Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Improved Convergence of Iterative Ontology Alignment Using Block-Coordinate Descent

Uthayasanker Thayasivam and Prashant Doshi
THINC Lab, Dept. of Computer Science
University of Georgia
Athens, GA 30602, USA
{uthayasa,pdoshi}@cs.uga.edu

Abstract

A wealth of ontologies, many of which overlap in their scope, has made aligning ontologies an important problem for the semantic Web. Consequently, several algorithms now exist for automatically aligning ontologies, with mixed success in their performances. Crucial challenges for these algorithms involve scaling to large ontologies, and as applications of ontology alignment evolve, performing the alignment in a reasonable amount of time without compromising on the quality of the alignment. A class of alignment algorithms is iterative and often consumes more time than others while delivering solutions of high quality. We present a novel and general approach for speeding up the multivariable optimization process utilized by these algorithms. Specifically, we use the technique of block-coordinate descent in order to possibly improve the speed of convergence of the iterative alignment techniques. We integrate this approach into three well-known alignment systems and show that the enhanced systems generate similar or improved alignments in significantly less time on a comprehensive testbed of ontology pairs. This represents an important step toward making alignment techniques computationally more feasible.

Introduction

Ontology repositories such as the National Center for Biomedical Ontologies, which currently hosts 294 ontologies in the life sciences are indicative of the fact that there is now an abundance of ontologies and that these are finding important uses. This wealth of ontologies, many of which overlap in their scope, has made aligning ontologies an important problem for the semantic Web. Consequently, several algorithms (Jian et al. 2005; Li, Li, and Tang 2007; Jean-Mary, Shironoshita, and Kabuka 2009; Doshi, Kolli, and Thomas 2009; Wang and Xu 2009; Hanif and Aono 2009; Bock and Hettenhausen 2010) now exist for automatically aligning ontologies, with mixed success in their performances. Crucial challenges for these algorithms involve scaling to large ontologies and performing the alignment in a reasonable amount of time without compromising on the quality of the alignment. Although ontology alignment is traditionally perceived as an offline and one-time task, the second challenge is gaining importance. In particular, as Hughes and Ashpole (2004) note, continuously evolving ontologies and applications involving real-time ontology alignment such as semantic search and Web service composition stress the importance of computational complexity considerations. Additionally, established benchmarks such as the ontology alignment evaluation initiative (OAEI) (Shvaiko et al. 2011) recently began reporting the execution times of the participating alignment systems as well.

A class of algorithms that perform automated alignment is *iterative* in nature (Jian et al. 2005; Li, Li, and Tang 2007; Doshi, Kolli, and Thomas 2009; Wang and Xu 2009; Hanif and Aono 2009; Bock and Hettenhausen 2010). These algorithms repeatedly improve on the previous preliminary solution by optimizing a measure of the solution quality. Often, this is carried out as a guided search through the alignment space using techniques such as gradient descent or expectation-maximization. These algorithms run until convergence after which point the solution stays fixed but in practice, they are often terminated after an ad hoc number of iterations. Through repeated improvements, the computed alignment is usually of high quality but these approaches also consume more time in general than their non-iterative counterparts. While the focus on computational complexity has yielded ways of scaling the alignment algorithms to larger ontologies, such as through ontology partitioning (Hu, Zhao, and Qu 2006; Seddiqui and Aono 2009; Stoutenburg et al. 2010), there is a general absence of effort to speed up the ontology alignment process. We think that these considerations of space and time go hand in hand in the context of scalability.

In this paper, we introduce an approach for speeding up the convergence of iterative ontology alignment techniques. Objective functions that measure the quality of the solution are typically multidimensional. Instead of the traditional approach of modifying the values of a large number of variables in each iteration, we may decompose the problem into optimization subproblems in which the objective function is optimized with respect to a single or a small subset (block) of variables, while holding the other variables fixed. This approach of *block-coordinate descent* is theoretically shown to converge faster under considerably relaxed conditions on the objective function such as pseudoconvexity (and even the lack of it in certain cases) or the existence of optima in each variable (coordinate) block (Tseng

2001). While it forms a standard candidate tool for multidimensional optimization and has been applied in contexts such as image reconstruction (Pintér 2000; Fessler and Kim 2011) and channel capacity computation (Blahut 1972; Arimoto 1972), this paper presents its first application toward ontology alignment. Intuitively, the coordinate blocks in our application involve alignment variables between entities at specific heights in the ontology graph.

