J Data Semant (2012) 1:133–145
DOI 10.1007/s13740-012-0010-0

ORIGINAL ARTICLE

Breaking the Deadlock: Simultaneously Discovering Attribute Matching and Cluster Matching with Multi-Objective Metaheuristics

Haishan Liu · Dejing Dou · Hao Wang

Received: 1 September 2011 / Revised: 16 June 2012 / Accepted: 22 June 2012 / Published online: 4 August 2012

Abstract In this paper, we present a data mining approach to address challenges in the matching of heterogeneous datasets. In particular, we propose solutions to two problems that arise in integrating information from different results of scientific research. The first problem, *attribute matching*, involves discovery of correspondences among distinct numeric features (attributes) that are used to characterize datasets that have been collected and analyzed in different research labs. The second problem, *cluster matching*, involves discovery of matchings between patterns (clusters) across datasets. We treat both of these problems together as a multi-objective optimization problem. A multi-objective simulated annealing algorithm is described to find the optimal solution and compared with the genetic algorithm. The utility of this approach is demonstrated in a series of experiments using synthetic and realistic datasets that are designed to simulate heterogeneous data from different sources.

Keywords Multi-objective optimization · Cluster matching · Attribute matching · Metaheuristics

1 Introduction

The presence of heterogeneity among schemas supporting vast amount of information demands advanced solution for semantic integration of disparate data sources to facilitate interoperability and reuse of the information. The challenge is especially pronounced in many scientific domains where a massive amount of data are produced independently and thus each having their own data vocabulary. While manual integration is time-consuming and requires expensive specialized human capital, the development of automatic approaches becomes imminent to aid inter-institute collaborations. One purpose of the present paper was to suggest a method for solving a specific kind of schema/ontology matching problem under some severe constraints that can cause traditional methods to be ineffective. The constraints that we deal with are, namely, (1) little-to-no string-based or linguistic similarity between terminologies, and (2) all numeric typed data instances. This phenomenon is commonly seen in integrating scientific datasets which involves discovery of correspondences among distinct numeric-typed summary features ("attributes") that are used to characterize datasets that have been collected and analyzed in different research labs. We call this the *attribute matching* problem.

Another challenging task given the multiple data sources is to carry out meaningful meta-analysis that combines results of several studies on different datasets to address a set of related research hypotheses. Finding correspondences among distinct patterns that are observed in different scientific datasets is an example of meta-analysis. Supposing the patterns are derived by clustering analysis, this problem can be addressed by the application of cluster comparison (or cluster matching) techniques. Clustering is an unsupervised data mining task widely used to discover patterns and relationships in a variety of fields. The clustering result provides a pattern characterization from a data-driven perspective. If similar results are obtained across multiple datasets, this leads in turn to a revision and refinement of existing domain knowledge, which is a central goal of meta-analysis. However, there are noticeably few cluster comparison methods that are able to compare two clusterings derived from different datasets. The difficulty for the comparison is further exacerbated by the fact that the datasets may be described by

H. Liu (✉) · D. Dou · H. Wang
Computer and Information Science Department,
University of Oregon, Eugene, OR 97403, USA
e-mail: ahoyleo@cs.uoregon.edu

Table 1 Example distance matrices between (a) two sets of attributes and (b) two sets of clusters, respectively

	a'_1	$\cdots$	a'_m
a_1	$d_{11'}$	$\cdots$	$d_{1m'}$
$\vdots$		$\ddots$	
a_m	$d_{m1'}$		$d_{mm'}$

(A)

	c'_1	$\cdots$	c'_n
c_1	$d_{11'}$	$\cdots$	$d_{1n'}$
$\vdots$		$\ddots$	
c_n	$d_{n1'}$		$d_{nn'}$

(B)

attributes from heterogeneous schemas or ontologies. Even those methods that are able to measure clustering similarity across different datasets (e.g., the ADCO [1] method) have to assume the homogeneous meta-data (e.g., the same schemas).

Given this situation, in order to carry out cluster comparison for meta-analysis, researchers often need to perform ontology or schema matching first to mitigate the gap for meta-data. In previous work [18], we examined a practical attribute matching problem on neuroscience data where schema elements from one dataset share no lexical similarity with those from the other. Moreover, structural similarity is also limited. One can only resort to instance-based (extensional) methods. However, since all attributes are numerical, information clues available to an instance-level matcher is very restricted. Traditional instance-based matchers typically make use of constraint-based characterization, such as numerical value ranges and averages to determine correspondences. However, this is often too rough in the case of all-numerical dataset. Two attributes may have similar ranges and averages but totally different internal value distributions (an example is shown in Sect. 4.1). Given this, we propose to represent the attribute value distribution at a finer granularity by partitioning the values into groups. To do this, clustering is performed, and resulting clusters are then aligned across two datasets (assuming that the same pattern exists in both datasets). In this way, each attribute can be characterized by, instead of a single value, a vector of per-cluster statistical quantities (i.e., the *segmented statistical characterization*). A distance function can then be applied based on this representation. Table 1a shows an example distance table on the cross join of two sets of attributes. To discover attribute matching from this table can be reduced to solving a minimum assignment problem (assuming matching is bijective), which is a classical combinatory optimization problem that has a polynomial solution using the Hungarian Method [14].

Unfortunately, however, the above solution requires us to be able to align clusters across datasets, which is a difficult problem in its own right. If fully automated, as mentioned above, methods such as ADCO adopt a so-called *density profile* [1] representation of clusters that requires homogeneous meta-data or a priori knowledge about the attribute matching in heterogeneous scenarios. Then the cluster matching can be carried out in a similar manner to the attribute matching by casting to the assignment problem (see Table 1b, for example). This leads to a circular causality, or a deadlock, between the attribute matching (under the segmented statistical characterization) and cluster matching (under the density profile representation) problems—none of them can be solved automatically without the other one being solved first.

To solve this difficulty, in the present paper, viewing the two matching problems as combinatorial optimization problems with distinct yet interrelated objective functions, we propose a novel approach using a multi-objective heuristics to discover attribute matching and cluster matching simultaneously. The objectives in the optimization are to minimize distances of attribute matching and cluster matching, respectively. We explore the widely used simulated annealing algorithm as the metaheuristics algorithm and briefly compare its performance with the evolutionary multi-objective algorithm in experiments.

The rest of this paper is organized as follows: We review the basics of multi-objective optimization and describe the relationship between various components of the proposed method and existing methods in Sect. 2. We present detailed description of our method for simultaneously discovering attribute matching and cluster matching in Sect. 3. We report experimental and comparison results in Sect. 4. We discuss assumptions and implications of the proposed method in Sect. 5 and conclude the paper in Sect. 6.

2 Background and Related Work

2.1 The Multiobjective Optimization Problem and Pareto-Optimality

Multi-objective optimization problem (also called multi-criteria, multi-performance or vector optimization) can be defined mathematically as to find the vector $X = [x_1, x_2, \ldots, x_k]^T$ which satisfies the following m inequality constraints and l equality constraints:

$$g_i(X) \geq 0, \quad i = 1, 2, \ldots, m$$
$$h_i(X) = 0, \quad i = 1, 2, \ldots, l$$

and optimize the objective function vector

$$F(X) = [f_1(X), f_2(X), \ldots, f_N(X)]^T$$

where $X = [x_1, x_2, \ldots, x_k]^T$ is called the decision variable vector.

