2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Comparison of Multi-Objective Evolutionary Algorithms for the Ontology Meta-Matching Problem

Giovanni Acampora, Hisao Ishibuchi, Autilia Vitiello

Abstract— **In recent years, several ontology-based systems have been developed for data integration purposes. The principal task of these systems is to accomplish an ontology alignment process capable of matching two ontologies used for modeling heterogeneous data sources. Unfortunately, in order to perform an efficient ontology alignment, it is necessary to address a nested issue known as *ontology meta-matching problem* consisting in appropriately setting some regulating parameters. Over years, evolutionary algorithms are appeared to be the most suitable methodology to address this problem. However, almost all of existing approaches work with a single function to be optimized even though a possible solution for the ontology meta-matching problem can be viewed as a compromise among different objectives. Therefore, approaches based on multi-objective optimization are emerging as techniques more efficient than conventional evolutionary algorithms in solving the meta-matching problem. The aim of this paper is to perform a systematic comparison among well-known multi-objective Evolutionary Algorithms (EAs) in order to study their effects in solving the meta-matching problem. As shown through computational experiments, among the compared multi-objective EAs, OMOPSO statistically provides the best performance in terms of the well-known measures such as hypervolume, Δ index and coverage of two sets.**

I. Introduction

DATA integration is concerned with unifying data that share some common semantics but originate from distributed and unrelated sources [1]. When one works on data integration, it is necessary to face the heterogeneity characterizing data provided by different sources. In the literature, multiple types of heterogeneity are well known and classified in four categories [1]: (1) structural heterogeneity, involving different data models; (2) syntactical heterogeneity, involving different languages and data representations; (3) systemic heterogeneity, involving hardware and operating systems; and (4) semantics heterogeneity, involving different concepts and their interpretations. Since the 1990s, the emergence of distributed computing and middleware technology has particularly supported the overcoming of the syntactic and structural heterogeneities, and allowed us to focus on issues at the information level [2]. Therefore, currently, one of the most important and complex problems within data integration is the semantic heterogeneity [1].

Over the years, ontologies are emerging as the most useful means to address this issue thanks to their capability of formally describing the semantics of a particular domain of interest. Several ontology-based systems for data integration can be found in the literature [3][1]. Among the main architectures that are implemented in ontology-based data integration systems, the most popular one consists in taking into consideration multiple ontologies, each modeling an individual data source, and detecting a set of mappings among them for achieving integration. This process, named *ontology alignment*, requires to set some parameters in order to correctly perform its task. Since these parameters strongly affect the quality of the produced alignments, the selection of their most appropriate values represents a nested issue known as *meta-matching problem*.

Since the beginning, heuristic approaches, and, in particular, evolutionary algorithms, are appearing as the most suitable methodology to address the meta-matching problem [4]. In detail, the evolutionary-based meta-matching systems [5][6] typically use a single objective to evaluate the alignment quality during the generation process, even though a suitable computation of parameters could be better performed by evaluating the right compromise among different objectives involved in the meta-matching process. As a consequence, approaches based on multi-objective optimization algorithms are emerging as an innovative and efficient methodology to face the meta-matching problem [7][8].

However, no existing works perform a formal performance evaluation to establish the efficiency of different multi-objective Evolutionary Algorithms (EAs) in handling the meta-matching problem. Consequently, in this paper, a systematic comparison among well-known multi-objective EAs used to tune the ontology matching parameters is carried out in terms of well-known performance indices such as hypervolume, Δ index and coverage of two sets. The compared EAs are an improved version of the Non-dominated Sorting Genetic Algorithm [9] (NSGA-II), an improved version of the Strength Pareto Evolutionary Algorithm [10] (SPEA2), an improved version of the Pareto Envelope based Selection Algorithm [11] (PESA-II), the Optimal Multi-Objective Particle Swarm Optimization [12] (OMOPSO), the Multi-Objective Evolutionary Algorithm based on Decomposition [13] (MOEA/D) and the Duplicate Elimination Non-dominated Sorting Evolutionary Algorithm [14] (DENSEA). The experiments involve a dataset provided by OAEI commonly used for experimentation on the ontology alignment problem. The comparison is performed through a statistical multiple comparison procedure.

Giovanni Acampora is with the School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, UK email: giovanni.acampora@ntu.ac.uk.

Hisao Ishibuchi is with the Department of Computer Science, Graduate School of Engineering, Osaka Prefecture University, Japan email: hisaoi@cs.osakafu-u.ac.jp.

Autilia Vitiello is with the Department of Computer Science, University of Salerno,84084 Fisciano, Salerno, Italy e-mail: avitiello@unisa.it.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

II. The Ontology Meta-Matching Problem

According to the most common definition [15], an *ontology* is an explicit specification of a conceptualization, i.e., the formal specification of concepts and objects within a given domain application, and the relationships existing between them. All components of an ontology (concepts, objects and relationships) are generally denoted with the term *entity*. Given a pair of ontologies O_1 and O_2, an ontology alignment process is a function f which returns an alignment A, i.e., a set of correspondences, each one of them relates two entities of the ontologies under alignment which are considered similar semantically.