We evaluate this approach by integrating it into multiple ontology alignment systems. Although several iterative alignment techniques have been proposed, we selected Falcon-AO (Jian et al. 2005), MapPSO (Bock and Hettenhausen 2010) and Optima (Doshi, Kolli, and Thomas 2009) as representative algorithms. Not only have these established tools participated in previous OAEI benchmarks and performed satisfactorily, their implementations and source code are freely accessible as well. Using a comprehensive testbed of several ontology pairs – some of which are very large – spanning multiple domains, we show a significant reduction in the execution times of the alignment processes, indicating faster convergences. This enables the application of these techniques to more ontology pairs in a given amount of time, or to more subsets in large ontology partitions.

Background

We briefly introduce the ontology alignment problem and among the different alignment techniques that have been proposed, we focus on three, which are iterative and recognized.

Iterative Ontology Alignment

The ontology alignment problem is to find a set of correspondences between two ontologies O_1 and O_2. Because ontologies may be modeled as labeled graphs (though with some possible loss of information), the problem is often cast as a matching problem between such graphs. An ontology graph, O, is defined as, $O = \langle V, E, L \rangle$, where V is the set of labeled vertices representing the entities, E is the set of edges representing the relations, which is a set of ordered 2-subsets of V, and L is a mapping from each edge to its label. A correspondence, $m_{a\alpha}$, between two entities, $x_a \in O_1$ and $y_\alpha \in O_2$ consists of the relation, $r \in \{=, \subseteq, \supseteq\}$, and confidence, $c \in \mathbb{R}$. However, the alignment systems that we use focus on the possible presence of $=$ relation (also called, *equivalentClass*) between entities only. In this case, an alignment may be represented as a $|V_1| \times |V_2|$-dimensional matrix that represents the correspondence between the two ontologies, $O_1 = \langle V_1, E_1, L_1 \rangle$ and $O_2 = \langle V_2, E_2, L_2 \rangle$:

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1|V_m|} \\ m_{21} & m_{22} & \cdots & m_{2|V_m|} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ m_{|V_d|1} & m_{|V_d|2} & \cdots & m_{|V_d||V_m|} \end{bmatrix}$$

Each assignment variable, $m_{a\alpha}$, in M is the confidence of the correspondence between entities, $x_a \in V_1$ and $y_\alpha \in V_2$. Consequently, M could be a real-valued matrix commonly known as the similarity matrix between the two ontologies. However, in some systems including in two that we utilize, the confidence is binary with 1 indicating a correspondence, otherwise 0, due to which the match matrix M becomes a binary matrix representing the alignment.

A class of alignment algorithms is iterative in nature. These utilize a seed matrix, M^0, either input by the user or generated automatically. Beginning with the seed, the match matrix is iteratively improved until it converges. Two types of iterative techniques are predominant. The first type of iterative algorithms improve the real-valued similarity matrix from the previous iteration, M^{i-1}, by directly updating it as shown below:

$$M^i = U(M^{i-1}) \tag{1}$$

where U is a function that updates the similarities.

The other type searches over the space of match matrices, denoted as $\mathcal{M}$, in order to find the alignment that optimizes an objective function, which gives a measure of the quality of the alignment, in the context of the alignment from the previous iteration. This approach is appropriate when the search space is bounded such as when the match matrix is binary, although with a cardinality of $2^{|V_1||V_2|}$ this space could get very large. Formally,

$$M^i_* = \arg\max_{M \in \mathcal{M}} Q(M, M^{i-1}_*) \tag{2}$$

where, M^i_* is the alignment that optimizes the Q function in iteration i.

Equations 1 and 2 help solve a multidimensional optimization problem iteratively with $m_{a\alpha}$ in M as the variables. Next, we briefly review three ontology alignment algorithms that optimize iteratively. Their selection is based on their accessibility and competitive performance on previous OAEI benchmarks, and is meant to be representative of iteration-based alignment algorithms.

