence permutation* representation as an alternative, which tends to be more efficient.

Correspondence Set. The *correspondence set* representation is the straightforward solution that regards an alignment naturally in the way it is defined in definition 2, namely as a set of correspondences. This representation allows for easy assessment of the alignment regarding the objective function. However, it requires the update operation U to be aware of global alignment constraints, such that only valid candidate alignments are produced by U. In this case, for instance, only 1:1 alignments are considered, and thus U must be specified in a way that only valid 1:1 alignments are generated when updating an individual.

Correspondence Permutation. The *correspondence permutation* representation denotes a novel data structure that overcomes the problem of the correspondence set representation.

Definition 4. *For two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, let m be the number of entities of type t in $\mathcal{O}_1$, and let n be the number of entities of type t in $\mathcal{O}_2$. A* correspondence permutation *is a function $\pi : \{1, \ldots, m\} \to \{1, \ldots, n\} \cup \{\square\}$, such that for all $1 \leq i < j \leq m$ with $\pi(i), \pi(j) \neq \square$ holds $\pi(i) \neq \pi(j)$.*

Intuitively, a correspondence permutation can be understood as an array of m elements: $\pi(1)\ \pi(2)\ \cdots\ \pi(m)$. Thereby each of the numbers $1 \ldots n$ occurs at most once in this array, whereas $\square$ can occur arbitrarily often.

Assuming that the array index represents the entity index of (the smaller) ontology $\mathcal{O}_1$ and that the array element at position i denotes the index $\pi(i)$ of the entity in ontology $\mathcal{O}_2$, it is straightforward that each array entry represents a correspondence $\langle e_i, f_{\pi(i)} \rangle$ if $f_{\pi(i)} \neq \square$, and no correspondence for e_i otherwise. Figure 3 shows an example of a correspondence permutation that represents a possible alignment for the two ontologies from Fig. 1. The array index represents the entity index of the ontology from Fig. 1a, while the array elements represent entity indices of the ontology from Fig. 1b.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	15	16	17	□	□	□	□	18	19	□	20	25	21	23	29

Fig. 3. Example of a correspondence permutation using indices from Fig. 1

Since by definition 2 correspondences can only exist between entities of the same type, a solution representation would require $|\mathcal{T}|$ correspondence permutations for the different entity types. Following definition 3, let $\langle I, p \rangle$ be a population. Then for each $x \in I$, $p(x) = \pi_x$.

If for any $t \in \mathcal{T}$, $\sharp_t \mathcal{O}_1 \neq \sharp_t \mathcal{O}_2$, the ontologies should be swapped such that $m < n$ for increasing memory efficiency.

4 The Charm of Nature-Inspired Optimisation Heuristics

The two examples of population-based optimisation techniques presented in Sect. 2.2 have been applied to the ontology alignment problem and implemented in two prototypes[4]:

- Ontology **Map**ping using **P**article **S**warm **O**ptimisation [2] (*MapPSO*)
- Ontology **Map**ping using **Evo**lutionary Programming (*MapEVO*).

Both algorithms share the same objective function which is composed of various base matchers that evaluate correspondences with respect to local and global characteristics. While the former cover lexical and linguistic similarities of entity labels and comments, the latter comprises structural similarities that evaluate correspondences in the context of the whole alignment. More precisely this means that the quality of a correspondence is assessed according to the presence of other correspondences in the alignment, as discussed in Sect. 3.1.

4.1 Ontology Mapping Using Particle Swarm Optimisation

A discrete particle swarm optimisation algorithm has been developed in order to tackle the ontology alignment problem [2]. The algorithm maintains a population of particles, each representing a valid alignment using the *correspondence set* representation. Since the problem space $\mathbb{P}$ is discrete, the classical notion of velocity fails, since it is tailored for continuous optimisation problems[5]. This approach picks up the notion of velocity as proposed by Correa *et al.* [5] and refines it in several respects in order to facilitate a more direct convergence towards a stable optimal alignment [2].

