

Improving Ontology Alignment through Memetic Algorithms

Giovanni Acampora*, Pasquale Avella[†], Vincenzo Loia*, Saverio Salerno[‡] and Autilia Vitiello*

*Dept. of Computer Science

University of Salerno, Fisciano, Italy 84084

Email: {gacampora, loia, avitiello}@unisa.it

[†]Dept. of Engineering

University of Sannio, Benevento, Italy 82100

Email: avella@unisannio.it

[‡]Dept. of Electronic Engineering and Information Engineering

University of Salerno, Fisciano, Italy 84084

Email: salerno@unisa.it

Abstract—Born primarily as means to model knowledge, ontologies have successfully been exploited to enable knowledge exchange among people, organizations and software agents. However, because of strong subjectivity of ontology modeling, a matching process is necessary in order to lead ontologies into mutual agreement and obtain the relative alignment, i.e., the set of correspondences among them. The aim of this paper is to propose a memetic algorithm to perform an automatic matching process capable of computing a suboptimal alignment between two ontologies. To achieve this aim, the ontology alignment problem has been formulated as a minimum optimization problem characterized by an objective function depending on a fuzzy similarity. As shown in the performed experiments, the memetic approach results more suitable for ontology alignment problem than other evolutionary techniques such as genetic algorithms.

Index Terms—Memetic Algorithms, Ontology Alignment, Optimization.

I. INTRODUCTION

Nowadays, ontologies are recognized to be the most appropriate means to support information sharing and reuse in all modern knowledge based systems. They have been successfully exploited in many research areas such as artificial intelligence (AI), information retrieval (IR), database theory, and most recently, World Wide Web (in particular for Semantic Web), e-Commerce and e-Learning. According to [1], an ontology is an explicit specification of a conceptualization, i.e., the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them[2]. Unfortunately, the variety of ways that a domain can be conceptualized (by considering different concepts, constraints and properties) results in the creation of different ontologies with contradicting or overlapping parts[3]. Therefore, a so called *mapping process* is necessary in order to reconcile heterogeneous ontologies and allow system interoperability. The recent researches about ontologies are heavily focused in the development of several automatic reconciliation processes to overcome this heterogeneity. One of the principal techniques is the so-called *matching process* which consists in detecting correspondences among existing ontologies in order to make

them consistent. Different from other mapping methods such as *integration* and *merging* which return a single ontology, the matching process keeps separate the original ontologies and generates in output an *alignment*, i.e., the set of correspondences among them. In order to perform a matching process and to find an appropriate alignment between two ontologies, a lot of techniques[4][5][6] based on different theories have been exploited. The main aim of this paper is to propose an innovative method to perform an automatic matching process based on the so-called *memetic algorithms*, i.e., optimization hybrid methods which combine evolutionary algorithms with local search techniques [7]. In order to solve the ontology alignment problem through a memetic approach, a formulation of the alignment problem as a minimization problem has been provided by defining an appropriate objective function based on a fuzzy similarity measure able to quantify the matching level between ontologies entities.

In the experimental results section, our proposal is compared with genetic algorithms in order to show, in the ontology alignment problem context, the more suitability and efficiency of the memetic approach than other evolutionary techniques. The evaluation of the algorithms is executed through a statistical analysis, and precisely, by means of the well-known Wilcoxon's signed rank test[8]. The used benchmarks are provided by the Ontology Alignment Evaluation Initiative (OAEI) 2009 [9].

II. THE ONTOLOGY ALIGNMENT PROBLEM

In this section, a set of formal definitions concerning to matching process are given. According to [10], a matching process consists of generating an alignment (A') from a pair of ontologies (o and o') given in input. So, before formally defining the matching process, the conventional definitions of ontology and alignment are provided.

Definition 1 (Ontology): An ontology O is a triple

$$O = (C, P, I)$$

where C is the set of *classes*, P the set of *properties* and I the set of *individuals*.

Definition 2 (Alignment): An alignment is a set of so-called *mapping elements*. A mapping element is a 5-uple: $\langle id, e, e', n, R \rangle$ where

- id is a unique identifier of the given mapping element;
- e and e' are the entities (e.g., tables, XML elements, properties, classes) of the first and the second ontology respectively;
- n is a confidence measure in some mathematical structure (typically in the $[0,1]$ range) holding for the correspondence between the entities e and e' ;
- R is a relation (e.g., equivalence ($=$); more general ($\supseteq$); disjointness ($\perp$); overlapping ($\cap$)) holding between the entities e and e' .

In order to compute a mapping element, a matching process uses a *similarity measure* which determines the closeness between the entities e and e' in the range $[0, 1]$ where 0 stands for complete inequality and 1 for complete equality. A lot of kinds of similarity measures exist in literature. In particular, there are techniques which rely on vector representations of ontologies and use distances measures between these vectors and other approaches which consider an ontology as a set of entities and so depend on the techniques used for establishing the distance among ontology entities. A comparison between similarity distances is provided in [11].