Falcon-AO An important component of the Falcon-AO ontology alignment system (Jian et al. 2005) is its iterative graph matching called GMO (Hu et al. 2005). GMO measures the structural similarity between the ontologies that are modeled as bipartite graphs (Hayes and Gutierrez 2004). Matrix M in GMO is real-valued and this similarity matrix is iteratively updated (Eq. 1) by updating each similarity with the average of its neighborhood similarities until M stops changing. Equation 1 manifests in Falcon-AO as a series of matrix operations:

$$M^i = G_1 M^{i-1} G_2^T + G_1^T M^{i-1} G_2 \tag{3}$$

Here, G_1 and G_2 are the adjacency matrices of the bipartite graph models of the two ontologies O_1 and O_2, respectively. In the first term of the summation, the outbound neighborhood of entities in O_1 and O_2 is considered, while the second term considers the inbound neighborhood.

MapPSO The MapPSO alignment system (Bock and Hettenhausen 2010) utilizes discrete particle swarms to perform the optimization. Each of K particles in the swarm represents a valid candidate alignment, which is updated iteratively. In each iteration, given the particle(s) representing the

best alignment(s) in the swarm, alignments in other particles are adjusted as influenced by the best particle.

Equation 2 manifests in MapPSO as a two-step process consisting of retaining the best particle(s) (alignment(s)) and replacing all others with improved ones influenced by the best alignment in the previous iteration. The measure of the quality of an alignment in the k^{th} particle is determined by the mean of the measures of its correspondences as shown below:

$$Q(M_k^i) = \frac{\sum_{a=1}^{|V_1|}\sum_{\alpha=1}^{|V_2|} m_{a\alpha} \times f(x_a, y_\alpha)}{|V_1||V_2|} \quad (4)$$

where $m_{a\alpha}$ is a correspondence in M_k^i and f represents a weighted combination of a number of syntactic and possibly semantic similarity measures between the entities in the two ontologies.

Improved particles are generated by keeping aside a random number of best correspondences according to f in the alignment in the particle, and replacing others based on the correspondences in the previous best particle. MapPSO's architecture is capable of exploiting parallelism and adapts naturally to parallel architectures.

Optima Optima (Doshi, Kolli, and Thomas 2009) formulates ontology alignment as a maximum likelihood problem, and searches for the match matrix, M_*, which gives the maximum conditional probability of observing the ontology O_1, given the other ontology, O_2, under the match matrix M_*.

Optima employs expectation-maximization to solve this optimization problem in which it iteratively evaluates the expected log likelihood of each candidate alignment and picks the one which maximize it. It implements Eq. 2 as a two-step process of computing expectation followed by maximization, which is iterated until convergence.

The expectation step consists of evaluating the expected log likelihood of the candidate alignment given the previous iteration's alignment:

$$Q(M^i|M^{i-1}) = \sum_{a=1}^{|V_1|}\sum_{\alpha=1}^{|V_2|} Pr(y_\alpha|x_a, M^{i-1}) \times logPr(x_a|y_\alpha, M^i)\pi_\alpha^i \quad (5)$$

where x_a and y_α are the entities of ontologies O_1 and O_2, respectively, and π_α^i is the prior probability of y_α. $Pr(x_a|y_\alpha, M^i)$ is the probability that node x_a is in correspondence with node y_α given the match matrix M^i. The prior probability is computed using the following equation,

$$\pi_\alpha^i = \frac{1}{|V_1|}\sum_{a=1}^{|V_1|} Pr(y_\alpha|x_a, M^{i-1})$$

The generalized maximization step involves finding a match matrix, M_*^i, that improves on the previous one:

$$M_*^i = M^i \in \mathcal{M} :\ Q(M^i|M_*^{i-1}) \geq Q(M_*^{i-1}|M_*^{i-1}) \quad (6)$$

Block-Coordinate Descent

Block-coordinate descent (BCD) (Tseng 2001) is an established technique to gain faster convergence in the context of large-scale N-dimensional optimization problems. In this technique, the variables, referred to as coordinates, are partitioned into C blocks and, within each iteration, the objective function, Q, is optimized with respect to one of the coordinate blocks while the other coordinates are held fixed. In order to converge using BCD, we must meet the following cyclic rule, which ensures that each coordinate block is chosen sufficiently often (Tseng 2001). Let S denote a block of coordinates, which is indexed by a non-empty subset of $\{1,2,\ldots,N\}$. We may define a set of such blocks as, $B = \{S_0, S_1, \ldots, S_C\}$, which is a set of subsets each representing a coordinate block with the constraint that, $S_1 \cup S_2 \cup \ldots \cup S_C = \{1,2,\ldots,N\}$. Then,

Cyclic rule: There exists a constant, $T \leq N$, such that every block, S, is chosen at least once between the i^{th} iteration and the $(i+T-1)^{th}$ iteration, for all i.