4.2 Ontology Mapping Using Evolutionary Programming

An evolutionary programming algorithm has been developed in order to tackle the ontology alignment problem. The algorithm utilises a population of species, each representing a valid alignment using the *correspondence permutation* representation. The approach does not incorporate a social component in terms of a velocity influenced by other individuals in the population. Instead it eliminates bad solutions and reproduces good ones in periodic selection processes. In order to avoid that only few species dominate the population, this selection process is not done in every generation. Experiments have shown that for the maximum number of generations n and population size $|I|$, the best results are obtained when conducting the selection process every $n/|I|$ generations, then eliminating a fraction of $1/|I|$ of the species and allowing the same number to reproduce themselves. Let s be a selection function transforming a set of species

[4] Both prototypes are available at `http://sourceforge.net/projects/mappso/`.

[5] In traditional (continuous) particle swarm optimisation, a velocity vector is added to the current position of a particle in each iteration in order to change its position in the search space.

into another one, periodically as described above. There are two update operations u_s and u_e, such that $U(\langle I_i, p_i\rangle) = \langle I_{i+1}, p_{i+1}\rangle$ where $I_{i+1} = s(I_i)$ and $p_{i+1} = (u_e \circ u_s)(p_i)$. u_s and u_e are defined for the correspondence permutation data structure as follows. For the sake of simplicity let us consider only entities of type t. Let $n = \sharp_t \mathcal{O}_1$ and $m = \sharp_t \mathcal{O}_2$. Without loss of generality let $m < n$.

- The *switch* operator u_s transforms a correspondence permutation π into a new one π' by picking two indices i and j, such that $1 \leq i < j \leq m$ and $\pi(i), \pi(j) \neq \square$. Subsequently it transforms the correspondence permutation as follows by setting $\pi'(i) = \pi(j)$ and $\pi'(j) = \pi(i)$ as well as $\pi'(k) = \pi(k)$ for all $k \notin \{i, j\}$.

  Each index is selected for taking part in a switch with probability $P_s = 1 - x$, where $x \in [0, 1]$ is the evaluation of correspondence $\langle e_i, f_{\pi(i)}\rangle$ in its global alignment context. Given that an evaluation of 1 is best and an evaluation of 0 is worst, this decreases the chance for good correspondences to be switched.
- The *exchange* operator u_e also transforms a correspondence permutation π into a new one π'. Let $R = \{1, \ldots, n\} \setminus \{\pi(1), \ldots, \pi(m)\}$ an archive of entity indices of $\mathcal{O}_2$ not currently participating in a correspondence. For each $i \in \{1, \ldots, m\}$, $\pi' = u_e(\pi)$ such that

$$\pi'(i) = \begin{cases} \square & \text{with probability } P_\square \text{ if } \pi(i) \neq \square, \\ k \in R & \text{with probability } P_{\text{fill}} \text{ if } \pi(i) = \square, \\ k \in R & \text{with probability } P_{\text{ex}} \text{ if } \pi(i) \neq \square, \\ \pi(i) & \text{otherwise} \end{cases}$$

  The exchange probabilities $P_\square$, P_{fill}, and P_{ex} depend on the evaluation of correspondence $\langle e_i, f_{\pi(i)}\rangle$ in its global alignment context, as well as the progress of the optimisation run [20]. The latter criterion, for instance, decreases P_{fill} in order to reduce the chance of new correspondences being created towards the end of the optimisation run.

Figure 4 illustrates the application of the *switch* operator, followed by an application of the *exchange* operator. The example uses entity indices from the ontologies in Fig. 1.

4.3 Advantages of Population-Based Optimisation Heuristics

Apart from the holistic alignment evaluation, the use of population-based optimisation heuristics has several other advantages. Firstly, they are inherently parallelisable [24], since individuals can be evaluated independently on parallel computing nodes. The *MapPSO* prototype has been successfully deployed to the Amazon Web Services™ (AWS)[6] cloud computing service [3].

Anytime behaviour is another feature of population-based optimisation heuristics. An optimisation run can be interrupted at any time returning the best solution found so far. In runtime-critical alignment scenarios, this might be a desirable property.

[6] `http://aws.amazon.com/`

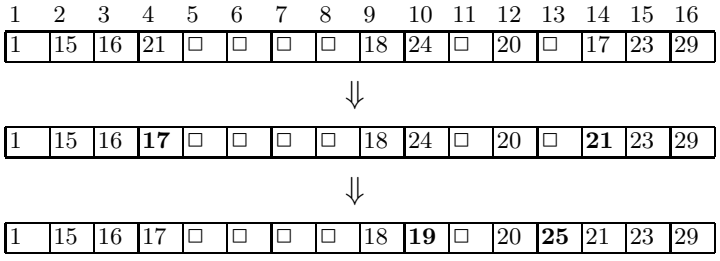


Fig. 4. Example of the application of the *switch* operator followed by an application of the *exchange* operator

5 Evaluation

Evaluating ontology alignment systems is a non-trivial task, since it requires the existence of high quality benchmarks [1], *i.e.* reference alignments for given pairs of ontologies. Such gold standards, however, do not exist for large ontologies, since they require human effort in order to verify their correctness. The Ontology Alignment Evaluation Initiative (OAEI)[7], however, does provide benchmarks for small ontologies.