Traditionally, ontology alignment processes produce an alignment by computing a so-called *confidence value* η for all possible pairs of entities belonging to the ontologies to be aligned. In detail, the confidence value η (typically in the $[0, 1]$ range) represents the similarity level existing between the two entities composing the correspondence. In order to compute the confidence value for each correspondence, the common technique is to perform a *similarity aggregation* procedure which computes a real value for each correspondence by combining different similarity measures belonging to various categories: lexical, linguistic and structural. In detail, lexical measures compute a string distance-based similarity between two entities by taking into account the morphology of the words which characterize them (such as names, comments, etc.); a linguistic one determines a similarity value between two entities by taking into account semantic relations such as synonymy and hypernymy; structural ones compute a similarity value between two entities by considering their kinship (parents and children). In this work, we consider the following similarity measures extracted by [16]: *Entity Name Distance Measure* and *Comment Distance Measure* among lexical similarity measures, *Hierarchy Distance Measure* and *Domain and Range Restrictions Distance Measure* belonging to structural ones and the linguistic *Word Net Synonymy Name Distance Measure*. Since the application of a single measure is often not enough to produce an acceptable output alignment, the common technique is to combine different matchers to compute a single aggregated similarity value. Formally, let us consider an alignment A, a correspondence c belonging to the alignment A and h similarity measures, the confidence value η for c can be defined as follows:

$$\eta(c) = \sum_{i=1}^{h} w_i \times sim_i(c) \qquad \text{subject to } \sum_{i=1}^{h} w_i = 1 \tag{1}$$

where w_i is the weight associated to the i^{th} similarity measure and $sim_i(c)$ is the similarity value computed by the i^{th} similarity measure. Only correspondences with a confidence value greater than a given threshold value $t \in [0, 1]$ are considered valid and can be inserted in the output alignment A (*filter operation*).

From this description of an ontology alignment process, it is evident that the quality of the produced alignments is strongly dependent on the specification of similarity measures, weights and thresholds used as matching parameters. Therefore, this specification represents a crucial issue in the ontology alignment scenario known as the nested *ontology meta-matching problem* [4].

The aim of this work is to compare a set of well-known multi-objective EAs to optimize the combination of the weights w_i (with $i = 1, \ldots, h$) and the threshold value for the ontology meta-matching problem.

III. Multi-objective Evolutionary Algorithms for the Ontology Meta-Matching Problem

This paper aims at addressing the meta-matching problem as a multi-objective optimization problem and comparing different existing algorithms for solving it. In detail, in our vision, the meta-matching problem is solved as an optimization one where (1) the outcome solutions represent the most appropriate specifications of the set of matching parameters and (2) the search is guided by the exploitation of two objectives evaluating the alignment quality during the generation process. As fully described later, the two considered objectives take into account the number of correspondences composing the alignment and the average of similarities characterizing them. Since it is often complex to weight these objectives “a priori”, in this paper, we investigate the effects of different “a posteriori” multi-objective EAs.

In general, a multi-objective optimization problem can be described as a vector function f that maps a tuple of m parameters (decision variables) to a tuple of n objectives. Formally:

$$\min \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))$$

$$\text{subject to } \mathbf{x} = (x_1, x_2, \ldots, x_m) \in X$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_n) \in Y$$

where $\mathbf{x}$ is called the *decision vector*, X is the *parameter space*, $\mathbf{y}$ is the *objective vector* and Y is the *objective space*. The set of solutions of a multi-objective optimization problem consists of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degrading in another - these vectors are known as *Pareto optimal*. Mathematically, the concept of Pareto optimality is as follows. Let us consider a minimization problem and two decision vectors $\mathbf{a}, \mathbf{b} \in X$. Then, $\mathbf{a}$ is said to dominate $\mathbf{b}$ (also written as $\mathbf{a} \prec \mathbf{b}$) iff

$$\forall i \in \{1, 2, \ldots, n\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \wedge$$

$$\exists j \in \{1, 2, \ldots, n\} : f_j(\mathbf{a}) < f_j(\mathbf{b})$$

Additionally, $\mathbf{a}$ is said weakly dominate $\mathbf{b}$ (also written as $\mathbf{a} \preceq \mathbf{b}$) iff $\mathbf{a} \prec \mathbf{b}$ or $f(\mathbf{a}) = f(\mathbf{b})$. All decision vectors which are not dominated by any other decision vector of a given set are called *non-dominated* regarding this set. The decision vectors that are non-dominated within the entire search space are referred to Pareto optimal and constitute the so-called Pareto-optimal set or *Pareto-optimal front*.

By analyzing the general structure of a multi-objective problem, it is clear that in order to formulate the meta-matching problem as a multi-objective one, it is necessary to define the decision vector and the used objectives. Hereafter, these points will be deepened before describing the considered multi-objective EAs.