A simple way to meet this rule is by sequentially iterating through each block although we must continue iterating until each block converges.

Integrating BCD into Iterative Alignment

Ontology alignment techniques, such as those mentioned previously, optimize over a multidimensional space. As the objective functions are often complex and non-differentiable, numerical iterative techniques are appropriate but these tend to progress slowly. In this context, we may speed up the convergence using BCD as we describe below.

Approach

In order to integrate BCD into the iterations, the match matrix, M, must be first suitably partitioned into blocks. Though a matrix may be partitioned using one of several ways, we adopt an approach that is intuitive in the context of ontology alignment.

An important heuristic, which has proved highly successful in both ontology and schema alignment, matches parent entities in two ontologies if their respective child entities were previously matched. This motivates grouping together those variables, $m_{a\alpha}$ in M, into a coordinate block such that the x_a participating in the correspondence belong to the same height leading to a partition of M. The height of an ontology node is the length of the shortest path from a leaf node. Let the partition of M into the coordinate blocks be $\{M_{S_0}, M_{S_1}, \ldots, M_{S_C}\}$, where C is the height of the ontology O_1. Thus, each block is a submatrix with as many rows as the number of entities of O_1 at a height and number of columns equal to the number of all entities in O_2. For example, the correspondences between the leaf entities of O_1 and all entities of O_2 will form the block, M_{S_0}. In the context of a bipartite graph model as utilized by Falcon-AO and Optima, which represents properties in an ontology as vertices as well and are therefore part of M, these would be included in the coordinate blocks.

Iterative ontology alignment integrated with BCD optimizes with respect to a single block, M_{S_c}, at an iteration while keeping the remaining blocks fixed. In order to meet the cyclic rule, we choose a block, M_{S_c}, at iterations, $i = c + qC$ where $q \in \{0,1,2,\ldots\}$. We point out that BCD is

applicable to both types of iterative alignment techniques as outlined in the previous section. Alignment algorithms which updated the similarity matrix iteratively as in Eq. 1 will now update only the current block of interest, M_{S_c}, and the remaining blocks are carried forward as is, as shown below:

$$\begin{aligned} M^i_{S_c} &= U_S(M^{i-1}_{S_c}) \\ M^i_{\tilde{S}_c} &= M^{i-1}_{\tilde{S}_c} \end{aligned} \tag{7}$$

for all $\tilde{S}_c$ in the complement of S_c from B. Note that $M^i_{S_c}$ combined with $M^i_{\tilde{S}_c}$ for all $\tilde{S}_c$ forms M^i. Update function, U_S, modifies U in Eq. 1 to update just a block of the coordinates.

Analogously, the iterative alignment techniques which search for the candidate alignment that maximizes the objective function as in Eq. 2, will now choose a block, M_{S_c}, at each iteration. They will search over the *reduced search space* pertaining to the subset of all variables as selected in M_{S_c}, for the best candidate coordinate block. Formally,

$$\begin{aligned} M^i_{S_c,*} &= \underset{M_{S_c} \in \mathcal{M}_{S_c}}{\arg\max}\, Q_S\left(M_{S_c}, M^{i-1}_*\right) \\ M^i_{\tilde{S}_c,*} &= M^{i-1}_{\tilde{S}_c,*} \end{aligned} \tag{8}$$

where $\mathcal{M}_{S_c}$ is the space of alignments limited to block, S_c. The original objective function, Q, is modified to Q_S such that it provides a measure of the quality of the block, M_{S_c}, only given the previous best match matrix. Note that the previous iteration's matrix, M^{i-1}_*, contains the best block that was of interest in that iteration.

Given the general modifications above brought about by BCD, we describe how these manifest in the three iterative alignment systems that form our focus.