In order to perform a matching process two input ontologies o and o' are needed, but also other optional inputs can be specified: a partial alignment A which is to be completed by the process to obtain A' ; some parameters such as weights and thresholds and some external resources such as lexicons and databases. Therefore, the matching or alignment process can be formally defined as follows:

Definition 3 (Matching process): The matching process can be seen as a function ϕ which, from a pair of ontologies o and o' to align, an input alignment A , a set of parameters p , a set oracles and resources r , returns a new alignment A' between these ontologies:

$$A' = \phi(o, o', A, p, r).$$

In order to evaluate the quality of the output alignment A' provided by a matching process, typically conformance measures derived from the information retrieval field [12] are exploited. These methods require the employment of a *reference alignment* R . The most commonly used conformance measures are *precision*, *recall* and *f-measure* which represents a combination of the first ones. Formally:

Definition 4 (Precision): Given a reference alignment R , the precision of an alignment A is given by

$$Pr(A, R) = \frac{|R \cap A|}{|A|}.$$

Definition 5 (Recall): Given a reference alignment R , the recall of an alignment A is given by

$$Rec(A, R) = \frac{|R \cap A|}{|R|}.$$

Definition 6 (F-measure): Given a reference alignment R and a number α between 0 and 1, the F-measure of an alignment A is given by

$$M_\alpha = \frac{Pr(A, R) \cdot Rec(A, R)}{(1 - \alpha) \cdot Pr(A, R) + \alpha \cdot Rec(A, R)}.$$

If $\alpha = 1$, then the F-measure is equal to precision and if $\alpha = 0$, the F-measure is equal to recall. In between, the higher α , the more importance is given to precision with regard to recall. Very often, the value $\alpha = 0.5$ is used, i.e.,

$$M = \frac{2 \times Pr(A, R) \times Rec(A, R)}{Pr(A, R) + Rec(A, R)},$$

the harmonic mean of precision and recall.

In order to perform a matching process and to find an appropriate alignment between two ontologies, a lot of techniques based on different theories have been exploited. All methods can be classified as *instance-based* and *schema-based*: in order to perform a matching process, the former use all entities of an ontology including the individuals, the latter use only entities as classes and properties. A furthermore classification based on the kinds of inputs is possible among schema-based methods obtaining the following three categories: *terminological*, *structure* and *semantics* methods. A complete discussion can be found in [13]. Only recently, also evolutive methods such as machine learning and genetic algorithms have been exploited to solve the alignment problem[14][15][16].

Taking a cue from these new researches, our work proposes to detect correspondences between two ontologies by means of a memetic algorithm which performs a schema-based matching process taking in input two ontologies (leaving out the other input parameters) and considering a string distance based similarity measure.

III. ONTOLOGY ALIGNMENT AS OPTIMIZATION PROBLEM

In this section, a formal definition of the ontology alignment problem is given by reformulating it as a minimization problem based an objective function that uses a fuzzy similarity in order measure the correspondence between ontology entities.

Beside the general scheme presented in the section II, the alignment process can be opportunely restricted in order to deliver a particular kind of alignment useful for a particular kind of applications. In particular, in our approach, the ontology alignment problem is depending upon the following features:

- a simplified definition of mapping element, renamed *correspondence* (see definition 7), which implies the equality as the relation R ;
- cardinality $n : 1$ (many-to-one), i.e., an entity of the first ontology can be associated with an only entity of the second one, whereas, an entity of the second ontology can be associated also with more entities of the first one.

Definition 7 (Correspondence): For two ontologies O_1 and O_2 , a correspondence is defined as:

$$c = (e_i, e_j)$$

where e_i is the i^{th} entity of ontology O_1 whereas e_j is the j^{th} entity of ontology O_2 and the implied relation is the equality with a confidence measure in $[0, 1]$.

Basing on these features, the formal definition of the ontology alignment problem taken into account in this work can be provided.

Definition 8 (Alignment): An alignment A between two ontology O_1 and O_2 is the set:

$$A = \{c_k = (e_i, e_j) | e_i \in O_1 \text{ with } i = 0, \dots, h_1 - 1, \\ e_j \in O_2 \text{ with } j \in \{0, \dots, h_2 - 1\} \text{ and} \\ k = 1, 2, \dots, h_1\}$$

where $h_1 = |O_1|$ and $h_2 = |O_2|$.