Real-life problems require simultaneous optimization of several incommensurable and often conflicting objectives. Usually, there is no single optimal solution, but there is a set of alternative solutions. These solutions are optimal in the sense that no other solutions in the search space are superior to each other when all the objectives are considered [25].

They are known as Pareto-optimal solutions. To define the concept of Pareto optimality, we take the example of a minimization problem with two decision vectors $a, b \in X$. Vector a is said to dominate b if

$$\forall i = \{1, 2, \ldots, N\} : f_i(a) \leq f_i(b)$$

and

$$\exists j = \{1, 2, \ldots, N\} : f_j(a) < f_j(b)$$

When the objectives associated with any pair of non-dominated solutions are compared, it is found that each solution is superior with respect to at least one objective. The set of non-dominated solutions to a multi-objective optimization problem is known as the Pareto-optimal set (Pareto front) [27].

2.1.1 Metaheuristics on Solving Multi-Objective Optimization Problems

Metaheuristics are used for combinatorial optimization in which an optimal solution is sought over a large, discrete search-space. Popular metaheuristics for combinatorial problems include simulated annealing by Kirkpatrick et al. [12], genetic algorithms by Holland [11]. Extensive previous research has been devoted to extend these methods to multi-objective optimization problems as discussed in the following, which yield sets of mutually non-dominating solutions that are an approximation to the true Pareto front. In Sect. 3 we explore in detail the multi-objective simulated annealing algorithm applied to the dual matching problem. We compare the performance with the multi-objective genetic algorithm in Sect. 4.

Simulated Annealing in Multi-Objective Optimization: Simulated annealing (SA) is based on an analogy of thermodynamics with the way metals cool and anneal. It has been proved to be a compact and robust technique. Simulated Annealing was started as a method or tool for solving single objective combinatorial problems; these days it has been applied to solve single as well as multiple objective optimization problems in various fields. A comprehensive survey can be found in [25].

Evolutionary Multi-Objective Optimization: Evolutionary multi-objective optimization covers the use of many types of heuristic optimizers inspired by the natural process of evolution. As in nature, a population of individuals (solutions to the problem) exist and, through a process of change and competition between these individuals, the quality of the population is advanced. Deb [3] provides an introduction of evolutionary algorithms (e.g., genetic algorithm) for multi-objective as the state of the art.

2.2 The Schema Matching Problem

Our study of matching alternative attribute sets is closely related to the schema matching problem in data integration. According to the type of instance value, various instance-based approaches have been developed in previous research. For example, for textual attributes, a linguistic characterization based on information retrieval techniques can be applied [21]; for nominal attributes, evaluation of the degree of overlap of instance values is a preferred approach. Larson et al. [15] and Sheth et al. [23] discussed how relationships and entity sets could be integrated primarily based on their domain relationships. Similarity of partially overlapped instance set can be also calculated based on measures such as Hamming distance and Jaccard coefficient; for numeric attributes, most methods use aggregated statistics to characterize the attributes, e.g., 'SSN' and 'PhoneNo' can be distinguished based on their respective patterns [21]. Hybrid systems that combine several approaches to determine matching often achieve better performance. For example, SemInt [16] is a comprehensive matching prototype exploiting up to 15 constraint-based and 5 content-based matching criteria. The LSD (Learning Source Descriptions) [6] system uses several instance-level matchers (learners) that are trained during a preprocessing step. The iMAP [4] system uses multiple basic matchers, called searches, e.g., text, numeric, category, and unit conversion, each of which addresses a particular subset of the match space.

Due to the nature of many scientific datasets, we face several unique challenges. First, the data under study is semi-structured, thus invalidating those matching methods that presume a complete, known-in-advance schematic structure. In addition, totally different labels (usually acronyms or pseudowords) are widely adopted for the same or similar metrics, rendering lexical similarity-based methods unsuitable. Moreover, an important limitation of previous instance-based matching methods is their inability to handle numerical instances appropriately in certain domain applications. They use statistical characterization extracted from the numerical instances, such as range, mean and standard deviation, to determine matching. However such information is too rough to capture patterns in data that are crucial in determining the correspondence.

2.3 The Cluster Matching Problem

The cluster matching (cluster comparison) problem is related to the cluster validity problem, especially the technique of external/relative indexing that aims at comparing two different clustering results. Popular methods in this field, including the Rand index [22], Jaccard index [10], normalized mutual information [7], etc., are mostly based on examining membership of points to clusters. However, the basis of these

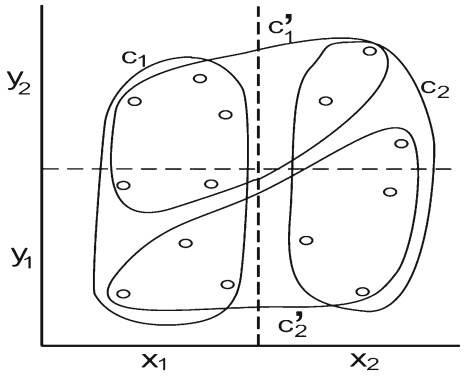

Fig. 1 Two clusterings $C = \{c_1, c_2\}$ and $C' = \{c'_1, c'_2\}$. Two attributes X (attribute 1) and Y (attribute 2) are discretized into two bins each. See [1] for details

methods is the comparison of different clustering results on the same dataset.

By contrast, in the present case we are aiming to match clusters across datasets that contain non-overlapping observations. Thus, membership-based cluster validity criteria are unsuitable. A recent clustering similarity index known as ADCO (Attribute Distribution Clustering Orthogonality) proposed by Bae et al. [1] can match clusters from non-overlapping datasets. The ADCO measure determines the similarity between two clusters based on their *density profiles*, which incorporate distribution information of data points along each attribute. The density profile representation of clusters are defined as follows.

Density Profile: To represent clusters using density profiles, the attribute's range in each cluster is first discretized into a number of bins, and the similarity between two clusters corresponds to the number of points of each cluster falling within these bins. The formal definition for this number of points is the *density* of an attribute-bin region for cluster c_k in clustering result C, denoted as $dens_C(k, i, j)$. It refers to the number of points in the region (i, j)—the j-th bin of the i-th attribute—that belongs to the cluster c_k of the clustering result C. For example, for clustering result C in Fig. 1, $dens_C(1, 1, 1) = 8$, because there are 8 data points in region $(1, 1)$—the first bin of the first attribute x—that belongs to the first cluster c_1.

The density profile vector V_C for a clustering result C is formally defined as an ordered tuple:

$$V_C = \Big[dens_C(1,1,1),\ \ldots,\ dens_C(1,1,Q),\ dens_C(1,2,1),\ \ldots,\ dens_C(1,M,Q),\ dens_C(2,1,1),\ \ldots,\ dens_C(N,M,Q) \Big], \quad (1)$$

where Q is the number of bins in each of the M attributes, and N is the number of clusters in C.