BCD Enhanced Falcon-AO

We enhance Falcon-AO by integrating BCD with its iterative GMO component. We begin by partitioning the similarity matrix used by GMO into C blocks based on the height of the entities in O_1 that are part of the correspondences, as mentioned previously. GMO is then modified so that at each iteration, a block of the similarity matrix is updated while the other blocks remain unchanged. If block, S_c, is updated at iteration, i, then Eq. 3 becomes:

$$\begin{aligned} M^i_{S_c} &= G_{1,S_c} M^{i-1} G_2^T + G^T_{1,S_c} M^{i-1} G_2 \\ M^i_{\tilde{S}_c} &= M^{i-1}_{\tilde{S}_c} \end{aligned} \tag{9}$$

Here, G_{1,S_c} focuses on that portion of the adjacency matrix of O_1 that corresponds to the outbound neighborhood of entities participating in correspondences of block S_c, while G^T_{1,S_c} focuses on the inbound neighborhood of entities in S_c. Adjacency matrix, G_2, is utilized as before. The outcome of the matrix operations is a similarity matrix, with as many rows as the variables in S_c and columns corresponding to all the entities in O_2. The complete similarity matrix is obtained at iteration, i, by carrying forward the remaining blocks unchanged, which is then utilized in the next iteration.

BCD Enhanced MapPSO

We may integrate BCD into MapPSO by ordering the particles in a swarm based on a measure of the quality of a coordinate block, S_c, in each particle in an iteration. Equation 4 is modified to measure the quality of the correspondences in just the coordinate block, S_c, in the k^{th} particle by taking the average.

$$Q_S(M^i_k) = \frac{\sum_{a=1}^{|V_{1,c}|} \sum_{\alpha=1}^{|V_2|} m_{a\alpha} \times f(x_a, y_\alpha)}{|V_{1,c}||V_2|} \tag{10}$$

where $V_{1,c}$ denotes the set of entities of ontology, O_1, of identical height participating in the correspondences included in block, S_c. As before, we retain the best particle(s) based on this measure and improve on coordinate block alignment, M^i_{k,S_c}, in the remaining particles using the best particle in the previous iteration while keeping the remaining coordinates unchanged.

BCD Enhanced Optima

As we mentioned previously, Optima utilizes generalized expectation-maximization to iteratively improve the likelihood of candidate alignments. Jeffery and Alfred (1994) discuss a BCD inspired expectation-maximization scheme and call it as space alternating generalized expectation-maximization (SAGE). Intuitively, SAGE maximizes the expected log likelihood of a block of coordinates, thereby limiting the hidden space, instead of maximizing the likelihood of the complete alignment. In each iteration, Optima enhanced using SAGE chooses a block of the match matrix, $M^i_{S_c}$, and its expected log likelihood is estimated. As in previous techniques, we choose the blocks in a sequential manner such that all the blocks are iterated in order.

Equation 5 is modified to estimate the expected log likelihood of the block of a candidate alignment as:

$$\begin{aligned} Q_S(M^i_{S_c}|M^{i-1}) = \sum_{a=1}^{|V_{1,c}|} \sum_{\alpha=1}^{|V_2|} & Pr(y_\alpha|x_a, M^{i-1}) \\ & \times logPr(x_a|y_\alpha, M^i_{S_c})\ \pi^i_{\alpha,c} \end{aligned} \tag{11}$$

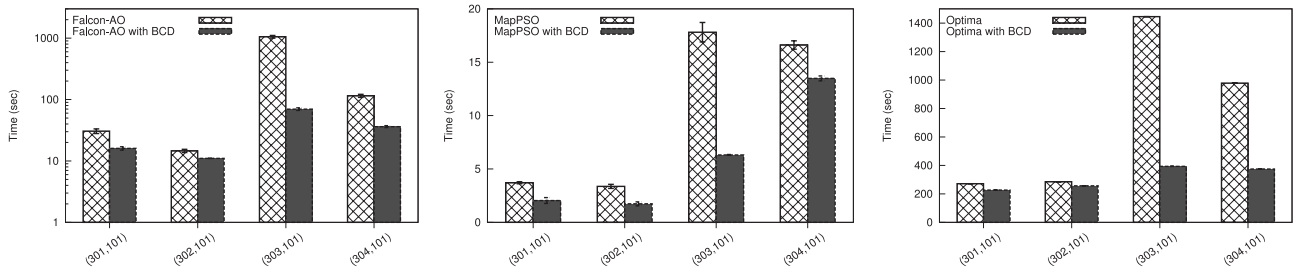
Recall that $V_{1,c}$ denotes the set of entities of ontology, O_1, participating in the correspondences included in S_c. Notice that the prior probability, $\pi^i_{\alpha,c}$, is modified as well to utilize just $V_{1,c}$ in its calculations.

