International Journal on Artificial Intelligence Tools
c World Scientific Publishing Company

A Graph Matching Algorithm and Its Application to Conceptual System Translation

Ying Feng
Computer Science Department, Indiana University, Bloomington, IN, 47405, USA
yingfeng@cs.indiana.edu

Robert L. Goldstone
Psychology Department, Indiana University, Bloomington, IN, 47405, USA
rgoldsto@indiana.edu

Vladimir Menkov
Aqsaqal Enterprises, Penticton, BC, V2A 8R3, Canada
vmenkov@cs.indiana.edu

ABSURDIST II, an extension to ABSURDIST, is an algorithm using attributed graph matching to find translations between conceptual systems. It uses information about the internal structure of systems by itself, or in combination with external information about concept similarities across systems. It supports systems with multiple types of weighted or unweighted, directed or undirected relations between concepts. The algorithm exploits graph sparsity to improve computational efficiency. We present the results of experiments with a number of conceptual systems, including artificially constructed random graphs with introduced distortions.

Keywords: Conceptual systems, graph matching, ontology translation, knowledge representation

1. Introduction

The problem of translating between conceptual systems is of substantial interest in cognitive science. It involves the following question: given two conceptual systems A and B, each one consisting of some concepts and relations between them, how can correspondences across the systems be established? A general system for translating between conceptual systems would be valuable in many domains. We would like a method for translating between two individuals who have been acculturated with different languages or terminologies, between two time-slices of the same individual to track their cognitive development, between two scientific communities committed to fundamentally different theoretical ontologies to determine potential continuities across historical change, between related databases using different XML representations, and between different knowledge structures within the same individual to

identify useful analogies for induction.

There are two major strategies for establishing corresponding concepts across conceptual systems. The first approach of External Grounding establishes correspondences by finding common external referents for conceptual elements across two systems. The second approach uses purely internal information within the systems to establish correspondences. John's and Jane's **Cat** concepts can potentially be placed into correspondence with each other because they play similar roles within their respective conceptual networks, including matching **is-a** relations to **Animal**, **eats** relations to **Friskies**, and **has-a** relations to **Paw**. Some argue that this account is hopelessly circular.[1] Jane's and John's **Cat** concept can only be placed into correspondence if one already knows that their **Friskies** concepts match, and this is just as much in need of explanation as how to match **Cat** concepts. However, other researchers[2] have argued that the circularity, while certainly present, is not hopeless. With proper algorithms, cross-system correspondences can mutually and simultaneously constrain one another.[3,4] The current work explores a method for integrating internal and external determinants of conceptual translation.

A conceptual system can be formalized as a *directed weighted labeled graph* (discussed below), also known as an *attributed graph*. Thus, matching two conceptual systems can be formalized as finding a match between two attributed graphs and evaluating the quality of this match.

2. Related work

Graph matching is a mature computer science research area. While the original research focus was on finding exact isomorphisms between two graphs, a substantial amount of work has been done on finding approximate, or "error-correcting" isomorphisms as well. Those latter efforts are directed at finding, for two graphs G_1 and G_2, an isomorphism $s(\ )$ that minimizes the "distance" between the graph $S = s(G_1)$ and the graph G_2. The definition of distance is application specific. It can be based, for example, on the number of editing operations needed to convert S into G_2, or on the Euclidean norm of the difference between the matrices describing S and G_2.

A detailed bibliography on the approximate isomorphism problem can be found in B. Messmer's dissertation.[5] There have been three main lines of research. First, the classic structure matching algorithms first developed for finding exact isomorphism can be extended to the case of approximate matching. Messmer's own work is in this tradition. Second, one can approach the approximate isomorphism problem as the problem of minimizing a certain error function on the space of correspondence matrices subject to structural constraints, and solve the problem using the full arsenal of optimization methods.[6,7,8] Third, a number of neural-net and related iterative approaches have been presented.[9,10,11]

This paper continues work on the simple approach called ABSURDIST (Aligning Between Systems Using Relations Derived Inside Systems Themselves).[3] It was

originally inspired by constraint propagation neural networks for consistent visual interpretation or analogical reasoning, but, as we show in this paper, can also be interpreted in optimization terms. We begin by describing graph representations for conceptual systems. Then we describe ABSURDIST II, an extended and improved translation algorithm based on ABSURDIST. Experimental results for matching concept systems using only internal, within-system information are given, followed by the experiments showing interactions between internal and external sources of information when both are available.