In order to formulate the considered alignment problem as optimization problem, it is necessary to introduce the concept of the *evaluation of a correspondence*. The evaluation is performed by a method whose aim is to determine the goodness of a correspondence to achieve an optimal alignment. In general, the evaluation method is a function $f : A \rightarrow \mathbb{R}$ which associates a real value to each correspondence in A . In our work, a *fuzzy similarity* between the name labels of the entities composing a correspondence is chosen to realize the evaluation method. Precisely, the goodness of a correspondence is evaluated through a fuzzy relation between entity names computed by means of the well-known *levenshtein string distance* [17]. Formally, by considering s_1 and s_2 the name labels of the entities involved in the correspondence $c \in A$, the exploited evaluation method is given by the formula:

$$f(c) = \mu(s_1, s_2)$$

where $\mu : S \times S \rightarrow [0, 1]$ (S is the set of all entities names) is a membership function defined as follows:

$$\mu(s_1, s_2) = \frac{lev(s_1, s_2)}{N}$$

where

- $lev(s_1, s_2)$ is the levenshtein distance between the two strings s_1 and s_2 , i.e, the minimum number of edits needed to transform the string s_1 into s_2 , with the allowable edit operations being insertion, deletion, or substitution of a single character;
- $N = \max(N_1, N_2)$ with N_1 and N_2 , respectively, the lengths of the strings s_1 and s_2 .

Finally, the alignment problem can be formulated as an optimization problem.

Definition 9 (Alignment problem): The alignment problem can be defined as the tuple (O_1, O_2, A_{set}, F) where:

- O_1 and O_2 are the ontologies to align;
- A_{set} is the set of all possible alignments between O_1 and O_2 ;
- $F : A \rightarrow \mathbb{R}$ is the objective function defined in the following way:

$$F(A) = \sum_{i=1}^{|A|} f(c_i) \text{ with } c_i \in A \quad (1)$$

In other words, it is possible to calculate the evaluation value of a specific alignment A by considering the sum of fuzzy evaluations of its correspondences. Since the correspondence evaluation function f is a distance measure between the correspondence entity names, the lower are $F(A)$ values and the alignment A is nearer to optimal one, so our aim is to minimize $F(A)$ values. Therefore, the considered alignment problem is a minimum optimization problem.

IV. OUR PROPOSAL FOR THE ALIGNMENT PROBLEM: MEMETIC ALGORITHMS

This paper aims of presenting a novel technique to solve the ontology alignment problem by exploiting an emergent hybrid method named Memetic Algorithms (MAs). The MAs are population-based search methods which combine evolutionary algorithms with local search (LS) methods. The name is inspired by Richard Dawkins concept of a *meme*, which represents a unit of cultural evolution that can exhibit local refinement [18]. Indeed, the MAs have the capability of realizing local search processes within the successive generations characterizing global search methods in order to refine population individuals.

In this section, a memetic algorithm which extends a genetic algorithm with the Stochastic Hill Climbing search is presented. Initially, the two components (the genetic algorithm and the Hill Climbing search) to be combined to realize our proposal are separately described. Then, their merging to perform an automatic matching process is discussed.

A. The First Component: a Genetic Algorithm

Our proposal is to realize a memetic algorithm as extension of a precise category of evolutionary methods, the genetic algorithms. For this reason, in this section the fundamental evolutive concepts and the genetic algorithm which will be extended to solve the considered alignment problem are given.

The interest in genetic algorithms began as early as the 1970s when Holland [19] first proposed them as search methods inspired by the mechanisms of evolutionary processes in nature. In particular, genetic algorithms are based on Darwinian principle of the natural selection which leads to the survival of the only fittest individuals. More in detail, in nature, evolution manifests itself as a succession of changes in species' features determined by genetic reproduction. The individual features, represented by chromosomes (formed in turn by genes), determine the survival capacity of individuals. Indeed, only fittest individuals capable of adapting to the changing environment survive and reproduce. Therefore, the evolution process is derived from the joint action of natural selection and the recombination of genetic material that occurs during reproduction and that generates diversity in the *gene pool* [20].

Genetic algorithms try to solve an optimization (or search) problem by manipulating a *population* of potential solutions and by reproducing the natural evolution process. Specifically, they operate on encoded representations of the solutions, called *chromosomes*, that equivalent to the representations of individual features in nature. The algorithm evolution starts from

a population of randomly generated individuals and consists in successive *generations*. In each generation, as in nature, a *selection process* provides the mechanism for selecting better solutions to survive. Each solution is evaluated by means a *fitness function* that reflects how good it is, compared with other solutions in the population. The higher is the fitness value of an individual and higher are its chances of surviving. Recombination of genetic material in genetic algorithms is simulated through two operators: *crossover* that exchanges portions between two randomly selected chromosomes and *mutation* that causes random alteration of the chromosome genes. The algorithm evolution terminates when specified conditions such as the maximum number of generations are reached. The general genetic algorithm structure is shown in the figure IV-A.