The generalized maximization step now involves finding a match matrix block, $M^i_{S_c,*}$, that improves on the previous one:

$$\begin{aligned} M^i_{S_c,*} = \quad & M^i_{S_c} \in \mathcal{M}_{S_c} : \\ & Q_S(M^i_{S_c,*}|M^{i-1}_*) \geq Q_S(M^{i-1}_{S_c,*}|M^{i-1}_*) \end{aligned} \tag{12}$$

Here, $M^{i-1}_{S_c,*}$ is a part of M^{i-1}_*.

At iteration i, the best alignment matrix, M^i_*, is formed by combining the block matrix, $M^i_{S_c,*}$, which improves the Q_S function as defined in Eq. 12 with the remaining from the previous iteration, $M^{i-1}_{\tilde{S}_c,*}$, unchanged.



(a) (b) (c)

Figure 1: Average total execution time consumed by the three iterative algorithms, (a) Falcon-AO, (b) MapPSO, and (c) Optima, in their original form and with BCD when aligning the 4 ontology pairs of the *bibliography* domain. Note that the time axis of (a) is in log scale. While the overall differences in run time are statistically significant, we point out an order of magnitude reduction for the (303,101) pair in (a). For a majority of the pairs, the algorithms converged in a lesser number of iterations as well.

Experiments

We empirically analyze the improvements in the speed of convergence and the associated tradeoffs, if any, in the final alignment, due to the integration of BCD in the iterative alignment techniques. We used a comprehensive testbed of several ontology pairs – some of which are very large – spanning multiple domains. We used ontology pairs from the OAEI competition in its recent version, 2011, as the testbed for our evaluation. Among the OAEI tracks, we focus on the test cases that involve real-world ontologies for which the reference (true) alignment was provided by OAEI. These ontologies were either acquired from the Web or created independently of each other and based on real-world resources. This includes all ontology pairs in the 300 range of the benchmark, which relate to *bibliography*, expressive ontologies in the *conference* track all of which structure knowledge related to conference organization, and the *anatomy* track, which consists of large ontologies from life sciences, describing anatomy of adult mouse and human. We list the ontologies participating in our evaluation in Table 1 and provide an indication of their sizes.

In our experiments, we aligned ontology pairs using all three iterative alignment systems, in their original forms and with BCD using the same seed alignment, M^0. The iterations were run until the algorithm converged and we measured the total execution time, final recall and precision, and the number of iterations performed until convergence. We stopped the execution if the system did not converge for an ontology pair within 5 hours. Note that this is a longer time limit compared to the OAEI 2011 benchmark limit of 2 hours. We averaged results of 5 runs on every ontology pair using both the original and the BCD enhanced version of each system. Because of the large number of total runs, we ran the tests on three different computing platforms while ensuring comparability. Two of these were Red Hat machines with Intel Xeon Core 2, processor speed of about 3 GHz with 4GB of memory, while the third was a Windows 7 machine with Intel Core i7, 1.6 GHz processor and 4GB of memory.

The 4 ontology pairs in *bibliography* domain are obtained from the 300 series of OAEI's benchmark track, which consists of real-world ontologies describing bibliographic references. We show the average total execution time consumed by each algorithm until convergence in its original form and with BCD in Fig. 1. While the introduction of BCD significantly reduces the amount of total execution time consumed by all three iterative techniques (Student's paired t-test, $p \ll 0.01$), the reduction is greater than an order of magnitude for ontology pair (303,101) in the context of Falcon-AO. Nevertheless, the final recall and precision of the resulting alignment remained unchanged for Optima. In the case of Falcon-AO, the integration of BCD caused the precision for ontology pairs (302,101) and (303,101) to improve by 2% from 63.04% to 65.9% and from 41.23% to 42.1%, respectively, while recall remained the same at 61.7% and 83.3%, respectively. The precision and recall for all other pairs remain unchanged. Enhancing MapPSO, which has a random component, with BCD improved the precision averaged across the

Ontology	Named Classes	Properties
Bibliography Domain		
101	37	70
301	16	40
302	14	30
303	57	72
304	41	49
Conference Domain		
ekaw	74	33
sigkdd	49	28
iasted	150	41
cmt	36	59
edas	104	50
confOf	38	36
conference	60	64
Life Science Domain		
mouse anatomy	2744	2
human anatomy	3304	3

Table 1: Ontologies from OAEI 2011 used in our evaluation and the number of named classes and properties in each. Notice that our evaluation includes large ontologies from different domains.