A. Decision vector

Each decision vector contains the values for the set of weights, indicating the contribution of each similarity measure, and the threshold value t, used to decide if a pair of entities must belong to alignment or not. Since all decision values (weights and threshold) are real numbers in the unit interval $[0, 1]$, the decision vector is defined as a real vector in the unit hypercube. When we have h similarity measures, the decision vector has h weights and a single threshold value. Thus, it is coded as a real number string of length $h + 1$. However, since the sum of all weights has to be equal to 1, a further step is needed in order to obtain the weight values, which can be used in the similarity aggregation step. In detail, each one of the first h values of the decision vector is scaled conforming to the sum of all the first h values. Formally:

$$w_i = \frac{g_i}{\sum_{i=1}^{h} g_i} \tag{2}$$

where g_i is the i^{th} value of the decision vector and w_i is the i^{th} weight used to perform the similarity aggregation task.

B. Objectives

The objectives evaluate the quality of the alignment corresponding to a decision vector. In detail, starting from a decision vector σ, it is possible to create an alignment by following two steps:

- the computation of the confidence value η (as shown in equation 1) for each pair of entities between the two ontologies to be aligned by considering the weights w_i, with $i = 1, 2, \dots, h$, composing the decision vector σ after the scaling operation;
- the selection of pairs of entities with a confidence value greater than or equal to the threshold value t contained in the decision vector σ.

Then, the evaluation of the created alignment is computed by considering these two observations:

- a lower average of the dissimilarity of the pair of entities corresponds to a better alignment;
- by considering the same average of the dissimilarity, a lower difference between the number of correspondences composing the evaluated alignment and the number of correspondences composing the optimal alignment Ω reflects a better alignment.

These two observations lead to the definition of two objectives, y_1 and y_2, formally defined as follows. Let us consider a decision vector σ with the corresponding alignment A, the first objective y_1 is defined below:

$$y_1 = \frac{\sum_{i=1}^{|A|} \eta(c_i)}{|A|} \tag{3}$$

where $\eta(c_i)$ is the confidence value of the i^{th} correspondence belonging to the alignment A and $|A|$ is the number of correspondences composing the alignment under evaluation.

The second objective y_2 is defined as follows:

$$y_2 = \Phi(abs(|A^{optimal}| - |A|)) \tag{4}$$

where $|A^{optimal}|$ represents the number of correspondences composing the optimal alignment $A^{optimal}$, $|A|$ is the number of correspondences composing the alignment under evaluation, abs is a function which computes the absolute value and Φ is a function of normalization in range $[0, 1]$. In particular, the function Φ is introduced to allow the evaluation of the objective y_2 in a range which does not change according to the ontologies under alignment and the size of the corresponding $A^{optimal}$. In detail, let us consider $diff = abs(|A^{optimal}| - |A|)$, the function Φ follows the following formula:

$$\Phi(diff) = \frac{diff}{diffMax}$$

where $diffMax$ represents the max number of correspondences for which $A^{optimal}$ and A may differ.

In order to conclude, it is worth noting that the value $|A^{optimal}|$ is not known, and, hence, it is approximated as follows:

$$|A^{optimal}| = min(|O_1|, |O_2|)$$

where $|O_1|$ and $|O_2|$ represent, respectively, the cardinality of ontologies O_1 and O_2 and min is a function which computes the minimum between two values.

C. Six existing multi-objective evolutionary approaches

In this section, we present the compared multi-objective EAs in our study. We consider six state-of-the-art algorithms:

1) NSGA-II: it is an improved version of NSGA proposed by Deb et al. [17]. The principal features of this algorithm are the exploitation of elitism and the crowded tournament selection. In detail, crowded tournament selection is a selection mechanism based on tournament selection whereby, a group of individuals takes part in a tournament and the winner is judged by the fitness levels that each individual brings to the tournament. In NSGA-II, the most fit individuals are determined by a ranking mechanism (or crowded comparison operator) composed of two parts. The first part ‘peels’ away layers of non-dominated fronts, and ranks solutions in earlier fronts as better. The second one computes a dispersion measure, the *crowding distance*, to determine how close a solution’s nearest neighbors are, with larger distances being better. At each generation, the best solutions with regard to these two measures are saved as the next population, and genetic operators are applied to form a new child population.

2) SPEA2: it is an enhancement version of that originally proposed in [18]. It is characterized by storing non-dominated solutions externally in a second, continuously updated population named archive. At each generation, it determines the most fit individuals within the union of archive and child populations by computing a fitness value composed of two parts. The first part is a raw fitness value based on how many solutions it dominates, and the second is a density estimate based on its proximity to other solutions in the objective space. The computed most fit solutions are saved as the next population, and genetic operators are applied to form a new child population. In particular, as in NSGA-II, it uses the binary tournament as the selection operator.

3) PESA-II: It uses different mechanisms than SPEA2 and NSGA-II. In detail, the main differences are that its archive population is not of fixed size and only allows non-dominated solutions to be members. If the archive ever exceeds the number of solutions in a population, a *squeeze factor* is calculated for all members of the archive and used to reduce the size of the archive (the solutions with the highest squeeze factor are removed). Genetic operators are then applied to archive members to form a new child population.

4) OMOPSO: it uses an adaptive mutation operator and an adaptive inertia weight to improve the searching capacity. Besides, a new crowding operator based on level sorting is used to improve the distribution of non-dominated solutions along the Pareto front and maintain the population diversity. The new crowding distance does not need to sort order for every objective and, as a consequence, it has less complexity than that used by NSGA-II. Finally, the ϵ-dominance is used to keep the size of the external archive of the non-dominated solutions.