The ADCO measure: After the density profile vectors of two clustering results C and C' are obtained, the degree of similarity between C and C' can be determined by calculating the dot product of the density profile vectors: $sim(C, C') = V_C \cdot V_{C'}$.

The $ADCO(C, C')$ measure is defined as $sim(C, C')$ normalized by the maximum achievable similarity when using either of the two clusterings:

$$ADCO(C, C') = \frac{sim(C, C')}{NF(C, C')},$$

where $NF(C, C') = max\big[sim(C, C),\ sim(C', C')\big]$.

2.4 Collective Classification for Schema and Data Matching

Collective classification in relational data has become an important and active research topic in the last decade, where class labels for a group of linked instances are correlated and need to be predicted simultaneously [13]. It has been applied to tackle multiple integration problems that are traditionally solved independently. Wick et al. [26] describe a discriminatively trained model based on Markov random field to perform joint reasoning about schema matching, coreference, and canonicalization. Namata et al. [20] proposed an approach consisting of coupled collective classifiers to discover a latent graph structure underlying an observed one by addressing entity resolution, link prediction, and node labeling simultaneously. The difference between these previous methods and our proposed method mainly lies in the fact that we do not require a training phase. Instead, the matchings are discovered by simultaneously optimizing interrelated objective functions which circumvents the labor for acquiring labeled data and the expense of statistical inference.

Last but not least, we have made important extensions in this paper compared with our conference paper version [17]. In this paper, we implement an evolutionary multi-objective approach as the metaheuristics algorithm to discover attribute and cluster matching. We have also conducted more experiments on real-world datasets to demonstrate the utility of the proposed method.

3 Method

3.1 The Multi-Objective Simulated Annealing Framework

Problem Definition: We tackle two data integration tasks in this work, namely, the attribute matching and cluster matching problems. We cast the dual matching problems to a multi-objective optimization problem so that the matchings can be solved simultaneously. The two objective functions to be optimized are defined as the total distance of matched elements in attribute and cluster matching, respectively. To this

end, we explore methods to represent attributes and clusters so that the distance measure can be reasonably defined. We assume that the optimal matching lies at the Pareto front in this multi-objective problem.

We use metaheuristics search algorithm to solve this multi-objective optimization problem. In the following we describe the widely used simulated annealing algorithm and how it can be adapted to multi-objective optimization and applied to solve the matching problems. Later in the Experiment Section, we briefly describe an evolutionary multi-objective algorithm (i.e., genetic algorithm) and compare their performance.

To solve the dual matching problems, we adopt a strategy of multi-objective simulated annealing (MOSA) described in [24], in which the acceptance criterion in the simulated annealing process is Pareto-domination based fitness. Fitness of a solution is defined as one plus the number of dominating solutions in Pareto-optimal set. The larger the value of fitness, the worse is the solution. The probability of making the transition from the current state $\mathbf{X}$ to a candidate new state $\mathbf{X}'$ is specified by an acceptance probability function $P = exp(\frac{-\Delta S}{T})$, where ΔS is the change of fitness and T is a global time-varying parameter called the temperature. The P function is chosen so that the probability of accepting a move decreases when the difference of fitness increases—that is, small worsening moves are more likely than large ones. Initially, the temperature is high so any move may be accepted, which makes it possible to explore the full solution space. As the number of iterations increases, temperature decreases, and fitness difference between the current and generated solutions may increase. Both of them make the acceptance move more selective and it results in a well-diversified solution in true Pareto-optimal solutions. Details of our adaptation of the above multi-objective simulated annealing framework is outlined in Algorithm 1.

Mathematically, the processes involved in the proposed multi-objective simulated annealing framework can be defined as follows:

$$
\begin{aligned}
X &= [x_a, x_c] \\
F &= [f_a, f_c] \\
P_a([x_a^{(n-1)}, x_c^{(n-1)}]) &= [x_a^{(n)}, x_c^{(n-1)}] \\
P_c([x_a^{(n-1)}, x_c^{(n-1)}]) &= [x_a^{(n-1)}, x_c^{(n)}] \\
G_{c|a}([x_a^{(n)}, x_c^{(n-1)}]) &= [x_a^{(n)}, x_c^{(n)}] \\
G_{a|c}([x_a^{(n-1)}, x_c^{(n)}]) &= [x_a^{(n)}, x_c^{(n)}] \\
G \circ P([x_a^{(n-1)}, x_c^{(n-1)}]) &= [x_a^{(n)}, x_c^{(n)}]
\end{aligned}
$$

X is the decision vector that contains two variables for attribute matching, x_a, and cluster matching, x_c, respectively (details in Sect. 3.2). F is the objective function vector that contains two criterion functions (f_a and f_c) to evaluate attribute matching and cluster matching decisions (details in Sect. 3.4). P is the random perturbation function that takes a decision vector in the $(n-1)$th iteration and partially advances it to the nth iteration (we use P_a or P_c to distinguish between the random selections). The partial candidate decision generation function G takes the output of P and fully generate a decision vector for the nth iteration (by advancing the left-out variable in P to its nth iteration). Thus, the compound function $G \circ P$ fulfills the task of generating an nth-iteration candidate decision vector given the $(n-1)$th one (details in Sect. 3.5.2).

Algorithm 1 Multi-Objective Simulated Annealing

Require: Empty Pareto-optimal set of solutions Σ
Require: Empty current decision vector $\mathbf{X} = [x_a, x_c]$
Require: Initial temperature T
  $count = 0$
  while $T > threshold$ **do**
    $initialize(\mathbf{X})$
    Put $\mathbf{X}$ in Σ
    $\mathbf{X}' = generate\_solution(\mathbf{X})$
    $S_{\mathbf{X}'} = evaluate\_solution(\mathbf{X}')$
    $\Delta S = S_{\mathbf{X}'} - S_X$
    if $r = rand(0, 1) < exp(\frac{-\Delta S}{T})$ **then**
      $\mathbf{X} = \mathbf{X}'$
      $S_X = S_{X'}$
    end if
    count = count + 1
    //Periodically restart
    if $count == restart\_limit$ **then**
      $\mathbf{X} = select\_random\_from\_Pareto(\Sigma)$
      continue
    end if
    $reduce\_temperature(T)$
  end while

3.2 Decision Variable

The domains of the decision variables in the matching problems take values on a permutation space. In other words, by formalizing the problem of finding correspondent elements of two sets S and S' of cardinality n as an optimization problem, the solution is completely specified by determining an optimal permutation of $1, \ldots, n$. For instance, for two sets of three elements, their indexes range over $\{0, 1, 2\}$. Applying a permutation $\pi = \{2, 0, 1\} \in P_3$ on S' can be viewed as creating a mapping (bijection) from elements on the new positions of S' to elements on the corresponding positions in S. In this example, the permutation π on S' specifies the following correspondences: $S_0 \leftrightarrow S'_2$, $S_1 \leftrightarrow S'_0$, and $S_2 \leftrightarrow S'_1$.