The evaluation presented in this section is thus split into two parts. On the one hand there is an evaluation of the alignment quality using the gold standards provided by OAEI test cases. This illustrates the ability of the proposed approach to produce alignments of good accuracy. On the other hand a scalability evaluation is provided to show the applicability of the approach to large scale ontologies, where no gold standard is available.

5.1 Alignment Quality

In order to measure the quality of ontology alignments, the metrics *precision*, *recall*, and *F-measure* as known from information retrieval are applied. To this end, given an alignment A and a reference alignment R, the *precision* of A w.r.t. R is defined as

$$\mathrm{prec}(A,\, R) = \frac{|A \cap R|}{|A|} \tag{4}$$

denoting the fraction of found correspondences that are correct. The *recall* of A w.r.t. R is defined as

$$\mathrm{rec}(A,\, R) = \frac{|A \cap R|}{|R|} \tag{5}$$

denoting the fraction of expected correspondences that were found. The *F-measure* metric is the harmonic mean of precision and recall.

The OAEI provides several tracks where reference alignments are available that have been verified by human experts. The largest of those manually created reference alignments is in the *anatomy* track with 1522 correspondences.

[7] `http://oaei.ontologymatching.org/`

Table 1. Precision, recall, and F-measure scores for *MapPSO* in the OAEI benchmarks track (2008–2010), grouped for tests 1xx, 2xx, and 3xx, as well as the harmonic mean for all 111 tests [8,9]

	2008			2009			2010		
	prec	rec	f-meas	prec	rec	f-meas	prec	rec	f-meas
1xx	0.92	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00
2xx	0.48	0.53	0.50	0.75	0.73	0.74	0.67	0.59	0.63
3xx	0.49	0.25	0.33	0.54	0.29	0.38	0.72	0.39	0.51
h-mean	0.51	0.54	0.52	0.64	0.59	0.61	0.68	0.60	0.64

OAEI Benchmarks. The OAEI *benchmarks* track consists of 111 pairs of ontologies. One of the ontologies to be aligned is the same in all tests cases, while the other one is systematically altered by reducing features that are available for alignment systems [9, Sect. 3.1].

This section presents results for the prototypes *MapPSO* and *MapEVO* in the OAEI benchmarks track. The *MapPSO* system has participated in the track since 2008 and constantly increased its performance. Table 1 shows the precision, recall, and F-measure scores for each year.

Due to its novelty the *MapEVO* system has not officially participated in any previous OAEI campaign yet, however it has been tested using the latest 2010 benchmarks data set. The results of *MapEVO* compared to *MapPSO* in the 2010 OAEI benchmarks are shown in Table 2. The results were obtained using 16 species and 400 iterations for *MapEVO* and 60 particles and 250 iterations for *MapPSO*. These numbers were empirically determined such that further increasing them would not result in better scores.

As Table 2 illustrates, applying two different population-based optimisation heuristics results in roughly the same evaluation scores, with *MapEVO* performing slightly better, particularly for the 2xx tests. For both *MapPSO* and *MapEVO* the same objective function, *i.e.* base matcher configuration and aggregation was used, and both approaches score similar at the same tests. This shows that the alignment quality is largely influenced by the design of the objective function, while two different optimisation heuristics find the same optimum (according to this objective function) for each test.

Table 2. Precision, recall, and F-measure scores for *MapEVO*, *MapPSO* in the OAEI benchmarks track 2010, grouped for tests 1xx, 2xx, and 3xx, as well as the harmonic mean for all 111 tests. Results for *MapPSO* are taken from the official OAEI 2010 results.