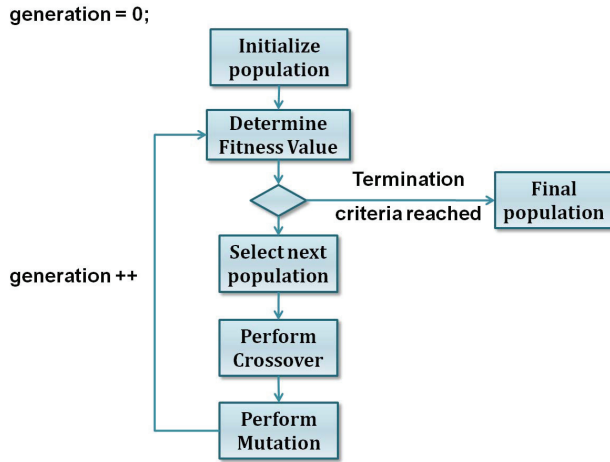


Fig. 1. The typical steps of a genetic algorithm.

In order to customize the building of the starting genetic algorithm for the considered alignment problem, it is necessary to define the chromosome structure in order to represent the solution of our problem, i.e., an alignment, and a fitness function which allows the evaluation of the considered solutions.

In particular, with regard to fitness function, simply it is to minimize the described function F (see eq.1), whereas, the proposed chromosome structure requires a more detailed description.

By remembering the definitions 7 and 8 and considering that each ontology entity is enumerated from 0 to $n + m + k - 1$ where m, n, k are, respectively, the cardinalities of the sets C, P, I which compose an ontology, a chromosome representing an alignment A is a integer vector where each cell represents a correspondence between an entity of the first ontology and any entity of the second one. More in detail, by considering that the i^{th} vector cell contains the integer j , the cell represents the correspondence (e_i, e_j) . The vector length is the same of the cardinality of alignment A , i.e., the number of the entities of the first ontology. Formally:

Definition 10 (Alignment Chromosome): A chromosome S representing an alignment A between two ontology O_1 and O_2

is the set:

$$S = \{(e_0, e_{j_0}), (e_1, e_{j_1}), (e_2, e_{j_2}), \dots, (e_h, e_{j_h})\}$$

where $h = |O_1| - 1$ and $j_l \in \{0, 1, 2, \dots, |O_2| - 1\}$ with $l = 0, 1, 2, \dots, h$.

Therefore, the chromosome dimension depends on alignment problem instance.

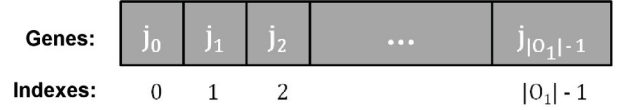


Fig. 2. The general structure of an alignment chromosome: each gene represents the correspondence (e_i, e_{j_i}) where $i = 0, 1, 2, \dots, |O_1| - 1$ and $j_i \in \{0, 1, 2, \dots, |O_2| - 1\}$

In order to complete the initial genetic algorithm, the description of termination criteria has to be given. In details, our proposal is to consider a combined termination criterion which stops the genetic algorithm if the maximum number of iterations or the alignment f-measure value (definition 6) is achieved.

Once defined the alignment chromosome, the fitness function and the termination criteria, the genetic algorithm (shown in the table I) could be run in order to solve the described alignment problem.

B. The Second Component: the Hill Climbing Search

The local search strategies such as steepest-descent, conjugate-gradient and linear approximation methods perform iterative search for optimum solution in the *neighborhood* of a candidate. Starting from a single candidate solution, these methods use *local* information to decide which neighbor they should move to next [21]. In each iteration, typically, a mutation operator is applied in order to improve current candidate. The process ends when there are not changes which improve current solution or after a specific number of iterations.

In particular, the Hill Climbing algorithm is a local search iterative method which starts with an arbitrary solution to the problem at issue. During iterations, the algorithm attempts to find a better solution by incrementally changing a single component of the current one. If the change improves the current solution, then the new one becomes the current one. The search is repeated until no further improvements can be found or after a maximum number of iterations. There are several variants of Hill Climbing search depending on how the next solution is tried. For example, the *Simple Hill Climbing algorithm* chooses the first closer node to solution, whereas the *Steepest Ascent Hill Climbing search* compares all successors and then selects the closest.

Our proposal is to use the stochastic version of Hill Climbing search which represents a trade-off between the previous mentioned variants in terms of efficiency and efficacy. The Stochastic Hill Climbing algorithm, presented in the table II, selects a neighbor at random, and then decides whether to move to that neighbor or to examine another by considering the amount of improvement.

TABLE I
LISTING OF THE GENETIC ALGORITHM.

Input: entity name string sets S_1 and S_2 of two ontologies O_1 and O_2 to align; the number of the entities n_1 and n_2 of the ontologies; parameters of GA (size of the population pop_size , crossover rate p_c , mutation rate p_m), termination criteria (maximum number of iterations max_iter and desired f-measure value $f_{measure}$).