5) MOEA/D: it differs from the other EAs because it is based on a decomposition approach. In detail, it decomposes the multi-objective problem in a set of scalar subproblems and solves them simultaneously. At each generation, the population is composed of the best solution found so far for each subproblem. The neighborhood relations among these subproblems are defined based on the distances between their aggregation coefficient vectors. The optimal solutions of neighboring subproblems should be very similar. Each subproblem is optimized by using information only from its neighboring subproblems. This approach aims to facilitate the diversity maintenance. In detail, the decomposition is performed by using the Tchebycheff approach in this paper.

6) DENSEA: it is based on the non-domination sorting criterion selection and has incorporated some elitism, but it is characterized by offering population diversity maintenance based on deletion and replacement of duplicate solution. In detail, the implemented deletion operator for duplicate solutions works as follows: the algorithm deletes the accumulated duplicate solutions due to the reduced non-dominated solutions quantity in the objective space and replaces each deleted solution by inserting the individual that has the same ordering in the second half of the population until the completion of 50% of the population size (N/2). In that way, the inclusion of diverse solutions replacing duplicates is fostered, helping to maintain the population diversity.

IV. Experiments and Results

This section is devoted to study the effects of the six considered multi-objective evolutionary algorithms (NSGA-II, SPEA2, PESA-II, OMOPSO, MOEA/D and DENSEA) on solving the ontology meta-matching problem. In the performed experiments, each compared algorithm ends after 250 evaluations of fitness and runs by using the following parameters: population size equals to 50 and crossover and mutation rates equal to, respectively, 0.8 an 0.01. As for the genetic operators, all multi-objective EAs use the Simulated Binary Crossover (SBX) and the Polynomial Mutation. Regarding selection, each multi-objective EA uses its own selection operator (see section III-C).

Hereafter, more details about the performed comparison are given after a description of the exploited dataset and performance metrics.

A. Dataset

In all our experiments, we have exploited a well-known dataset, named benchmark track[1], provided by OAEI and commonly used for experiments on ontology alignment problems. In detail, the dataset deals with the topic of scientific publications and it is composed of a set of 50 test cases organized in five groups (see Table I). Each test case represents a specific alignment task devoted to align a reference ontology with its artificially built variation, except for the last four test cases which are aimed at matching the reference ontology to four real ontologies.

TABLE I
Benchmark track description

Test case Id Range	# Test cases	Description
101-104	3	The ontologies under alignment are the same or the first one is the "OWL Lite restriction" of the second one.
201-210	10	The ontologies under alignment have the same structure, but different lexical and linguistic features.
221-247	18	The ontologies under alignment have the same lexical and linguistic features, but different structure.
248-266	15	The ontologies under alignment have different lexical, linguistic and structure features.
301-304	4	The ontologies under alignment are real world cases.

In particular, in our comparison, we have randomly chosen 25% of test cases in each group. This is a reasonable choice because the test cases belonging to the same group are characterized by the same features and complexity. The selected test cases are: 103, 201, 209, 230, 238, 240, 247, 251, 254, 257, 265 and 301.

B. Performance metrics

The quality of an obtained solution set for a multi-objective optimization problem can be evaluated by considering the following three aspects: (1) the distance of the resulting non-dominated front to the Pareto-optimal front should be minimized; (2) a good (in most cases uniform)

[1] http://oaei.ontologymatching.org/2011/

distribution of the solutions is desirable; (3) the spread of the obtained non-dominated front should be maximized, i.e., for each objective a wide range of values should be covered by the non-dominated solutions [19]. Since these three tasks cannot be measured adequately with one performance metric, over years, many performance metrics have been introduced for evaluating the quality of the non-dominated fronts for multi-objective optimization problems. In particular, in [20], all existing Performance Indices (PIs) are categorized in three groups: cardinality-based PIs, accuracy PIs and distribution and spread PIs. In our comparison, we evaluate the considered algorithms by using a metric for each one of the defined groups:

- Hypervolume [18] (accuracy PI): this metric takes into consideration the size of the dominated volume in the objective space. In the two-dimensional (2-D) case, this metric is mathematically described as follows:
$$H = \{\sum_i S_i | x_i \in P\}$$
where P is the non-dominated solution set under evaluation and S_i is the area dominated by the solution x_i. The areas S_i are computed with respect to a reference point in the objective space which typically is assumed to be composed of the maximum value for each objective. A greater value of H indicates both a better convergence to as well as a good coverage of the evaluated front [21].
- Δ index [9] (distribution and spread PI): this metric is based on distance and includes information about both spread and distribution. The Δ index is computed by the following formula:
$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|P|-1} |d_i - \overline{d}|}{d_f + d_l + (|P|-1) \cdot \overline{d}}$$
where P is the front to be evaluated, d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of P, $\overline{d}$ is the average of all distances d_i, $i \in [1, |P|-1]$, representing the Euclidean distance between consecutive solutions. It is worth noting that the denominator is the value of the numerator for the case when all solutions lie on one solution. In addition, it is interesting to note that this is not the worst case spread of solutions possible. Indeed, we can have a scenario in which there is a large variance in d_i, obtaining a value for the metric greater than one. A good distribution, instead, would make all distances d_i equal to $\overline{d}$ and would make $d_f = d_l = 0$ (with existence of extreme solutions in the non-dominated set). Thus, for the most widely and uniformly spreadout set of nondominated solutions, the numerator would be zero, making the metric to take a value zero [9].
- Coverage of two sets [18] (binary cardinality-based PI): this metric is a binary one because it is computed by considering two fronts to be compared one against the other. It is also referred as C metric and it is computed by the following formula:
$$C(A,B) = \frac{|\{y \in B : \exists x \in A \text{ t.c. } x \preceq y\}|}{|B|}$$
where A and B are the two fronts to be compared and $\preceq$ represents the weak dominance relation. The function $C(\cdot,\cdot)$ maps the ordered pair (A,B) to the [0,1] interval. In detail, the value $C(A,B)$ equal to 1 means that all the solutions in B are dominated by the front A. As opposite, $C(A,B)$ equal to 0 represents the situation where none of the solutions in B is dominated by the front A. It is worth mentioning that $C(A,B)$ does not have to be equal to $1 - C(B,A)$. Thus, both $C(A,B)$ and $C(B,A)$ should be computed for evaluation. In particular, we consider that A outperforms B if $C(A,B) > C(B,A)$.

To represent the spread and the shape of the distribution for the values obtained by the comparison of the proposed algorithms in terms of the aforementioned performance indices, we exploit a graphical method, called *box plot*. In detail, the upper and lower ends of the box are the upper and lower quartiles, while a thick line within the box encodes the median. Dashed appendages summarize the spread and shape of the distribution, and crosses represent outside values.

C. Comparison among the considered multi-objective EAs

This section is devoted to compare the considered multi-objective EAs to identify the best performer for the ontology meta-matching problem. The experiments consist of running each algorithm 30 times on each one of the considered test cases and storing the resulting non-dominated set as the outcome of each optimization run. The comparison is carried out by using the aforementioned metrics of performance: hypervolume, Δ index and coverage of two sets (C).

Figs. 1, 2, 3 show, respectively, the hypervolume, the Δ index and C values for each algorithm and for each test case by using the aforementioned boxplot diagrams. By analyzing them, OMPSO seems to provide the best performance among all compared multi-objective algorithms. Indeed, the median of the hypervolume values is for each test case greater than the corresponding medians of the other three EAs except for the test case 103 where OMPSO has the same performance of NSGA-II, SPEA2 and PESA-II. In addition, on ten of the twelve test cases, OMOPSO outperforms the other EAs in terms of the median of the Δ index values. Finally, OMOPSO covers 100% of the fronts computed by DENSEA in all test cases, and, on eleven of the twelve test cases, OMOPSO covers more than 88% of the fronts computed by MOEA/D, more than 80% of the fronts computed by SPEA2 and NSGA-II and more than 70% of the fronts computed by PESA-II. In contrast, DENSEA covers 0% of the OMOPSO outcomes in all test cases, and, on eleven of the twelve test cases, MOEA/D covers less than 5%, SPEA2 covers less than 10%, NSGA-II covers less than 20%, PESA-II covers less than 25% of the OMOPSO outcomes.

In order to examine the existence of statistical significance in the performance difference among the considered algo-