3. Graph Representation of Conceptual Systems

The ABSURDIST algorithm is based on matching the concepts of two conceptual systems based on the structural similarity of the two systems. To be able to express the similarity in quantitative terms, we need to develop a framework for measuring "edge similarity", i.e. for comparing the totality of relations existing between concepts A_q and A_r in system A with that between concepts B_x and B_y in system B.

A conceptual system with n concepts $\{A_1, \ldots, A_n\}$ can be represented as an $n \quad n$ matrix $G \in (\{0\} \cup S)^{n \quad n}$. In this matrix each element $g_{ij} \in (\{0\} \cup S)$ describes the totality of the relations existing between concepts A_i and A_j. The set S is chosen to be suitable for representing all possible non-empty combinations of relations that may exist between a pair of nodes. (The value 0 is used to describe the absence of any relations between the two concepts).

Alternatively, the conceptual system can be represented as a directed graph with n concepts, with each concept represented by a node. Concepts are connected by edges carrying *labels* from the same set S. Its value is the same as the value of the matrix element g_{ij}, and describes the totality of the relations between concepts A_i and A_j.

Depending on the complexity of the conceptual system, various types of graph representations can be chosen. First, consider a system with only binary valued relations between the concepts. Such a system can be described by an undirected or directed graph with no explicitly stored labels associated with edges.

In more complex systems, the relation may have some kind of strength or weight associated with it. For example, the **Cooccurrence** relation in a conceptual system describing the vocabulary of a corpus may have a weight corresponding to the degree of cooccurrence of the terms it links. In some systems based on belief credence, a relation may have a probability- or likelihood-related weight. For such a system we would want to use a graph where real-valued labels are associated with the edges.

Finally, in an even more general case when M possibly weighted relation types R_1, $R_2, \ldots, R_M$ exist in the system, the system can be represented by a directed graph whose labels are vectors from the M-dimensional space $S = \mathrm{R}^M$. We use the notation $\vec{a}_{qr}$ for the label associated with the edge going from A_q to A_r in the graph. The i-th component of this vector, a^i_{qr}, is the weight of the relation R_i on

this edge. If no edge goes from A_q to A_r, then the value of $\vec{a}_{qr}$ will be a zero vector, $\vec{a}_{qr} = \vec{0}$.

For convenience, our implementation of ABSURDIST II assumes that all relations are directed. If the conceptual system involves an undirected (symmetric) relation, such as **Similar-to**, existing between concepts A_1 and A_2, then we simply store it twice, as **Similar-to**(A_1, A_2) and **Similar-to**(A_2, A_1). We also assume that all relation weights are in the $[0; 1]$ range, which means that all edge labels in fact belong to $[0; 1]^M$.

The original ABSURDIST algorithm uses the concept of "similarity of psychological distances" $S(D(A_q, A_r), D(B_x, B_y))$ which measures how similar the relation between the concepts A_q and A_r in system A is to the relation between the concepts B_x and B_y in system B. To extend it to directed graphs with labels from $[0; 1]^M$, we first define the *directed edge similarity function* $S_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy})$ as

$$S_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}) = 1 \quad D_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}). \tag{1}$$

The *directed edge difference function* $D_{\mathrm{d}}(\vec{a}, \vec{b})$ used above is simply the normalized 1-norm of the vector difference of the edge labels, viz.

$$D_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}) = \|\vec{a}_{qr} \quad \vec{b}_{xy}\|_1 = \frac{1}{M} \sum_{i=1}^{M} |a^i_{qr} \quad b^i_{xy}|. \tag{2}$$

It is possible, of course, to define the edge difference using the 2-norm or the ∞-norm of the vector difference, instead of the 1-norm.