Formally, let P_n ($n \in \mathbb{N}$) be the symmetric group of all permutations of the set $\{1, 2, \ldots, n\}$. Given two sets S and S' with the same cardinality of n, performing identity permutation on one set and an arbitrary permutation $\pi \in P_n$ on the other specifies a matching (or mathematically speaking, mapping) between the two sets. In the multi-objective optimization formalism for solving attribute matching and cluster

matching problems, the decision vector has two variables: $X = [x_a, x_c]$. If we have M attributes and N clusters to match respectively, then $x_a \in P_M$ and $x_c \in P_N$.

3.3 Data Representation

The central objects of interest in our study, namely, the numeric-typed attributes and clusters, need to be represented in a way that meaningful quantities can be defined to measure the "goodness" of a matching decision. To this end, we propose to use the *segmented statistical characterization* to represent attributes, and the *density profiles* to represent clusters. Details of these representations are described below.

3.3.1 Representation of Attributes

Numeric-typed attributes can be represented by the segmented statistical characterization, in which data instances are first partitioned into groups (e.g., through unsupervised clustering) and then characterized by a vector of indicators, each denoting a statistical characterization of the corresponding group. For example, if values of an attribute A are clustered into n groups, then it can be represented by a vector of segmented statistical characterization as follows:

$$V_A = [\mu_1, \mu_2, \ldots, \mu_n],$$

where we choose the mean value μ_i for cluster i as the statistical indicator in our implementation.

3.3.2 Representation of Clusters

Clusters can be represented by density profiles [1] as described in Sect. 2. The attribute's range in each cluster is discretized into a number of bins, and the similarity between two clusters corresponds to the number of points of each cluster falling within these bins. Given this, density profile vector V_C for a clustering C is formally defined as an ordered tuple by Eq. 1 where $dens_C(k, i, j)$ refers to the number of points in the region (i, j)—the j-th bin of the i-th attribute—that belongs to the cluster c_k of clustering C.

3.4 Objective Functions

The objective functions in the attribute matching and cluster matching problems are criteria to evaluate the "goodness" of matchings. We use the sum of pair-wise distances between matched elements (see Table 1 for example) as the objective function. Given this, to determine the form of objective functions amounts to defining proper pair-wise distance measures for the attribute and cluster matching problems, respectively, as detailed in the following.

3.4.1 Distance Function Between Two Attributes

The pairwise distance between two attributes is defined as the Euclidean distance between their segmented statistical characterization vectors, and f_a calculates the sum of pair-wise distances under the attribute matching specified by x_a:

$$\begin{aligned} f_a(x_a) &= \sum_{k=1}^{M} \mathcal{L}\left((V_a)^k,\ (V'_a)^{x_a(k)}\right) \\ &= \sum_{k=1}^{M} \sqrt{\sum_{i=1}^{N} \left(\mu_i^k - (\mu')_i^{x_a(k)}\right)^2} \end{aligned} \tag{2}$$

where $x_a \in P_M$.

3.4.2 Distance Function Between Two Clusters

The ADCO similarity described in Sect. 2.3 can be transformed to a distance defined as follows [1]:

$$D_{ADCO}(C, C') = \begin{cases} 2 - ADCO(C, C'), & \text{if } C \neq C' \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

We use D_{ADCO} as the pair-wise distance between two clusters under the density profile representation, and f_c calculates the sum of pair-wise distances under the cluster matching specified by x_c

$$\begin{aligned} f_c(x_c) &= \sum_{k=1}^{N} D_{ADCO}\left((V_c)^k,\ (V'_c)^{x_c(k)}\right) \\ &= \sum_{k=1}^{N} \left(2 - \sum_{i=1}^{M}\sum_{j=1}^{Q} \left(dens(k,i,j) \times dens(x_c(k),i,j)\right) \Big/ \right. \\ &\left. \max\left[\sum_{i=1}^{M}\sum_{j=1}^{Q} dens(k,i,j)^2, \sum_{i=1}^{M}\sum_{j=1}^{Q} dens(x_c(k),i,j)^2\right]\right), \end{aligned} \tag{4}$$

where $x_c \in P_N$.

3.5 Generation of New Solution

In each iteration of the simulated annealing process, we randomly generate the candidate decision in the neighborhood of the last-iteration decision by applying two consecutive processes, namely, the random perturbation and the partial candidate decision generation, as described below.

3.5.1 Random Perturbation

In each iteration, we select at random one variable (either x_a or x_c) in the decision vector and perturb it by randomly swapping two positions in the selected variable. This advances that variable from the $(n-1)$th iteration to the nth iteration. Then

the following partial candidate generation process is carried out to bring the other variable also to the nth iteration.

3.5.2 Partial Candidate Decision Generation

Given $x_c^{(n)}$, derive $x_a^{0(n)}$:

$$
\begin{aligned}
x_a^n &= \arg\min_{\pi} \; f_a(\pi, x_c^{(n)}) \\
&= \arg\min_{\pi} \sum_{k=1}^{M} \mathcal{L}\left((V_a)^k, \; (V'_a)^{\pi(k)}\right) \\
&= \arg\min_{\pi} \sum_{k=1}^{M} \sqrt{\sum_{i=1}^{N} \left(\mu_i^k - (\mu')_{x_c^{(n)}(i)}^{\pi(k)}\right)^2}
\end{aligned} \tag{5}
$$

Given $x_a^{(n)}$, derive $x_c^{(n)}$:

$$
\begin{aligned}
x_c^n &= \arg\min_{\pi} \; f_c(\pi, x_a^{(n)}) \\
&= \arg\min_{\pi} \sum_{k=1}^{N} D_{ADCO}\left((V_c)^k, \; (V'_c)^{\pi(k)}\right) \\
&= \arg\min_{\pi} \sum_{k=1}^{N} \left(\sum_{i=1}^{M}\sum_{j=1}^{Q} \left(dens(k,i,j) \times dens(\pi(k), x_a^{(n)}(i), j)\right) \Big/ \right. \\
&\qquad \left. \max\left[\sum_{i=1}^{M}\sum_{j=1}^{Q} dens(k,i,j)^2, \sum_{i=1}^{M}\sum_{j=1}^{Q} dens(\pi(k), x_a^{(n)}(i), j)^2 \right] \right)
\end{aligned} \tag{6}
$$

To calculate π that satisfies Eqs. 5 and 6, rather than iterating through all possible permutations, we can consider the equation as a minimum-cost assignment problem. Table 1a, for example, illustrates a distance table between two attribute sets A and A'. Matching of the two sets can be considered as an assignment problem where the goal is to find an assignment of elements in $\{A_i\}$ to those in $\{A'_i\}$ that yields the minimum total distance without assigning each A_i more than once. This problem can be efficiently solved by the Hungarian Method in polynomial time of $O(K_{min}^3)$ [14]. It is worth noting that by formulating the problem as the assignment problem, we assume the matching between two sets to be a one-to-one function.