Output: the best optimized alignment (represented by the final best chromosome) between the ontologies O_1 and O_2 .

```

1:  $gen \leftarrow 0$ ;
2:  $pop \leftarrow \text{generatePopulation}(pop\_size)$ ; // Generate randomly an initial population  $pop$  of  $pop\_size$  chromosomes
3:  $\text{evaluateFitness}(pop)$ ; // Evaluate fitness value for each chromosome
4:  $best\_chromosome \leftarrow \text{getBestChromosome}(pop)$ ; // Select the best chromosome of the current population
5:  $quality \leftarrow \text{evaluateAlignment}(best\_chromosome)$ ; //  $quality$  contains the f-measure value for the current best chromosome
6: while ( $gen \leq max\_iter$  AND  $quality \leq f_{measure}$ ) do
7:    $\text{executeCrossover}(pop, p_c)$ ; // Crossover chromosomes according to a crossover rate  $p_c$  to add new chromosomes to population  $pop$ 
8:    $\text{executeMutation}(pop, p_m)$ ; // Mutate chromosomes with a mutation probability  $p_m$  to add new chromosomes to population  $pop$ 
9:    $\text{evaluateFitness}(pop)$ ; // Evaluate fitness value for new chromosomes
10:   $\text{executeSelection}(pop, pop\_size)$ ; // Select  $pop\_size$  chromosomes to generate next new population  $pop$ 
11:   $best\_chromosome \leftarrow \text{getBestChromosome}(pop)$ ; // Select the best chromosome of the current population
12:   $\text{preserveBestChromosome}(pop, best\_chromosome)$ ; // Preserve the best chromosome of the current population: Elitist Strategy
13:   $gen \leftarrow gen + 1$ ; // Increment number of iterations
14:   $quality \leftarrow \text{evaluateAlignment}(best\_chromosome)$ ; //  $quality$  contains the f-measure value for the current best chromosome
15: end while
16: return  $best\_chromosome$ ;

```

TABLE II
LISTING OF THE STOCHASTIC HILL CLIMBING ALGORITHM.

Input: an alignment chromosome $align_solution$ representing the initial solution, the maximum number of iterations max_iter .

Output: the alignment chromosome sol which represents the solution optimized by means of local search.

```

1:  $sol \leftarrow align\_solution$ ;
2:  $iter \leftarrow 0$ ;
3: while ( $iter < max\_iter$ ) do
4:    $new\_sol \leftarrow \text{getRandomNeighbor}(sol)$ ;
5:   if ( $\text{evaluate}(new\_sol) < \text{evaluate}(sol)$ ) then
6:      $sol \leftarrow new\_sol$ ;
7:      $iter \leftarrow 0$ ;
8:   end if
9: end while
10: return  $sol$ ;

```

C. Our Memetic Algorithm to Solve the Alignment Problem

In spite of their known efficiency, all local search approaches suffer from the same drawback: they tend to get stuck in local optima. Instead, thanks to their capability of exploring and exploiting promising regions of search space, the genetic algorithms have the strong benefit not to be prone to stalling at local optima. However, they can take to relatively long time to locate the exact local optimum in a region of convergence (and may sometimes not to find the optimum with sufficient precision) [21]. For this reason, the idea to create an hybrid method that combines the genetic algorithms with the local search methods, i.e., the MAs.

The MAs, representing the marriage between global search and local improvement, have the complementary advantages of EAs (generality, robustness and global search efficiency) and local search methods (rapid convergence toward local minima). In the figure IV-C the typical structure of MAs

which, in literature, are known also as hybrid EAs (e.g., [22]), genetic local searchers (e.g.,), Lamarckian GAs (e.g., [23]), Baldwinian GAs (e.g., [24]), etc., is presented.

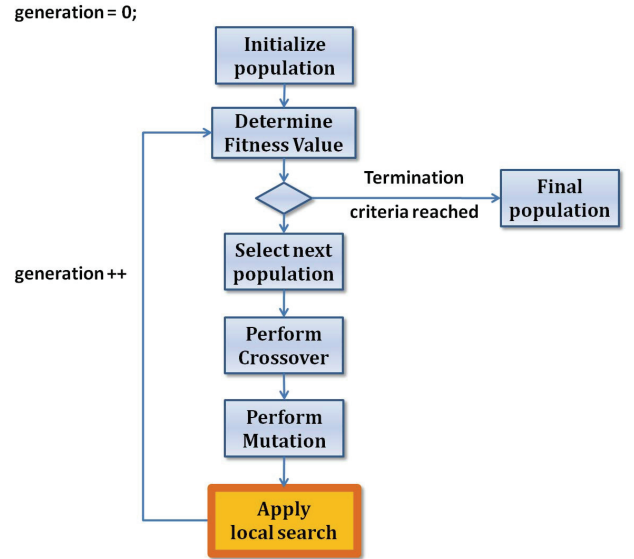


Fig. 3. The general structure of a memetic algorithm

In order to design a *competent* [25] MA a lot of issues must be addressed over the two principal components. Perhaps the foremost of these issues may be stated as [26]: *What is the best tradeoff between local search and the global search provided by evolution?*. This issue leads naturally to questions such as the following:

- *Local search frequency*: How often should local search be applied within the evolutionary cycle?
- *Order respect to genetic operators*: When should local search be applied?