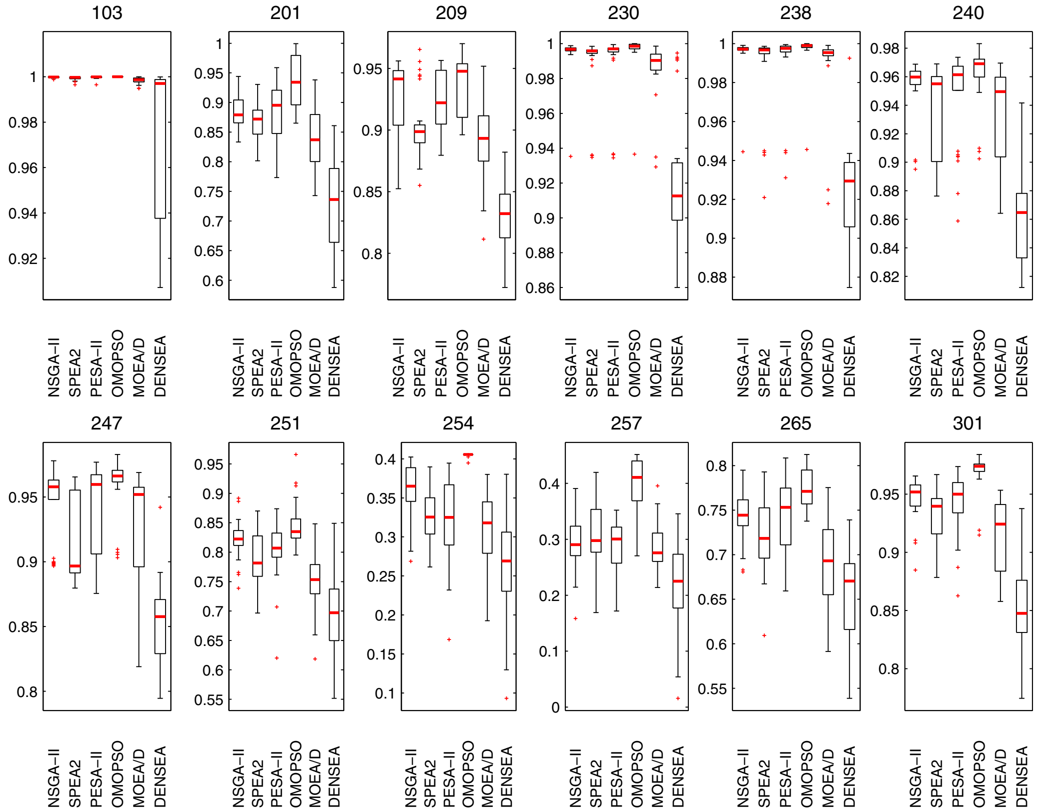

Fig. 1. Box plots based on the hypervolume metric. Each rectangle contains six box plots representing the distribution of the hypervolume values for each algorithm for a specific test case.

rithms, we use a statistical multiple comparison procedure for each metric. In detail, the considered statistical multiple comparison procedure is composed of two steps: in the first one, a statistical technique such as the Friedman's test is used to determine whether the results provided by the considered algorithms present any inequality; in the second one, which is performed only if in the first step an inequality is found, a post-hoc test such as Holm's test is led in order to determined which algorithm better outperforms.

In the case of the hypervolume and Δ index metrics, the samples used to perform Friedman's test are composed of the medians of the values for each test case. Instead, as for the C metric, the sample of each algorithm is composed of the integer values representing the number of the algorithms that it outperforms in each test case. Table II shows the ranking obtained by all compared EAs during the Friedman's tests performed for hypervolume, Δ index and C metrics. The computed Friedman's statistics are, respectively, 53.810, 44.1 and 52.048. Since they are greater than the critical value for five degrees of freedom $\chi^2_{0,1} = 9.2364$ (to be considered being six the number of compared algorithms), the null hypothesis is rejected for each metric and it is possible to assess that there is a significant difference between at least two of the compared algorithms as for all considered metrics.

Attending to this result, a post-hoc statistical analysis is needed to conduct pairwise comparisons to detect concrete differences among compared algorithms. Holm's procedure is a multiple comparison method that works by setting a control algorithm and comparing it with the remaining ones. Normally, the algorithm which obtains the lowest value of ranking in the Friedman's test is chosen as control algorithm. In our experimentation, as shown in Table II, OMOPSO is characterized by the lowest value of ranking for each metric, and, hence, it represents the control algorithm for the Holm's test for all considered metrics. All data computed by the Holm's procedure for each metric are displayed, respectively, in Tables III, IV and V. By analyzing them, Holm's procedure rejects all hypothesis for each metric. As a consequence, it is possible to state that OMOPSO statistically outperforms the other considered EAs at 90% confidence level (α is set to 0,1) for each one of the considered metrics.

TABLE II

FRIEDMAN'S TEST RANKING FOR ALL CONSIDERED METRICS.

algorithm	hypervolume	Δ index	C
NSGA-II	2,708	3,083	2,667
SPEA2	3,792	2,5	4
PESA-II	2,458	3,75	2,75
OMOPSO	**1,125**	**1,167**	**1**
MOEA-D	4,917	5,083	4,583
DENSEA	6	5,417	6