	MapEVO			*MapPSO*		
	prec	rec	f-meas	prec	rec	f-meas
1xx	0.96	1.00	0.98	1.00	1.00	1.00
2xx	0.89	0.51	0.58	0.67	0.59	0.63
3xx	0.77	0.44	0.56	0.72	0.39	0.51
h-mean	0.89	0.53	0.66	0.68	0.60	0.64

It is worth mentioning that in the OAEI 2009, symmetric precision and recall scores [7] were computed in addition to the classical scores. The track organisers reported very good results of *MapPSO* in terms of these relaxed metrics, which are comparable to the best participating systems [8, Sect. 3.2]. The reason for this might be the fact that metaheuristics are typically used to find a solution, which is very close to the optimum, followed by a local search step that fine-tunes this solution. *MapPSO* does currently not apply such a local search step, which results in close-to-optimal results, which are honoured by the relaxed evaluation metrics, but penalised using the classical ones.

OAEI Directory. The OAEI *directory* track comprises a collection of 4639 single test cases that represent segments of large web directories [9, Sect. 6.1].

The *MapPSO* system participated in this track in 2010 and scored similar to the best performing system. In the history of this track since 2006, *MapPSO* achieved the fourth highest scores among all participating systems [9, Sect. 6.2]. The reference alignment for this track is not publicly available, so a comparative evaluation with the *MapEVO* prototype could not be conducted.

OAEI Anatomy. The OAEI *anatomy* track provides two ontologies from the biomedical domain. The reference alignment for these real-world ontologies has been manually created by domain experts and contains 61 % of trivial correspondences that do not require the use of a domain-specific thesaurus [9, Sect. 4.1].

The anatomy ontologies have been processed by the *MapEVO* prototype outside of the OAEI context. Using the same objective function as in the OAEI benchmarks with 4 species and 5000 iterations, *MapEVO* scored with a precision of 0.82, recall of 0.42, and F-measure of 0.56. The high precision score demonstrates the ability to approach a solution that is of high quality. Since the objective function used in this experiment did not make any use of an external biomedical thesaurus[8] only a portion (43 %) of the correspondences could be identified, which is close to the 61 % of trivial correspondences that can be found without external knowledge.

5.2 Scalability

In order to demonstrate the ability to process large ontologies, an alignment of the Gene Ontology (GO)[9] with ∼31,650 classes, and the Medical Subject Headings (MeSH)[10] with ∼15,340 classes was computed using the *MapPSO* prototype. The experiment has been executed using the Amazon Web Services™ (AWS) cloud infrastructure [3]. Since no reference alignment is available for these data sets, no thorough evaluation of the alignment quality could be provided. However the convergence of the algorithm could be measured in terms of the fitness

[8] Many participants utilise the UMLS thesaurus (`http://www.nlm.nih.gov/research/umls/`) for this purpose.