To compute the excitation matrix in ABSURDIST II (Section 4.1, below) we will use, in place of the "similarity of psychological distances" used in the original ABSURDIST,[3] the *undirected, or symmetrized, edge similarity*:

$$S(\vec{a}_{qr}, \vec{b}_{xy}) = S(\vec{a}_{rq}, \vec{b}_{yx}) = 1 \quad D(\vec{a}_{qr}, \vec{b}_{xy}) \tag{3}$$

with $D(\vec{a}_{qr}, \vec{b}_{xy}) = D(\vec{a}_{rq}, \vec{b}_{yx})$ defined via the symmetrized formula:

$$D(\vec{a}_{qr}, \vec{b}_{xy}) = \frac{1}{2}(D_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}) + D_{\mathrm{d}}(\vec{a}_{rq}, \vec{b}_{yx})). \tag{4}$$

4. ABSURDIST II Graph Matching Algorithm

The original ABSURDIST algorithm[3] created a correspondence matrix of possible cross-system translations and updated it at each iteration using a net input including external similarity, excitation, and inhibition components. After the iterative process converged, or after a fixed number of iterations, the values in the correspondence matrix were used to select the mapping between the two graphs.

In this section we describe the ABSURDIST II algorithm, which, while retaining the main concepts of the original ABSURDIST, introduces a number of improvements over the original ABSURDIST algorithm: the input representations can include not just weighted, undirected, unlabeled relations, but also any combination of undirected or directed, weighted or unweighted, and labeled or unlabeled relations (as described in section 3, above); sparsity in the systems' representations is

exploited to decrease computational complexity; the overall activity in correspondence units is dynamically adjusted to assure appropriate patterns of connectivity across systems; a post-iteration process is introduced to assure a 1-to-1 mapping selection based on the correspondence matrix. Moreover, we explain the algorithm in a more straightforward optimization framework.

4.1. *The iterative algorithm*

The goal of the ABSURDIST II algorithm is to obtain an $n \quad m$ correspondence matrix $C \in \mathbb{R}^{nm}$, whose element $C(q,x)$ denotes the correspondence between the concepts A_q in system A and B_x in system B. All elements of the matrix are in the range $[c_{\min}, c_{\max}]$, with $c_{\min} = 0$, $c_{\max} = 1$. In other words, $C \in Q$, where the cube Q is defined as

$$Q = \{C \in \mathbb{R}^{nm} : (\forall q, x) \quad (c_{\min} \quad C(q,x) \quad c_{\max})\}.$$

The elements of a correspondence matrix are interpreted in such a way that if the element $C(q,x) = c_{\min}$, then there is no correspondence between the two concepts in the two systems; the maximum possible value, $c_{\max}$, means the maximum possible correspondence. If a correspondence matrix is a permutation matrix, i.e. it contains a single 1 in each row and in each column, it can be unambiguously interpreted as an alignment of the two conceptual systems by the appropriate permutation.

The ABSURDIST II carries out an iterative process in the cube Q. It starts with the starting point C_0. At each step t, we compute *net input*

$$N_t = N(C_t).$$

The correspondence matrix is modified by the net input in a way that is intended to keep the C_{t+1} within Q:

$$C_{t+1} = C_t + V_t, \tag{5}$$

where the update V_t is obtained by "damping" the net input in some way:

$$V_t = \mathrm{Damp}(LR_t, C_t). \tag{6}$$

The damping function Damp has the following property: each of the component $V_t(q,x)$ of the vector V_t is between 0 and $LR_t(q,x)$. The parameter $L > 0$ is known as the learning rate. As discussed in Section 4.2 below, there are are various way to construct the damping function as to ensure that C_{t+1} stays in Q.

As in the original ABSURDIST, the net input $N(C_t)$ matrix at step t is computed based on the correspondence matrix C_t:

$$N(C) = \alpha E + \beta R(C) \quad \chi I(C) \tag{7}$$

and includes the external similarity matrix E, excitation matrix $R(C)$, and inhibition matrix $I(C)$. The elements $E(A_q, B_x)$ of the external similarity matrix E describe our *a priori* beliefs about the similarity of concepts A_q and B_x in their respective concept systems, and are constant throughout the iteration process. The

excitation and inhibition at the step t are computed based on the correspondence matrix at this step:

$$R_t(A_q, B_x) = \frac{1}{\max(n \quad 1, m \quad 1)} \sum_{r \neq q} \sum_{y \neq x} S(\vec{a}_{qr}, \vec{b}_{xy}) C_t(A_r, B_y) \tag{8}$$

where the *edge similarity function* $S(\vec{a}_{qr}, \vec{b}_{xy}) \in [0; 1]$ is given by Eq. (3);

$$I_t(A_q, B_x) = \frac{1}{(n \quad 1) + (m \quad 1)} \left(\sum_{r \neq q} C_t(A_r, B_x) + \sum_{y \neq x} C_t(A_q, B_y) \right). \tag{9}$$

The excitation term R measures the strength of supportive hypotheses that are consistent with a given correspondence. Via excitation, correspondence between concepts A_r and B_y "spreads" to each such concept pair (A_q, B_x) where A_q takes part in the same kind of relations with A_r as B_x does with B_y. The goal of using the kind of expression we do is make A_q in A correspond to B_x in B if they take part in similar relations, *and* if the things they are related to themselves strongly correspond.