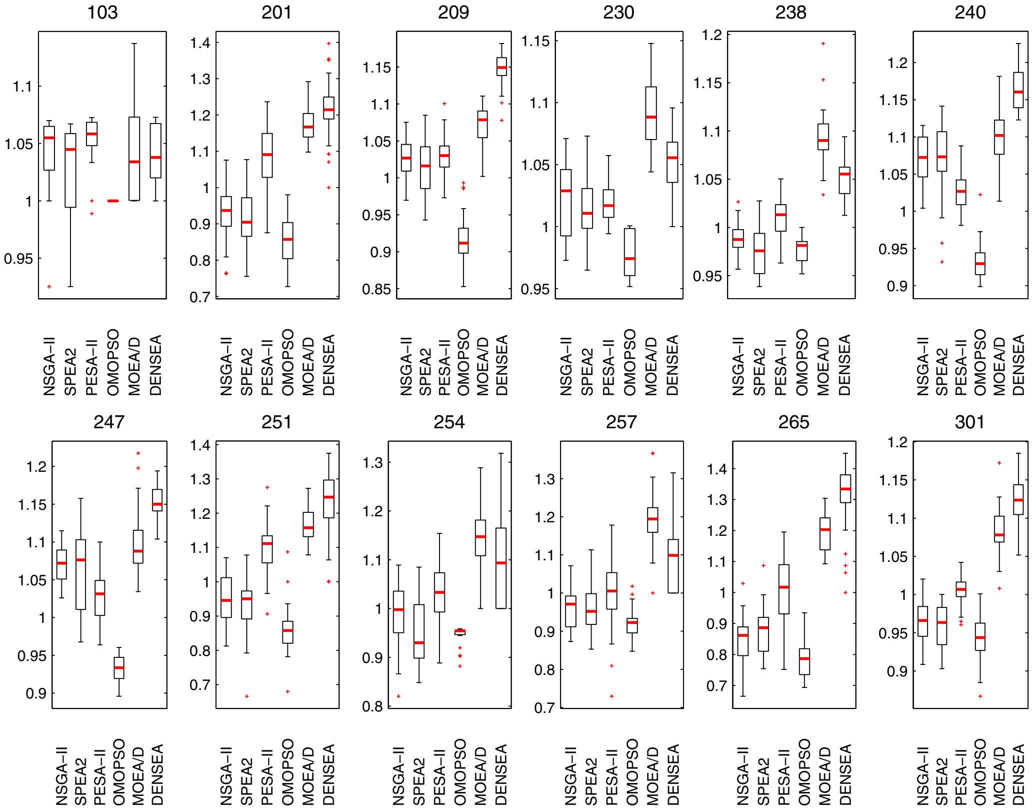

Fig. 2. Box plots based on the Δ index metric. Each rectangle contains six box plots representing the distribution of the Δ index values for each algorithm for a specific test case.

TABLE III
HOLM'S TEST FOR HYPERVOLUME METRIC

i	algorithm	z value	unadjusted p-value	$\frac{\alpha}{(k-i)}$, $\alpha = 0,1$
5	PESA-II	1,745743	0,080856	0,1
4	NSGA-II	2,07307	0,038166	0,05
3	SPEA2	3,491486	0,00048	0,033333
2	MOEA-D	4,964457	6,89E-07	0,025
1	DENSEA	6,382873	1,74E-10	0,02

TABLE IV
HOLM'S TEST FOR Δ INDEX METRIC

i	algorithm	z value	unadjusted p-value	$\frac{\alpha}{(k-i)}$, $\alpha = 0,1$
5	SPEA2	1,745743	0,080856	0,1
4	NSGA-II	2,509506	0,01209	0,05
3	PESA-II	3,382377	0,000719	0,033333
2	MOEA-D	5,12812	2,93E-07	0,025
1	DENSEA	5,564556	2,63E-08	0,02

TABLE V
HOLM'S TEST FOR C METRIC

i	algorithm	z value	unadjusted p-value	$\frac{\alpha}{(k-i)}$, $\alpha = 0,1$
5	NSGA-II	2,182179	0,029096	0,1
4	PESA-II	2,291288	0,021947	0,05
3	SPEA2	3,927922	8,57E-05	0,033333
2	MOEA-D	4,691685	2,71E-06	0,025
1	DENSEA	6,546537	5,89E-11	0,02

V. CONCLUSIONS

Ontology alignment and the linked meta-matching problem are relevant issues for the data integration domain. Over years, evolutionary algorithms have emerged as the most suitable method for addressing these issues. In this paper, we perform a systematic study consisting in the comparison of six well-known multi-objectives EAs to address the meta-matching problem in particular. The comparison is carried out in terms of popular performance indices such as hypervolume, Δ index and C metric. As shown by a statistical procedure, OMOPSO outperforms the other considered EAs in solving the meta-matching problem.

REFERENCES

[1] A. Buccella, A. Cechich, and N. R. Brisaboa, "Ontology-Based Data Integration Methods: A Framework for Comparison," in *Revista Colombiana de Computacion*, vol. 6, no. 1, 2005.

[2] A.P. Sheth, "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics," *Interoperating Geographic Information Systems*, M.F. Goodchild, M.J. Egenhofer, R. Fegeas, C.A. Kottman, (Eds.) pp. 5-30, Kluwer, 1998.

[3] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hubner, "Ontology - based Integration of Information - A Survey of Existing Approaches," in Proc. of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, pp. 108-117, 2001.