[9] `http://www.geneontology.org/`

[10] `http://www.nlm.nih.gov/mesh/meshhome.html`

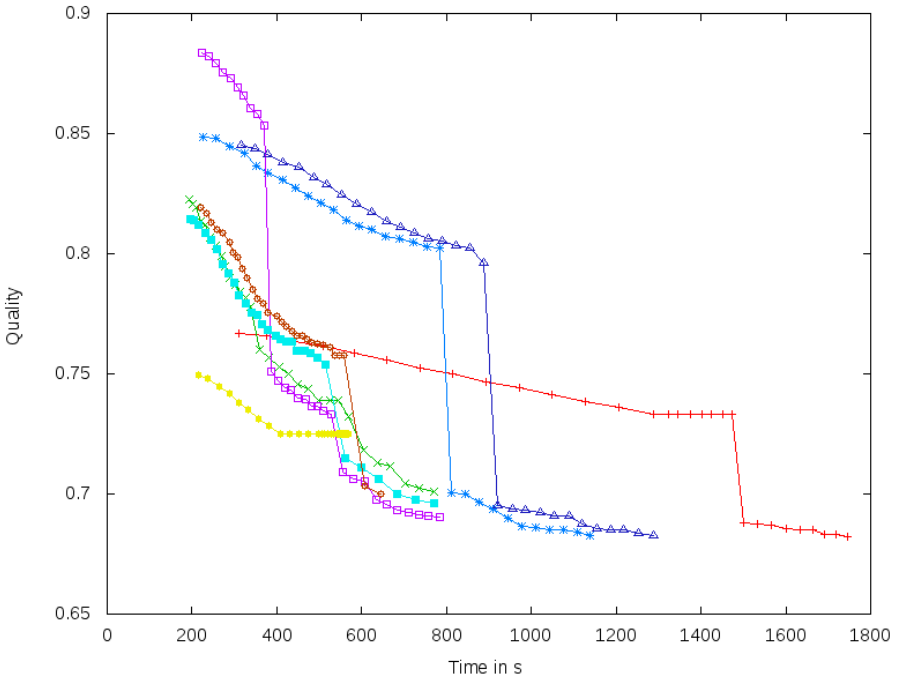


Fig. 5. Convergence behaviour regarding of the *MapPSO* algorithm (30 iterations) regarding alignment quality when aligning GO with MeSH. Each curve represents the quality of the alignment of a particular particle. Quality is denoted as a distance measure, thus lower values denote better quality.

scores of each particle throughout the iterations. Figure 5 shows the convergence behaviour of the *MapPSO* algorithm for this experiment. It can be observed that the quality of each alignment represented by a particle improves throughout the iterations. The case where a particle makes an adjustment towards the global best can be seen as a big step in a single iteration. In Fig. 6 the convergence towards the optimal alignment size is illustrated. Since an asynchronous update method was used [3] particles are progressing at a different pace, which is mostly depending on the alignment size of a particular particle.

5.3 Discussion

The fact that there are no reliable alignment gold standards available for large ontologies gave reason to split the evaluation in two: an *effectiveness* evaluation, and a *scalability* evaluation. The effectiveness evaluation takes well-established

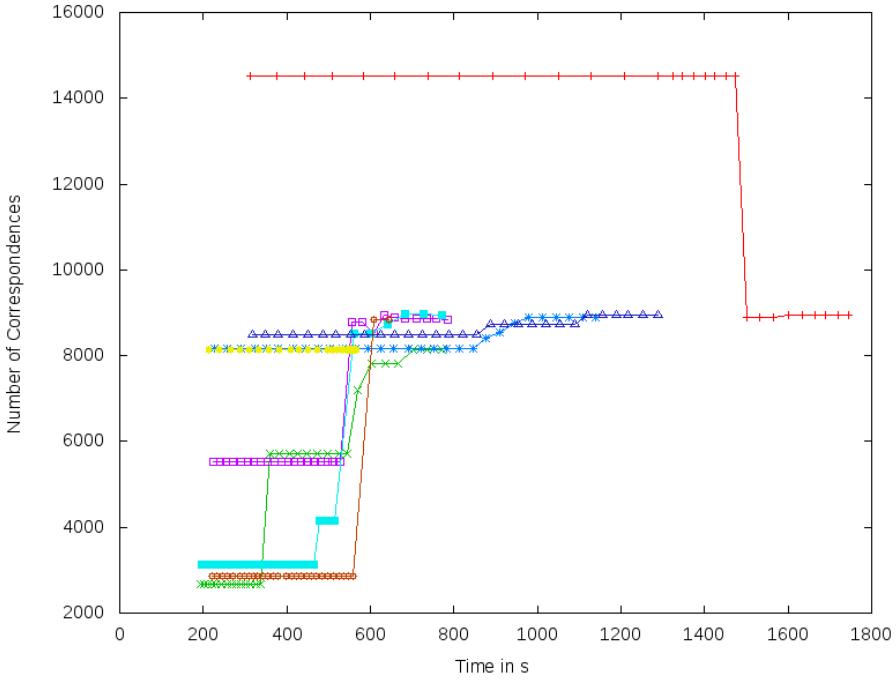


Fig. 6. Convergence behaviour of the *MapPSO* algorithm (30 iterations) regarding alignment size when aligning GO with MeSH. Each curve represents the size of an alignment of a particular particle.

OAEI data sets and demonstrates that the proposed systems *MapPSO* and *MapEVO* are capable of producing meaningful results. Comparing the systems with other OAEI participants shows, that other systems perform better in several test cases, particularly in the *benchmarks* data set[11]. The reason for this is, that the OAEI tests are small enough that an exhaustive, pairwise comparison of all entity pairs is feasible. In these cases there is no need to apply heuristic approaches, which are (without a local search component) inherently approximate. Experiences from OAEI participation in the years 2008–2010 indicate that the quality of results strongly depends on the specification of the objective function. Extending and tuning the objective function was the main reason for the increase in alignment quality from 2008 to 2010.