- *Individual selection mechanism*: Which individuals in the population should be improved by local search?
- *Local search intensity*: How much computational effort should be allocated to each local search?

Therefore, in order to complete the design of a memetic algorithm for ontology alignment problem, the aforementioned issues must be considered. Our proposal is to build a memetic algorithm which has the following features:

- *Local search frequency* = 1, i.e., the local search is applied within each evolutionary cycle;
- *Order respect to genetic operators* = *after*, i.e., local refinement is executed after crossover and mutation operators;
- *Individual selection mechanism* = *only the best*, i.e., only the best chromosome of population is improved by the local search process;
- *Local search intensity* = *equal to n local search iterations*, i.e., each local search process ends after running n iterations.

Finally, the whole proposed memetic algorithm is presented in the table III.

V. EXPERIMENTAL RESULTS

In this section, an experimentation is led in order to investigate the performances of the proposed memetic algorithm and show the suitability of memetic approach in the ontology alignment context. To achieve this aim, the proposed algorithm is compared with the genetic algorithm shown in the table I.

In our experimentation, the matching process involves ontologies characterizing the well-known benchmarks provided by the OAEI 2009. Table IV shows a brief description about the features of the used benchmarks.

TABLE IV
BENCHMARKS DESCRIPTIONS

Identifier	Description
101	Strictly identical ontologies
103	A regular ontology and other with a language generalization
104	A regular ontology and other with a language restriction
204	Ontologies with different naming conventions
221	A regular ontology and other with no specialisation
224	Identical ontologies without instances
225	Identical ontologies without restrictions

The evaluation of the algorithms is realized by means of a non-parametric statistical procedure for performing pairwise comparisons named Wilcoxon's signed rank test.

A. Characteristics of Experimentation

In this section, a description of the features of the performed experiments is reported.

In particular, the experiments are carried out as follows:

- The memetic and genetic algorithm are run 20 times for each selected benchmark;

- Each run stops when the computed alignment is characterized by a f-measure value equal to 0.8 or after executing 100000 iterations;
- The averages of number of fitness evaluations for each benchmark and for both algorithms are computed and shown in the table V.

The used parameters are the following ones:

- The GA parameters for both algorithms are:
 - population size = 20 chromosomes;
 - crossover rate = 0.8;
 - mutation rate = 0.02.
- The local search parameters are:
 - maximum number of iterations = 50.

Finally, the platform characteristics where the algorithms were run are:

- Processor: Intel Core 2 Duo;
- CPU Speed: 1.83 GHz;
- RAM Capacity: 2GB.

TABLE V
SAMPLES FOR THE WILCOXON'S TEST: THE SIGLE *AE* STANDS FOR average of the number of fitness evaluations over the executed 20 runs

Identifier	Memetic <i>AE</i>	Genetic <i>AE</i>
101	12.120,70	36.287,45
103	11.524,00	33.735,70
104	12.234,85	35.798,70
204	28.558,50	97.602,90
221	11.652,15	34.569,15
224	13.202,50	35.168,30
225	11.532,40	35.167,05

B. Experimentation: Wilcoxon's test

As described in [27], the Wilcoxon's test is a non-parametric procedure employed in a hypothesis testing situation involving a design with two samples. It is a pairwise test used for answering this question: do two samples represent two different populations? Therefore, it can be employed to detect significant differences between the behavior of two algorithms and so it is suitable for our experimentation: showing that the memetic approach is better than genetic one for solving ontology alignment problem.

In order to compute the Wilcoxon's test, two samples related to the two compared algorithms have to be chosen. In our experimentation, the performance of algorithms depends on the executed number of fitness evaluations n_E : the smaller the value n_E and better is the algorithm at issue. For this reason, the two considered samples are represented by the data shown in the table V, i.e., the obtained averages of the number of fitness evaluations related to the two compared algorithms.

In general, a hypothesis testing as the Wilcoxon's test is a procedure in which sample data are employed to evaluate a hypothesis. In details, in order to evaluate the research hypothesis, i.e., the statement of what a researcher predicts,

TABLE III
LISTING OF THE MEMETIC ALGORITHM.

Input: entity name string sets S_1 and S_2 of two ontologies O_1 and O_2 to align; the number of the entities n_1 and n_2 of the ontologies; GA parameters (size of the population pop_size , crossover rate p_c , mutation rate p_m), termination criteria (maximum number of iterations max_iter and desired f-measure value $f_{measure}$); local search parameters (maximum number of iterations max_local_iter).