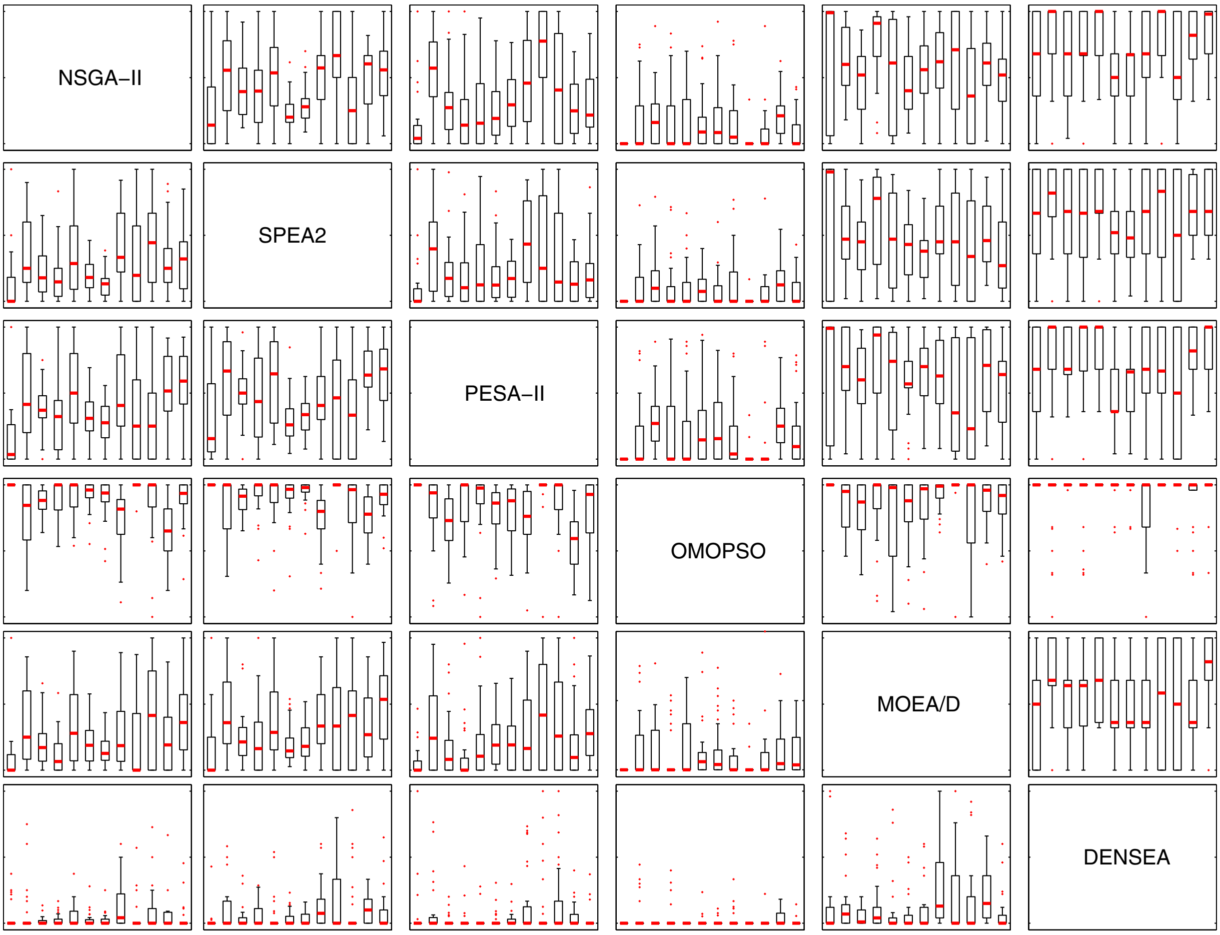

Fig. 3. Box plots based on the C metric. Each rectangle contains twelve box plots representing the distribution of the C values for a certain ordered pair of algorithms; the leftmost box plot relates to test case 103, the rightmost to case 301. The scale is 0 at the bottom and 1 at the top per rectangle. Furthermore, each rectangle refers to algorithm A associated with the corresponding row and algorithm B associated with the corresponding column and gives the fraction of B covered by A $C(A, B)$.

[4] J. Martinez-Gil and J. F. A. Montes, "Evaluation of two heuristic approaches to solve the ontology meta-matching problem," *Knowl. Inf. Syst.*, vol. 26, no. 2, pp. 225-247, 2011.

[5] J. Martinez-Gil, E. Alba and J. F. A. Montes, "Optimizing ontology alignments by using genetic algorithms," *Nature inspired Reasoning for the Semantic Web*, vol. 419, 2008.

[6] A.-L. Ginsca and A. Iftene, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," in The 9th Roedunet International Conference, pp. 118-122, 2010.

[7] G. Acampora, U. Kaymak, V. Loia and A. Vitiello, "Applying NSGA II for solving the Ontology Alignment Problem," in IEEE International Conference on Systems, man, and Cybernetics, pp. 1098-1103, 2013.

[8] X. Xue, Y. Wang and W. Hao, "Using MOEA/D for optimizing ontology alignments," *Soft Computing*, Springer Berlin Heidelberg, 2013.

[9] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evol. Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[10] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Technical Report 103, Swiss Federal Institute Technology, Zurich, Switzerland, 2001.

[11] D.W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 283-290, 2001.

[12] M. Reyes and C. Coello, "Improving PSO-based multiobjective optimization using crowding, mutation and e-dominance," in Evolutionary Multi-Criterion Optimization (EMO 2005), pp. 505-519, 2005.

[13] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[14] D. Greiner, J.M. Emperador and G. Winter, "Enhancing the multiobjective optimum design of structural trusses with evolutionary algorithms using DENSEA," 44th AIAA (American Institute of Aeronautics and Astronautics) Aerospace Sciences Meeting and Exhibit, 2006.

[15] T. R. Gruber, "A Translational Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, pp. 199-220, 1993.

[16] G. Acampora, V. Loia, S .Salerno and A. Vitiello, "A hybrid evolutionary approach for solving the ontology alignment problem," *Int. J. Intell. Syst.*, vol. 27, pp. 189-216, 2012.

[17] N. Srinivas and K. Deb, "Multi-objective function optimization using non-dominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221-248, 1995.

[18] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[19] E. Zitzler, K. Deb and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, 2000.

[20] T. Okabe, Y. Jin and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation,' The 2003 Congress on Evolutionary Computation, pp. 878-885, 2003

[21] G. Minella, R. Ruiz and M. Ciavotta, "A Review and Evaluation of Multiobjective Algorithms for the Flowshop Scheduling Problem," *INFORMS J. on Computing* vol. 20, no. 3, pp. 451-471, 2008.