On the other hand, the scalability experiment demonstrates the potential of using population-based metaheuristics since they are capable of exploiting parallel computing infrastructures naturally. Consequently they are able to compute results for two large ontologies.

[11] Notably, the *CODI* system that implements a similar holistic alignment optimisation approach shows similar results in the *benchmarks* data set.

6 Related Work

Few systems have previously been developed that allow for optimisation based on a holistic assessment of an alignment. Niepert *et al.* [21] apply Markov logic in order to express hard and soft constraints that characterise desired alignments. Their approach is implemented in the *CODI* system. Given the ontologies and a set of these constraints, the most probable alignment is computed using *maximum a-posteriori inference* by *integer linear programming*. The authors claim that their approach is the first to be able to avoid incoherence in alignments during the actual alignment process. This is an advantage over other systems where incoherence is reduced after an alignment is generated, which usually requires several iterations in order to find the best alignment without incoherencies, as *e.g.* in the *ASMOV* system [12]. A disadvantage of the *CODI* system is that it requires the computation of a large matrix of basic similarities in order to provide valuable a-priori likelihoods for correspondences to be correct. Moreover, the calculation of the most probable alignment is expensive and thus does not scale well for large ontologies.

The same scalability problem has been observed when utilising the constraint-based Answer Set Programming paradigm for ontology merging [4]. In that approach the linguistic features of lightweight taxonomies are exploited using a WordNet oracle in order to bring two taxonomies in a new merged structure. The Answer Set Programming solver processes a programme that declaratively describes constraints an alignment has to fulfil, and generates several answer sets with each one representing a solution according to the constraints.

Global alignment evaluation has been partially included to other state-of-the-art alignment systems, such as *RiMOM* [17], in terms of variations of the similarity flooding algorithm by Melnik *et al.* [19]. A recent global alignment evaluation metric regarding the alignment's structural preservation properties has been introduced by Joslyn *et al.* [13], and implemented for example in the *AgreementMaker* system [6]. In spite of using the structural global alignment metric, the system does only apply it in a selection step after computing several similarity matrices using local correspondence evaluation metrics.

Coming from the other direction, there are examples of successful applications of population-based optimisation heuristics to problems that are similar to the ontology alignment problem. Correa *et al.* [5], for instance, apply discrete particle swarm optimisation to the problem of feature selection for a Bayesian classifier. The quality of the classifier can only be assessed when considering all selected features at once, making the isolated evaluation of single features infeasible.

7 Conclusion

This paper presented a novel approach to tackle the problem of ontology alignment by utilising population-based optimisation heuristics. This overcomes the limitation of traditional alignment systems, which compute alignments by considering similarity measures purely on the correspondence level. We consider

ontology alignment as a holistic optimisation problem that takes into account those local aspects (on the correspondence level), as well as global aspects (on the alignment level). The latter comprise inter-correspondence relationships, alignment size, coherency constraints, as well as other domain-specific global constraints that can be encoded in the objective function. Related approaches [21,4] have a scalability issue when solving the global constraint satisfaction problem.

We present two prototypes based on particle swarm optimisation (*MapPSO*) and evolutionary programming (*MapEVO*), resp., in order to demonstrate the feasibility of population-based optimisation heuristics in ontology alignment. Evaluation results show that the prototypes are able to find good quality alignments on benchmark data sets. The two optimisation approaches *MapPSO* and *MapEVO* compute similar alignments (regarding quality) using the same objective function. This shows that approaching the gold standard strongly depends on the objective function that is used, and thus leaves room for improvement regarding its details w.r.t. the benchmark data set. More generally this means that a wisely chosen objective function (w.r.t. a particular alignment scenario) can achieve good results using the proposed algorithms. In a different experiment, a straight convergence of the algorithm can be observed when aligning large ontologies. Despite the lack of a reference alignment for this scenario, the evaluation results allow the conclusion that with a wisely chosen objective function, population-based optimisation can be used to align large ontologies.

Future work on this topic will be directed towards the utilisation of ant colony optimisation algorithms, as well as the exploration of possibilities to adjust the objective function in order to maximise precision or recall depending on the alignment scenario. Additionally, means for user interrogation during the alignment process could be integrated to dynamically fine-tune the objective function. In the case of particle swarm optimisation, it has been shown [11] that users can also directly be involved in "guiding" the swarm to promising areas in the search space. Another extension of the proposed algorithms is to carry out a final local search step after the metaheuristic has found a suitable near-optimal solution. This will fine-tune the results while still avoiding exhaustive searches.