Output: the best optimized alignment (represented by the final best chromosome) between the ontologies O_1 and O_2 .

```

1:  $gen \leftarrow 0$ ;
2:  $pop \leftarrow \text{generatePopulation}(pop\_size)$ ; // Generate randomly an initial population  $pop$  of  $pop\_size$  chromosomes
3:  $\text{evaluateFitness}(pop)$ ; // Evaluate fitness value for each chromosome
4:  $best\_chromosome \leftarrow \text{getBestChromosome}(pop)$ ; // Select the best chromosome of the current population
5:  $quality \leftarrow \text{evaluateAlignment}(best\_chromosome)$ ; //  $quality$  contains the f-measure value for the current best chromosome
6: while ( $gen \leq max\_iter$  AND  $quality \leq f_{measure}$ ) do
7:    $\text{executeCrossover}(pop, p_c)$ ; // Crossover chromosomes according to a crossover rate  $p_c$  to add new chromosomes to population  $pop$ 
8:    $\text{executeMutation}(pop, p_m)$ ; // Mutate chromosomes with a mutation probability  $p_m$  to add new chromosomes to population  $pop$ 
9:    $\text{evaluateFitness}(pop)$ ; // Evaluate fitness value for new chromosomes
10:   $\text{executeSelection}(pop, pop\_size)$ ; // Select  $pop\_size$  chromosomes to generate next new population  $pop$ 
11:   $best\_chromosome \leftarrow \text{getBestChromosome}(pop)$ ; // Select the best chromosome of the current population
12:   $\text{preserveBestChromosome}(pop, best\_chromosome)$ ; // Preserve the best chromosome of the current population: Elitist Strategy
13:   $local\_chr \leftarrow \text{executeHillClimbingSearch}(best\_chromosome, max\_local\_iter)$ ; // Execute the local search process on the best chromosome
14:   $\text{replaceWorstChromosome}(pop, local\_chr)$ ; // Select the worst chromosome and replace it with that obtained by the local search process
15:   $gen \leftarrow gen + 1$ ; // Increment number of iterations
16:   $quality \leftarrow \text{evaluateAlignment}(local\_chr)$ ; //  $quality$  contains the f-measure value for the current best chromosome
17: end while
18: return  $best\_chromosome$ ;

```

two statistical hypotheses are needed: the so called *null hypothesis* (H_0) and the so called *alternative hypothesis* (H_1). In our experimentation, the null hypothesis states no difference between algorithms, i.e. $H_0 : \theta_D = 0$, indicating that in the underlying populations represented by the two samples related to compared algorithms, the median of the difference scores equals zero. On the other side, a directional alternative hypothesis $H_1 : \theta_D < 0$ is chosen to indicate a difference between the memetic algorithm and the genetic one and specifically the superiority of the first one on the second one. Since the statement of our research hypothesis predicts the presence of a difference between algorithms which are being studied, and in particular, the superiority of memetic one, as result of our experimentation, the null hypothesis is expected to be rejected in favor of the alternative one.

The Wilcoxon's test is usually included in well-known statistical software packages such as SPSS, SAS and R. In particular, our experiment is performed by means of R tool by using the *wilcox* command in this way:

`wilcox.test(x, y, paired='TRUE', alternative='less')`

where x and y are two vectors which contain the averages of numbers of fitness evaluations, respectively, of the memetic and genetic algorithms.

However, before running the Wilcoxon's test, the used level of significance α for the performed experiment has to be decided. In particular, for our experimentation, a level of significance $\alpha = 0.05$ was chosen.

In the figure V-B, the R output for the executed Wilcoxon's test is shown. The computed p -value is 0.007813. Since the output of experimentation results in a p -value $< \alpha$, our experiment rejects the null hypothesis and affirms the superiority of memetic method over genetic approach in the context of the

ontology alignment problem.

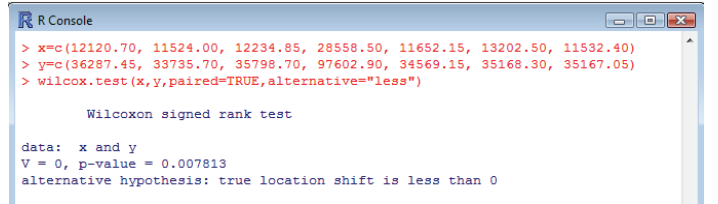


Fig. 4. The result of Wilcoxon's test

VI. CONCLUSIONS

The main aim of this paper is to apply the memetic approach to solve the ontology alignment problem. Memetic algorithms are an emergent optimization technique which combines evolutionary and local search methods. Therefore, in order to exploit a memetic algorithm capable of realizing a so-called matching process between two ontologies, it has been necessary to formulate the ontology alignment problem as a minimum optimization problem and to describe an appropriate objective function defined by means of a fuzzy similarity between the ontology entities. In order to prove the more suitability of the memetic proposal than other evolutionary techniques, an experimentation has been led by comparing the proposed memetic algorithm with the extended genetic one. The performed Wilcoxon's signed rank test shows the higher performances provided by the designed memetic approach.