4 Experiment

Because we are interested in understanding the property of the Pareto front obtained by our method, we conducted a series of experiments to highlight tradeoffs of the objectives functions. First, to illustrate that the proposed method is indeed capable of determining matchings between numeric-typed attributes and clusters, we synthesized a dataset simulating some extreme conditions under which previous methods are ineffective. Also, from the results obtained on the synthetic dataset, we empirically study tradeoffs between the two objective functions. Then, to evaluate the scalability of the method, we carry out a series of tests on a set of data with varied sizes. Finally, encouraged by these results, we applied our methods to actual neuroscience ERP (event-related potentials) data to highlight the applicability of our method to the neuroscience domain.

4.1 Synthetic Dataset

4.1.1 Data Generation

In the synthetic dataset, tables are generated in such a way that each attribute consists of several Gaussians with distinct mean and standard deviation, and for one attribute in the source table, there exists exactly one attribute in the target table whose Gaussians possess the same configuration (hence they match each other). However if the attribute is viewed as a single distribution, as is typical in previous methods, its mean and standard deviation would be indistinguishable from those of other attributes in the same table. For example, Fig. 2 illustrates the value distributions of three attributes (a_1, a_2, and a_3) from one dataset and their corresponding counterparts (a'_1, a'_2, and a'_3) from another.

4.1.2 Results

Figure 3 illustrates the Pareto front obtained from matching two synthetic datasets, each having 20 attributes and 5 clusters. Most notably, the gold standard results for both attribute matching and cluster matching are obtained from the leftmost point on the Pareto front. In other words, given the decision variables (X) corresponding to that point, we obtained 100 % correct matching results. We further observed that in our subsequent tests on other synthetic datasets with varied number of attributes and clusters, the derived Pareto fronts all contain the gold standard result, and the point corresponding to the gold standard can always be found towards the minimum end of f_a. Given this, we propose the following method to reduce the Pareto-optimal set to a single point corresponding to the most favored choice (X^*) in the decision space. The idea is to find the decision with the minimum weighted sum of objective values in the obtained Pareto-optimal set, i.e., $X^* = \arg\min_X \left[\alpha f_a(X) + \beta f_c(X)\right]$, where α and β are weights. We first conducted preliminary experiments to determine the best values for α and β (0.8 and 0.2, respectively) and used them in all subsequent experiments. This method works markedly well on the synthetic datasets. For all the tests described in Table 2, 100 % correct results for both attribute and cluster matchings are obtained (hence we omit the precision in the table).

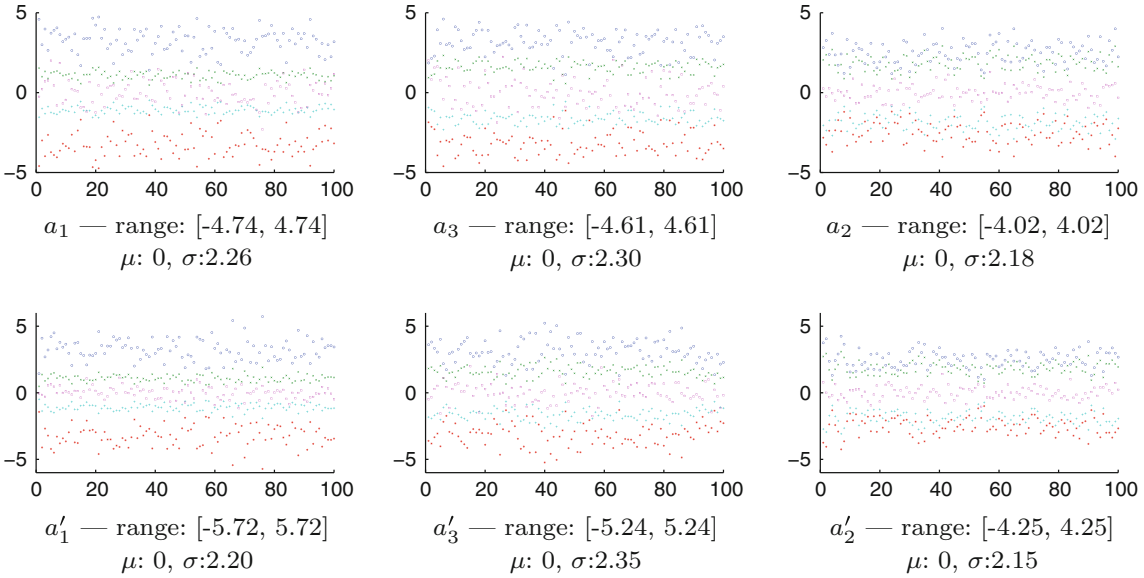

Fig. 2 Scatter plots of data instances from three sample attributes in one synthetic dataset (*upper frame*) and those of their corresponding attributes from another (*lower frame*) are illustrated to show their respective value distributions

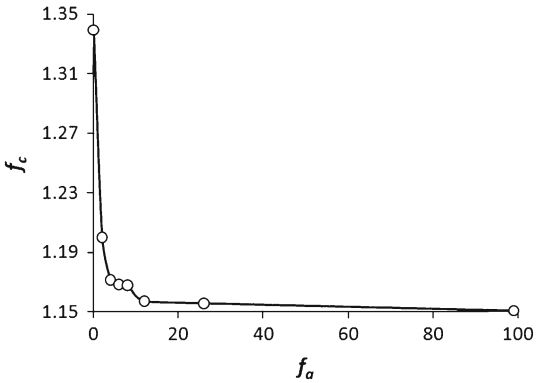

Fig. 3 An example Pareto front obtained from matching two synthetic datasets with 20 attributes and 5 clusters

Table 2 Running time of the annealing process on synthetic datasets with varied configurations of attribute and cluster sizes

# Attributes	# Clusters	Time (s)
5	20	0.28
20	20	1.81
20	40	7.04
20	60	17.80
40	20	4.66
40	40	11.74
40	60	25.93
60	20	10.95
60	40	20.70
60	60	37.35
100	100	172.23

The time is obtained by averaging over results of 5 runs of each test

4.1.3 Running Time

We systematically altered the number of attributes and clusters present in the data and conducted a series of tests to show the scalability of the proposed method. The running time under different configurations is reported in Table 2. The time is calculated by averaging over 5 runs of each test (on a 2.53 GHz dual-core CPU with 4 gigabytes memory), each run having 1000 iterations in the simulated annealing process. The main computationally expensive part of the annealing process is the generation of new candidate solution phase (function G) in which an assignment problem is solved using the Hungarian method. The complexity of the Hungarian method is cubic and is already the most efficient algorithm for solving the assignment problem (e.g., a brute force algorithm has a factorial complexity). Fortunately, rarely is the case that the number of attributes or clusters is large in real-world scenarios where the proposed technique is needed. For reasonable configurations in most practical applications, the computation time is within a tractable range as shown in Table 2.

4.2 Neuroscience Dataset

4.2.1 Data Acquisition

To address the problems of attribute and cluster matching in a real-world neuroscience application, we used a set of realistic simulated ERP (event-related potentials) datasets, which were designed to support evaluation of ERP analysis methods [8]. The datasets were specifically designed to

simulate heterogeneous data from different groups of subjects under different conditions (via distinct simulated brain activities), as well as distinct measurement methods (spatial and temporal metrics), and distinct patterns (reflecting two different pattern decomposition techniques). Real ERP data arise from superposition of latent scalp-surface electrophysiological patterns, each reflecting the activity of a distinct cortical network that cannot be reconstructed from the scalp-measured data with any certainty. Thus, real ERP data are not appropriate for evaluation of ERP pattern mapping. By contrast, simulated ERP data are derived from known source patterns and therefore provide the necessary gold standard for evaluation of our proposed methods.