Acknowledgement. The presented research was partially funded by the German Federal Ministry of Economics (BMWi) under the project "Theseus" (number 01MQ07019).

References

1. Bellahsene, Z., Bonifati, A., Duchateau, F., Velegrakis, Y.: On Evaluating Schema Matching and Mapping. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping. Springer, Heidelberg (2011)
2. Bock, J., Hettenhausen, J.: Discrete Particle Swarm Optimisation for Ontology Alignment. Information Sciences (2010)
3. Bock, J., Lenk, A., Dänschel, C.: Ontology Alignment in the Cloud. In: Proc. of the 5th Int. Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 689, pp. 73–84 (2010), `http://ceur-ws.org`

4. Bock, J., Topor, R., Volz, R.: Ontology Merging using Answer Set Programming and Linguistic Knowledge. In: Proc. of the 2nd Int. Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 304, pp. 301–305 (2007), http://ceur-ws.org
5. Correa, E.S., Freitas, A.A., Johnson, C.G.: A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set. In: Proc. of the 8th Genetic and Evolutionary Computation Conf. ACM, New York (2006)
6. Cruz, I.F., Antonelli, F.P., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N., Rosenthal, A. (eds.) Proceedings of the 4th International Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 551, pp. 49–60 (2009), http://ceur-ws.org
7. Ehrig, M., Euzenat, J.: Relaxed Precision and Recall for Ontology Matching. In: Proc. of the K-CAP Workshop on Integrating Ontologies. CEUR Workshop Proceedings, vol. 156, pp. 25–32 (2005), http://ceur-ws.org
8. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Trojahn, C., Vouros, G., Wang, S.: Results of the Ontology Alignment Evaluation Initiative. In: Proc. of the 4th Int. Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 551, pp. 73–126 (2009), http://ceur-ws.org
9. Euzenat, J., Ferrara, A., Meilicke, C., Nikolov, A., Pane, J., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Trojahn dos Santos, C.: Results of the Ontology Alignment Evaluation Initiative. In: Proc. of the 5th Int. Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 689, pp. 85–117 (2010), http://ceur-ws.org
10. Fogel, M.J., Owens, L.J., Walsh, A.J.: Artificial Intelligence through Simulated Evolution. Wiley, Chichester (1966)
11. Hettenhausen, J.: Interactive Multi-Objective Particle Swarm Optimisation with Heatmap Visualisation based User Interface. Master's thesis, Griffith University (2007)
12. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology Matching with Semantic Verification. Web Semantics 7(3), 235–251 (2009)
13. Joslyn, C.A., Paulson, P., White, A.: Measuring the Structural Preservation of Semantic Hierarchy Alignments. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N., Rosenthal, A. (eds.) Proceedings of the 4th International Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 551, pp. 61–72 (2009), http://ceur-ws.org
14. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (April 2001)
15. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. of IEEE Int. Conf. on Neural Networks, vol. 4. IEEE Computer Society (1995)
16. Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)
17. Li, J., Tang, J., Li, Y., Luo, Q.: RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. IEEE Transactions on Knowledge and Data Engineering 21(8), 1218–1232 (2009)
18. Meilicke, C., Stuckenschmidt, H.: Repairing Ontology Mappings. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence. AAAI Press (2007)

19. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: Georgakopoulos, D., Agrawal, R., Dittrich, K. (eds.) Proceedings of the 18th International Conference on Data Engineering, pp. 117–128. IEEE Computer Society, Washington, DC, USA (2002)
20. Mutter, M.: Ontology Alignment durch Evolutionäre Algorithmen. Diplomarbeit, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (2011)
21. Niepert, M., Meilicke, C., Stuckenschmidt, H.: A Probabilistic-Logical Framework for Ontology Matching. In: Proc. of the 24th AAAI Conf. on Artificial Intelligence. AAAI Press (2010)
22. Rahm, E.: Towards Large-Scale Schema and Ontology Matching. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping. Springer, Heidelberg (2011)
23. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: Proc. of IEEE Int. Conf. on Evolutionary Computation. IEEE Computer Society (1998)
24. Whitley, D.: An overview of evolutionary algorithms: practical issues and common pitfalls. Information and Software Technology 43(14), 817–831 (2001)