REFERENCES

- [1] Gruber, T. R., "A Translational Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol.5, pp. 199–220, 1993.
- [2] M. R. Genesereth and N. J. Nilsson, "Logical Foundations of Artificial Intelligence," San Mateo, CA: Morgan Kaufmann Publishers, 1987.

- [3] G. Stoilos, G. Stamou and S. Kollias, "A String Metric for Ontology Alignment," in *Proceedings of the ninth IEEE International Symposium on Wearable Computers*, pp.624-637, 2005.
- [4] H. H. Do and E. Rahm, "COMA - a system for flexible combination of schema matching approaches," in *Proceedings of the Very Large Data Bases Conference (VLDB)*, pp. 610621, 2001.
- [5] J. Madhavan, P. Bernstein, and E. Rahm. "Generic schema matching with Cupid," in *Proceedings of the Very Large Data Bases Conference (VLDB)*, pp. 4958, 2001.
- [6] N. Noy and M. Musen. "Anchor-PROMPT: using non-local context for semantic matching," in *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6370, 2001.
- [7] P. Moscato. "On evolution, search, optimization, GAs and martial arts: toward memetic algorithms," California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.
- [8] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol.1, pp.8083, 1945.
- [9] Ontology Alignment Evaluation Initiative (OAEI). <http://oaei.ontologymatching.org/2009>.
- [10] P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tessaris. D2.2.1: Specification of a common framework for characterizing alignment. Technical report, NoE Knowledge Web project deliverable, 2004. <http://knowledgeweb.semanticweb.org/>.
- [11] Jrme David and Jrme Euzenat. Comparison between ontology distances (preliminary results). In Proc. 7th conference on international semantic web conference (ISWC), Karlsruhe (DE), pages 245260, 2008.
- [12] C. J. Van Rijsbergen, *Information Retrieval. 2nd edition*, London, Butterworths, 1979.
- [13] Shvaiko, P., Euzenat, J., "A survey of schema-based matching approaches," *Journal on Data Semantics IV*, pp.146-171, 2005.
- [14] J. Wang, Z. Ding, C. Jiang, "GAOM: Genetic Algorithm based Ontology Matching," in *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, 2006.
- [15] Martinez-Gil, J.; Alba, E.; and Montes, J. F. A. "Optimizing ontology alignments by using genetic algorithms," In Guret, C.; Hitzler, P.; and Schlobach, S., eds., *Nature inspired Reasoning for the Semantic Web (NatuReS2008)*, volume 419 of *CEUR Workshop Proceedings*, 2008.
- [16] Anhai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy, "Ontology Matching: A Machine Learning Approach," in *Handbook on Ontologies in Information Systems*, pp. 397-416, 2003.
- [17] Levenshtein VI, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol.10, pp. 707710, 1966.
- [18] R. Dawkins, *The Selfish Gene*. New York: Oxford Univ. Press, 1976.
- [19] J.H. Holland, "Adaptation in Natural and Artificial Systems," Univ. of Michigan Press, Ann Arbor, Mich., 1975.
- [20] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *IEEE Comput.*, vol. 27, no. 6, pp 17-26, June 1994.
- [21] Y. S. Ong, Q. H. Nguyen, M. H. Lim and T. Jing, "A Development Platform for Memetic Algorithm Design," *Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on Advanced Intelligent Systems*, O-okayama Campus West Bldg. 9, Tokyo Institute of Technology, Japan, 20-24 September , 2006.
- [22] L. He and N. Mort, "Hybrid genetic algorithms for telecommunications network back-up routing," *BT Technol. J.*, vol. 18, no. 4, pp. 4256, 2000.
- [23] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson, "Automated docking using a lamarkian genetic algorithm and an empirical binding free energy function," *J Comput. Chem.*, vol. 14, pp. 16391662, 1998.
- [24] K. Ku and M. Mak, "Empirical analysis of the factors that affect the Baldwin effect," in *Lecture Notes in Computer Science*, 1998, Proc. Parallel Problem Solving From NaturePPSN-V, pp. 481490.
- [25] D. Goldberg, *The Design of Innovation: Lessons From and for Competent Genetic Algorithms*. Norwell, MA: Kluwer, 2002.
- [26] N. Krasnogor and J. Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," in *IEEE Transactions On Evolutionary Computation*, vol. 9, no. 5, October 2005.
- [27] S. Garca, D. Molina, M. Lozano, F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 Special Session on Real Parameter Optimization," *J Heuristics*, vol.15, pp.617644, 2009.
- [28] J.H. Zar, *Biostatistical Analysis*. Prentice Hall. 1999.