The raw data for this study consist of 80 simulated event-related potentials (ERPs), in which each ERP comprises simulated measurement data for a particular subject ($n = 40$). The 40 simulated subjects are randomly divided into two 20-subject groups, SG1 and SG2, each containing 40 ERPs (20 subjects in 2 experimental conditions). Each ERP consists of a superposition of 5 latent varying spatiotemporal patterns. These patterns were extracted from the two datasets, SG1 and SG2, using two techniques: temporal Principal Components Analysis (tPCA) and spatial Independent Components Analysis (sICA), two data decomposition techniques widely used in ERP research [5]. To quantify the spatiotemporal characteristics of the extracted patterns, two alternative metric sets, m1 and m2, were applied to the two tPCA and the two sICA-derived datasets. For a complete explanation of these alternative metrics, please see Appendix in [8].

In summary, the simulated ERP data generation process yielded eight test datasets in total, reflecting a 2 (attribute sets) $\times$ 2 (subject groups) $\times$ 2 (decomposition methods) factorial design. Therefore, for each attribute sets there are 4 datasets generated from different combinations of subject groups and decomposition methods, resulting $4 \times 4 = 16$ cases for the studies of attribute matching and cluster matching. The reason to include such variabilities was to test the robustness of our matching method to different sources of heterogeneities across the different datasets. Within all test datasets, 5 major ERP spatiotemporal patterns are present. They are P100, N100, N3, MFN, and P300. These patterns can be identified in the datasets by clustering analysis. Pretending that the latent patterns underlying discovered clusters are unknown, we hope to match clusters across datasets to recover the fact that the same patterns are present in all datasets.

4.2.2 Results

We applied the weighted sum method as the post-process step after obtaining the Pareto-optimal solutions to determine the most favored choice using the parameters (α and β) discovered in the preliminary experiments on synthetic datasets (cf. Sect. 4.1). The accuracy of attribute matching and cluster matching along with the number of points in the Pareto front are listed in Table 3 (all these results are obtained by taking average from 5 runs for each test case).

Table 3 Matching performance of the proposed method with MOSA on the 16 test cases from the neuroscience dataset

Test case	Source params	Target params	P_a	P_c	$\lvert\Sigma\rvert$
1	⟨ SG1, sICA, m1 ⟩	⟨ SG1, sICA, m2 ⟩	13/13	5/5	5
2	⟨ SG1, sICA, m1 ⟩	⟨ SG2, sICA, m2 ⟩	13/13	5/5	6
3	⟨ SG1, sICA, m1 ⟩	⟨ SG1, tPCA, m2 ⟩	10/13	5/5	6
4	⟨ SG1, sICA, m1 ⟩	⟨ SG2, tPCA, m2 ⟩	7/13	3/5	8
5	⟨ SG2, sICA, m1 ⟩	⟨ SG1, sICA, m2 ⟩	11/13	3/5	7
6	⟨ SG2, sICA, m1 ⟩	⟨ SG2, sICA, m2 ⟩	13/13	5/5	7
7	⟨ SG2, sICA, m1 ⟩	⟨ SG1, tPCA, m2 ⟩	10/13	5/5	6
8	⟨ SG2, sICA, m1 ⟩	⟨ SG2, tPCA, m2 ⟩	9/13	2/5	8
9	⟨ SG1, tPCA, m1 ⟩	⟨ SG1, sICA, m2 ⟩	7/13	5/5	4
10	⟨ SG1, tPCA, m1 ⟩	⟨ SG2, sICA, m2 ⟩	8/13	5/5	6
11	⟨ SG1, tPCA, m1 ⟩	⟨ SG1, tPCA, m2 ⟩	11/13	5/5	6
12	⟨ SG1, tPCA, m1 ⟩	⟨ SG2, tPCA, m2 ⟩	7/13	3/5	5
13	⟨ SG2, tPCA, m1 ⟩	⟨ SG1, sICA, m2 ⟩	7/13	3/5	5
14	⟨ SG2, tPCA, m1 ⟩	⟨ SG2, sICA, m2 ⟩	9/13	5/5	6
15	⟨ SG2, tPCA, m1 ⟩	⟨ SG1, tPCA, m2 ⟩	10/13	3/5	8
16	⟨ SG2, tPCA, m1 ⟩	⟨ SG2, tPCA, m2 ⟩	8/13	3/5	8

The source and target parameter configuration of the data acquisition process of each test case are shown. P_a and P_c denote the accuracy of attribute matching and cluster matching, respectively. Σ is the number of points in the obtained Pareto-front. The quantities listed in the table are obtained by averaging over 5 runs of each test

It can be observed from the results in Table 3 that more different factors involved in the acquisition of the two datasets for matching can negatively affect the matching performance. For example, in test case 1, the two datasets are drawn from the same subject group (SG1) and preprocessed using the same decomposition method (sICA), whereas in test case 4, the subject groups and decomposition methods are all different, resulting in greater variability and hence the performance is less satisfactory.

It is worth noting that our method greatly outperforms a baseline method called WS (see Fig. 4) that determines attribute matching based on data distribution at the whole attribute level, which is typical in previous systems such as SemInt [16]. In this figure we also demonstrate the accuracy of the segmented statistics characterization with expert-labeled patterns, meaning that the data are partitioned and aligned in the most accurate way, which marks the best achievable attribute matching performance. But it is not feasible because, as mentioned in Sect. 1, manually recognizing patterns (partitioning data) and aligning them across datasets require a priori knowledge of attributes in the datasets which is exactly what the problem of attribute matching tries to

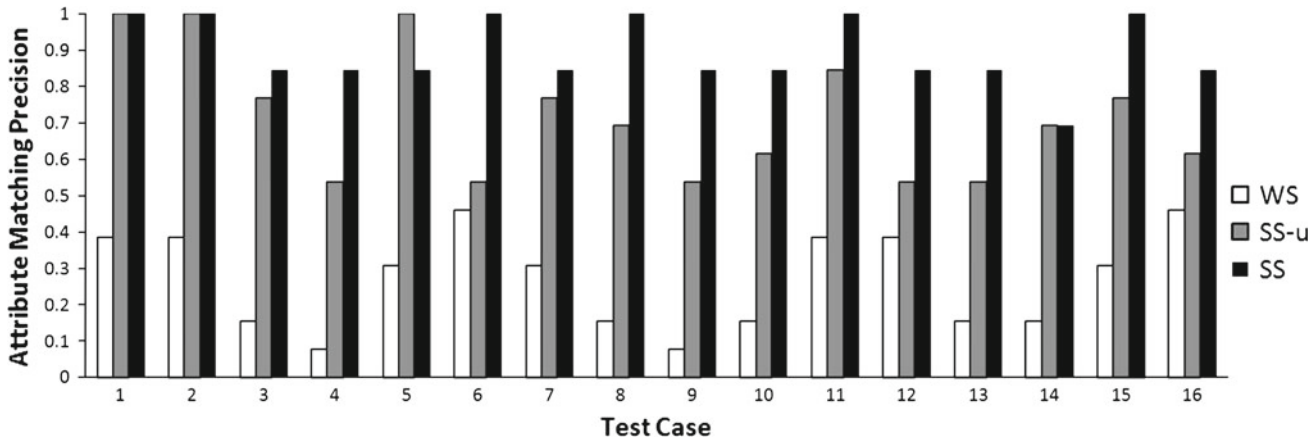

Fig. 4 A comparison of the attribute matching accuracy of three methods on the 16 test cases of the neuroscience dataset. The three methods being compared are matching based on whole-attribute statistics (*WS*), segmented attribute statistics without knowing a priori cluster matching (*SS-u*), and segmented attribute statistics with expert-aligned clusterings (*SS*)