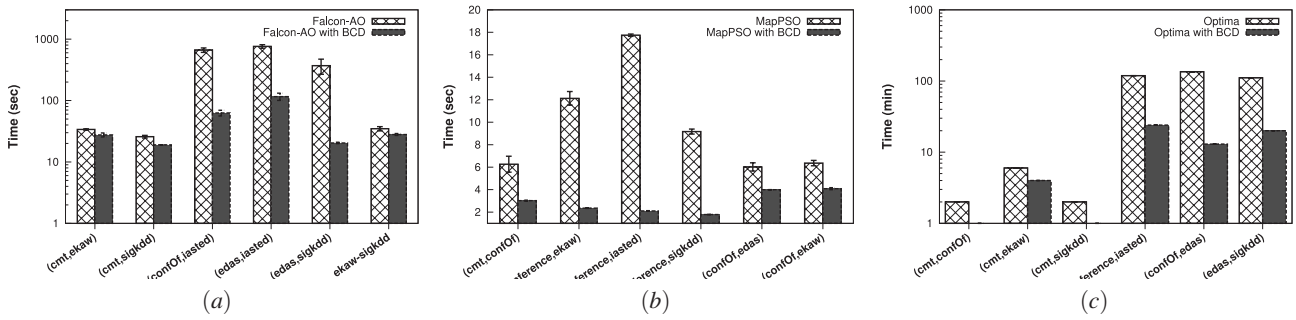


Figure 2: Average total execution time consumed by, (a) Falcon-AO, (b) MapPSO, and (c) Optima in their original form and with BCD, for 6 of the 21 ontology pairs from *conference* domain. Although we ran the algorithms for all the pairs, we selected ontology pairs which exhibited the highest and lowest differences in average execution times. Note that the time axis of (a) and (c) are in log scale. Notice the improvements in execution time for the larger pairs.

pairs and runs from 37% to 88%, while the recall remained steady at about 37%.

The ontologies in the *conference* domain vary widely in their size and structure. As shown in Fig. 2, the introduction of BCD to the three iterative techniques clearly improves their speeds of convergence and the differences are significant (Student's paired t-test, $p \ll 0.01$). In particular, we observed an order of magnitude reduction in time for aligning relatively larger ontologies such as *iasted* and *edas*. For example, pairs *(confOf,iasted)* in Falcon-AO, *(conference,iasted)* in MapPSO and *(confOf, edas)* in Optima showed such reductions. Importantly, BCD enhanced Optima successfully completed aligning all the ontology pairs in the *conference* domain while in its original form it was unable to completely align the larger ontology pairs within the cutoff of 5 hours. Analogous to the *bibliography* domain, BCD enhanced *Optima* did not show any change in precision and recall of the final alignment. However, Falcon-AO exhibited slight improvement in the average precision from about 13% to 14% while keeping the average recall unchanged at 73%. MapPSO with BCD resulted in a significant improvement in final precision from 9% to 43% on average although the difference in recall was not significant.

The very large *anatomy* ontologies for mouse and human were not successfully aligned by either MapPSO or Optima despite the use of BCD. However, BCD drastically reduced the average total execution time for aligning this ontology pair when using Falcon-AO from 162 minutes to 85 minutes. Furthermore, the alignment generated by Falcon-AO with BCD gained in precision from 74% to 76% while keeping the recall unchanged.

In summary, the introduction of BCD led to significant reductions in convergence time for all three iterative algorithms on multiple ontology pairs. Furthermore, the quality of the final alignments also improved in some cases while remaining unchanged in others.

Discussion

While techniques for scaling automated alignment to large ontologies have been previously proposed, there is a general absence of effort to speed up the alignment process. We presented a novel approach based on BCD to increase the speed of convergence of iterative alignment algorithms with no observed adverse effect on the final alignments. We also demonstrated this technique in the context of three different alignment systems and evaluated its impact on both the total time of execution and the final alignment's precision and recall. We reported significant reductions in the total execution times of the algorithms enhanced using BCD. These reductions were most noticeable for larger ontology pairs. Often the algorithms converged in a lesser number of iterations. Simultaneously, the integration of BCD improved the precision of the alignments generated by some of the algorithms while retaining the recall.

The capability to converge quickly allows an iterative alignment algorithm to run until convergence, in contrast to the common practice of terminating the alignment process after an arbitrary number of iterations. As predefining a common bound for the number of iterations is difficult, speeding up the convergence becomes vital.

Our particular approach toward creating blocks limits their number to the height of an ontology participating in the alignment. If the ontologies are shallow, the number of blocks created will be less, which may impact the improvement in convergence time brought about by BCD. On the other hand, tall ontologies may lead to an excessive number of blocks thereby magnifying the possibility of converging to local optima. Other partitioning techniques may not be ruled out, though we find the employed partitioning technique to be intuitive and effective. While we may partition M both row- and column-wise, this could lead to overpartitioning and impact the quality of the alignment. Furthermore, switching row and column ontologies in M may affect the final results though our choice of which ontology to partition was arbitrary.

We believe that the observed increase in precision of the alignment due to BCD is because of the optimized mappings found for the previous coordinate block, which influence the selection of the mappings for the current coordinate block. Additionally, the randomly generated mappings in MapPSO are limited to the block instead of the whole ontology, due to which the search becomes more guided. Given that on inte-

grating BCD the iterative algorithms produced better quality alignments, we infer that the original algorithms were converging to local optima, instead of the global optima, and that using BCD has likely resulted in convergence to (better) local optima as well. This is a significant insight because it uncovers the presence of local optima in the alignment space of these algorithms. This may limit the efficacy of iterative alignment techniques.

Acknowledgement

This research is supported in part by grant number R01HL087795 from the National Heart, Lung, And Blood Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, And Blood Institute or the National Institutes of Health.

References

Arimoto, S. 1972. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory* 18(1):14–20.

Blahut, R. E. 1972. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory* 18:460–473.

Bock, J., and Hettenhausen, J. 2010. Discrete particle swarm optimisation for ontology alignment. *Information Sciences* 1–22.

Doshi, P.; Kolli, R.; and Thomas, C. 2009. Inexact matching of ontology graphs using expectation-maximization. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(2):90–106.

Fessler, J. A., and Hero, A. O. 1994. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on Signal Processing* 42:2664–2677.

Fessler, J. A., and Kim, D. 2011. Axial block coordinate descent (abcd) algorithm for X-ray CT image reconstruction. In *Proceedings of Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, 262–265.

Hanif, M. S., and Aono, M. 2009. Anchor-flood: results for OAEI 2009. In *Proceedings of the Workshop on Ontology Matching at 8th International Semantic Web Conference*, 127–134.

Hayes, J., and Gutierrez, C. 2004. Bipartite graphs as intermediate model for RDF. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 47–61.

Hu, W.; Jian, N.; Qu, Y.; and Wang, Y. 2005. GMO: A graph matching for ontologies. In *K-Cap Workshop on Integrating Ontologies*, 43–50.

Hu, W.; Zhao, Y.; and Qu, Y. 2006. Partition-based block matching of large class hierarchies. In *Proceedings of the 1st Asian Semantic Web Conference (ASWC)*, 72–83.

Hughes, T. C., and Ashpole, B. C. 2004. The semantics of ontology alignment. In *Information Interpretation and Integration Conference (I3CON)*.

Jean-Mary, Y. R.; Shironoshita, E. P.; and Kabuka, M. R. 2009. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(3):235–251.

Jian, N.; Hu, W.; Cheng, G.; and Qu, Y. 2005. Falcon-AO: Aligning ontologies with Falcon. In *K-Cap Workshop on Integrating Ontologies*, 87–93.

Li, Y.; Li, J.; and Tang, J. 2007. RiMOM: Ontology alignment with strategy selection. In *Proceedings of the 6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, 51–52.

Pintér, J. D. 2000. Yair censor and stavros a. zenios, parallel optimization – theory, algorithms, and applications. *Journal of Global Optimization* 16:107–108.

Seddiqui, M. H., and Aono, M. 2009. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web* 7:344–356.

Shvaiko, P.; Euzenat, J.; Heath, T.; Quix, C.; Mao, M.; and Cruz, I. F., eds. 2011. *Proceedings of the 6th International Workshop on Ontology Matching*, volume 814 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Stoutenburg, S. K.; Kalita, J.; Ewing, K.; and Hines, L. M. 2010. Scaling alignment of large ontologies. *International Journal of Bioinformatics Research and Applications* 6:384–401.

Tseng, P. 2001. Convergence of block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications* 109:475–494.

Wang, P., and Xu, B. 2009. Lily: Ontology alignment results for OAEI 2008. In *Proceedings of the Workshop on Ontology Matching at 7th International Semantic Web Conference (ISWC)*.