discover (the circular causality problem). On the other hand, our method does not require human involvement (except the specification of the number of clusters (patterns) present in the data to run the clustering analysis) in determining both the attribute matching and cluster matching and is able to achieve close-to-optimal results.

4.3 Comparison with Multi-Objective Genetic Algorithm

The concept of Genetic Algorithm (GA) was developed by Holland and his colleagues [11]. GA is first inspired by the evolutionary process in which weak and unfit species within their environment are faced with extinction and stronger ones have greater opportunities to pass their genes to next generation. Comparing with Simulated Annealing (SA), Genetic Algorithm often offers a different perspective in the field of numerical optimization. Starts from a number of random generated population, cross over and evolve; GA has the ability to search in parallel around different and often fully scattered instances in the solution space, in contrast to the "single thread" search in Simulated Annealing. In this paper, we also implemented the Multi-Objective Genetic Algorithm (MOGA) as the metaheuristics to solve the dual matching problem.

To compare the performance between GA and SA, we first carry out an experiment on the same set of neuroscience data, as shown in Table 4. The iteration parameters of both algorithms are tuned so that the convergence time are about the same. The performances are then compared under such setting. We manually examine the Pareto front derived in each test case and find the solution that is closest to the gold standard and the accuracy of which is reported in Table 4 (averaged over 5 independent runs).

Table 4 Matching performance of the proposed method with MOGA on the 16 test cases from the neuroscience dataset

Test case	P_a (%)	P_c (%)	Σ
1	100	100	9
2	98.2	96.6	10
3	53.4	98.0	9
4	53.3	98.0	11
5	100	98.2	5
6	71.2	96.0	6
7	59.4	94.4	6
8	59.7	98.8	6
9	25.2	100.0	6
10	38.5	100.0	5
11	77.7	99.2	7
12	69.2	100.0	9
13	38.7	100.0	9
14	40.3	98.8	11
15	45.0	96.0	8
16	84.6	98.8	16

The source and target parameter configuration of each test case is the same as in Table 3

The number of population kept in each generation is an important parameter regarding the complexity and performance in MOGA. Intuitively, the more instances we keep, the broader the search space we can explore in each generation. Table 4 shows the result with the number of population set to 4. We have also tested other settings and found out that the accuracy in most cases increases with the number of population, but in rare cases the performance deteriorates. The overall performance of MOGA is comparable to that of MOSA but appears to be less robust. It is worth noting that

Table 5 Summary of the statistical characteristics of attributes in the Wine Quality dataset

	Fixed acidity	Volatile acidity	Citric acid	Residual sugar	Chlorides	Free sulfur dioxide	Total sulfur dioxide	Density	pH	Sulphates	Alcohol	Quality
Mean												
data1	6.86	0.28	0.34	6.35	0.05	35.58	138.98	0.99	3.19	0.49	10.53	5.88
data2	6.85	0.28	0.33	6.43	0.05	35.02	137.68	0.99	3.19	0.49	10.49	5.88
Stdev												
data1	0.84	0.1	0.12	4.98	0.02	16.4	41.86	0.02	0.16	0.11	1.25	0.89
data2	0.86	0.1	0.12	5.16	0.02	17.61	43.18	0	0.15	0.12	1.22	0.89

Table 6 Performance of the proposed method with MOSA and MOGA as metaheuristics, respectively, on the Wine Quality dataset

MOSA accuracy (%)	MOSA running time (ms)	MOGA accuracy (%)	MOGA running time (ms)
95.5	517	92.3	3356

the metaheuristics (MOSA and MOGA) we employed in the experiments are simple algorithms. More modern and sophisticated methods that explore various fitness assignment procedure, elitism, or diversification approaches will very likely improve the performance.

In order to further validate our method, we implement our method also on a real-world wine quality dataset [2] which is available through the UCI machine learning repository.[1] This dataset has 12 attributes and 4898 records. We apply uniform sampling to split it into two equal-sized subsets. The attributes are anonymized and randomly reordered in each subset to generate artificial heterogeneity.

We then apply the proposed method with MOSA and MOGA as metaheuristics, respectively. The test is focused on attribute matching because the gold standard is known while the gold standard of cluster matching is unknown. Table 5 summarizes the statistics for each attribute in the dataset. For both MOSA- and MOGA-derived Pareto optimal solutions, we manually select the one that is closest to the gold-standard matching (e.g., the solution with 10 out 12 attributes matched correctly). Each metaheuristics is invoked five times and the matching accuracy is averaged over these runs. The performance for attribute matching is shown in Table 6. The results demonstrate a markedly high accuracy for both MOSA and MOGA. It is worth noting that in most runs the Pareto fronts derived from MOSA and MOGA contain the gold standard matching (hence the high accuracy). It suggests a strategy to reduce the Pareto front in the matching problem by running MOSA or MOGA repeatedly after some times and only those "stable" points that appear more than certain proportion of the times are considered to be presented to decision makers.

[1] http://archive.ics.uci.edu/ml/datasets/Wine+Quality.

5 Discussion

5.1 Single Objective Versus Multi-Objective Approaches

In our previous work [18,19] we assumed the cluster matching is known prior to attribute matching. Then the attribute matching alone is simply a single objective problem. However, as we pointed out in the Introduction section, this is a gross simplification because attribute matching and cluster matching are intertwined and usually none can be known without the knowledge of the other. Therefore in this work, we focus on tackling this deadlock.

We argue that single objective approach is not applicable given the way we represent attributes and clusters. Specifically, we represent an attribute as an ordered tuple, $< v_1, v_2, \ldots, v_3 >$, where v_i is some statistics of the attribute in a cluster c_i of one dataset. Two attributes from different datasets can be compared only when we are able to arrange the tuples so that matching positions correspond to the same cluster. This assumes a certain kind of cluster matching. The vice versa is true for cluster matching in that we need some input on attribute matching. Essentially the problem at hand is to search in two permutation spaces, one for each matching problem, which naturally leads to our multi-objective approach. If one was to adopt a single objective approach, the two spaces would have to be concatenated and variables aggregated by some functions (e.g., weighted sum). We argue it might be flawed because there is no way to justify the ad hoc choice of such function. On the contrary, the multi-objective approach based on Pareto optimality circumvents the choice of aggregation, but focuses on obtaining a non-dominating set of solutions (the Pareto set). We demonstrate in our case studies one simple way to utilize the Pareto set by combining both objectives based on weights that are determined through pilot experiments. Note that applying weights before and after the optimization is fundamentally

different. The former carries more systematic risk of missing true optimum due to the arbitrary choice of weights, while the latter is just one way to post-process the Pareto set that is very likely to contain the optimum. In practice, the Pareto set itself can be well treated as the final product of the matching analysis. Note that we show the sizes of Pareto sets in Table 3 for the neuroscience test case, which are all reasonably small for examination to hand-pick best solutions.

5.2 Scalability Issues

The main computationally expensive part of the annealing process is the generation of new candidate solution phase (function G) in which an assignment problem is solved using the Hungarian method. The complexity of the Hungarian method is cubic and is already the most efficient algorithm for solving the assignment problem (e.g., a brute force algorithm has a factorial complexity). Although in our real-world case studies data are of relatively small dimensionality, we observe that there exist massive datasets that could render our method computationally impractical. For example, the ARCENE dataset [9] from the UCI machine learning repository contains mass-spectrometric output with 10,000 continuous input variables. ARCENE's task is to distinguish cancer versus normal patterns and the dataset is typically used as a benchmark for classification and feature selection algorithms. To match sets of attributes at this scale will definitely require more advanced adaptation of our metaheuristics search algorithm, such as approximation or partitioning of the search space to enable parallelism. On the other hand, we have shown in our case studies that our method boasts significant accuracy and the unique ability to distinguish attributes with similar statistics. For the ARCENE dataset, we create an artificial matching problem by first randomly selecting a subset of data with 150 attributes as the source, and then making a target dataset by injecting a small amount of noise to the source. We then run the simulated annealing algorithm to find both attribute and cluster matchings and achieved 132/150 accuracy for attribute matching and 4/5 accuracy for cluster matching. A baseline method that simply utilizes one single statistics for each attribute scores 95/150 accuracy. This shows that, without employing parallelism, our method provides for trade-off between accuracy and scalability.

6 Conclusion

In this paper, we present a novel approach to address challenges in the matching of heterogeneous datasets. In particular, we have proposed solutions to two matching problems (i.e., attribute matching and cluster matching) that arise in combining information from different results of scientific research. Our main contributions are

- A multi-objective approach to solve attribute matching and cluster matching problems simultaneously.
- A segmented statistical characterization representation to enable finer-grained modeling of internal distributions of attributes.
- An exploration of the widely used simulated annealing algorithm as the metaheuristics algorithm and a brief comparison with the evolutionary multi-objective algorithm in case studies.

The performance of this approach was demonstrated in a series of experiments using synthetic and realistic datasets that were designed to simulate heterogeneous data from different sources. In future work, we aim at improving the scalability of the proposed method through employing parallelism and approximation by relaxing some certain constraints. We also hope to incorporate more state-of-the-art research in the field of metaheuristics to improve the quality of Pareto optimal solutions and explore more ways to utilize these solutions.

Acknowledgments This work was supported by the NIH/NIBIB with Grant No. R01EB007684.

References

1. Bae E, Bailey J, Dong G (2010) A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. Data Min Knowl Discov 21:427–471
2. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J (1998) Modeling wine preferences by data mining from physicochemical properties. Decis Support Syst 47(4):547–553
3. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York
4. Dhamankar R, Lee Y, Doan A, Halevy A, Domingos P (2004) iMAP: discovering complex semantic matches between database schemas. In: Proceedings of the 2004 ACM SIGMOD international conference on management of data. ACM, New York
5. Dien J (2010) The ERP PCA Toolkit: an open source program for advanced statistical analysis of event-related potential data. J Neurosci Methods 187(1):138–145
6. Doan A, Domingos P, Levy AY (2000) Learning source description for data integration. In: WebDB (Informal Proceedings), pp 81–86
7. Fred AL, Jain AK (2003) Robust data clustering. In: IEEE Computer Society conference on computer vision and pattern recognition, vol 2, p 128
8. Frishkoff GA, Frank RM, Rong J, Dou D, Dien J, Halderman LK (2007) A framework to support automated classification and labeling of brain electromagnetic patterns. Comput Intell Neurosci (CIN): Special Issue EEG/MEG Anal Signal Process 7(3):1–13
9. Guyon I, Hur AB, Gunn S, Dror G (2004) Result analysis of the nips 2003 feature selection challenge. Adv Neural Inf Process Syst 17:545–552
10. Hamers L, Hemeryck Y, Herweyers G, Janssen M, Keters H, Rousseau R, Vanhoutte A (1989) Similarity measures in scientometric

research: the Jaccard index versus Salton's cosine formula. Inf Process Manag 25:315–318
11. Holland JH (1992) Adaptation in natural and artificial systems. MIT Press, Cambridge
12. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1987) Readings in computer vision: issues, problems, principles, and paradigms. In: Optimization by simulated annealing. Morgan Kaufmann, San Francisco, pp 606–615
13. Kong X, Shi X, Yu PS (2011) Multi-label collective classification. In: SDM'11, pp 618–629
14. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res Logistic Q 2:83–97
15. Larson JA, Navathe SB, Elmasri R (1989) A theory of attributed equivalence in databases with application to schema integration. IEEE Trans Softw Eng 15:449–463
16. Li WS, Clifton C (2000) Semint: a tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data Knowl Eng 33(1):49–84
17. Liu H, Dou D (2011) Breaking the deadlock: simultaneously discovering attribute matching and cluster matching with multi-objective simulated annealing. In: Proceedings of the international conference on ontologies, databases and application of semantics (ODBASE), pp 698–715
18. Liu H, Frishkoff G, Frank R, Dou D (2010) Ontology-based mining of brainwaves: a sequence similarity technique for mapping alternative descriptions of patterns in event related potentials (ERP) data. In: Proceedings of the 14th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), pp 43–54
19. Liu H, Frishkoff G, Frank R, Dou D (2012) Sharing and integration of cognitive neuroscience data: metric and pattern matching across heterogeneous ERP datasets. Neurocomputing 92:156–169
20. Namata GM, Kok S, Getoor L (2011) Collective graph identification. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '11. ACM, New York, pp 87–95
21. Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. VLDB J 10:2001
22. Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
23. Sheth AP, Larson JA, Cornelio A, Navathe SB (1988) A tool for integrating conceptual schemas and user views. In: Proceedings of the fourth international conference on data engineering. IEEE Computer Society, Washington, pp 176–183
24. Suman B (2003) Simulated annealing based multiobjective algorithm and their application for system reliability. Eng Optim 35:391–416
25. Suman B, Kumar P (2006) A survey of simulated annealing as a tool for single and multiobjective optimization. J Oper Res Soc 57:1143–1160
26. Wick ML, Rohanimanesh K, Schultz K, McCallum A (2008) A unified approach for schema matching, coreference and canonicalization. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '08. ACM, New York, pp 722–730
27. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. Springer, Berlin, pp 292–301