The inhibitory term I measures the strength of hypotheses that are inconsistent with a given hypothesis, where inconsistency involves a 2-to-1 mapping. It results in a negative contribution to $C(A_q, B_x)$ if A_q also has non-zero correspondence to other concepts from B than B_x, or if B_x also has non-zero correspondence to other concepts from A than A_q.

4.2. *Damping function and learning rate*

The damping function used in (Goldstone, Rogosky),[3] which we will be referring to as "GR2002", is constructed so that the update matrix $V_t = \mathrm{Damp}(LN_t, C_t)$ has the following elements:

$$V_t(A_q, B_x) = \begin{cases} LN_t(A_q, B_x) \quad (1 \quad C_t(A_q, B_x)), & \text{if } N_t(A_q, B_x) \quad 0; \\ LN_t(A_q, B_x) \quad C_t(A_q, B_x), & \text{if } N_t(A_q, B_x) < 0. \end{cases} \tag{10}$$

It can be shown that if the learning rate L is chosen sufficiently low, using damping (10) with the increment formula (5) will ensure that $C_{t+1} \in Q$ as long as $C_t \in Q$. A sufficient condition for this is

$$0 < L \quad \left(\max_{C \in Q} \|N(C)\|_\infty \right)^{1}. \tag{11}$$

Here the notation $\|C\|_\infty$ is used for the "infinity norm" of a matrix C, $\|C\|_\infty = \max_{A_q, B_x} |C(A_q, B_x)|$. With the excitation and inhibition given by Eqs. (8,9), the values of the edge similarity function $S(\vec{a}_{qr}, \vec{b}_{xy})$ and the external similarity values being both in the range $[0; 1]$, condition (11) is fulfilled if

$$0 < L \quad 1/\max\{|\alpha \min_{q,x} E(q, x) \quad \chi|, \ |\alpha \max_{q,x} E(q, x) + \beta \max(n \quad 1, m \quad 1)|\} \tag{12}$$

Other damping schemes can be used as well, e.g. "hard damping", which can be used with any $L > 0$:

$$V_t(A_q, B_x) = \begin{cases} 1 & C_t(A_q, B_x), \quad \text{if } C_t(A_q, B_x) + LN_t(A_q, B_x) > 1; \\ & C_t(A_q, B_x), \quad \text{if } C_t(A_q, B_x) + LN_t(A_q, B_x) < 0; \\ & LN_t(A_q, B_x), \quad \text{otherwise} \end{cases} \tag{13}$$

4.3. *ABSURDIST as a convergent optimization algorithm*

To analyze ABSURDIST, we will note that the net input matrix $N(C)$, as defined by Eqs. (7,8,9), with a symmetrized edge similarity (3), is a gradient of the following *energy functional* $K : \mathbb{R}^{nm} \to \mathbb{R}$:

$$K(C) = \alpha(E, C) + \frac{1}{2}(C, \mathcal{A}C) = \alpha(E, C) + \frac{\beta}{2}(C, \mathcal{A}_{\mathrm{R}}C) \quad \frac{\chi}{2}(C, \mathcal{A}_{\mathrm{I}}C). \tag{14}$$

Here (E, C) is a dot product in $\mathbb{R}^{nm}$, and $\mathcal{A} = \beta\mathcal{A}_{\mathrm{R}} \quad \chi\mathcal{A}_{\mathrm{I}}$ is an $nm \quad nm$ matrix whose elements are determined by the coefficients in the formulas for $R(C)$ and $I(C)$.

In terms of individual matrix elements, the parts of this functional can be represented as

$$(E, C) = \sum_q \sum_x E(A_q, B_x)C(A_q, B_x),$$

$$(C, \mathcal{A}_{\mathrm{R}}C) = \frac{1}{\max(n \quad 1, m \quad 1)} \sum_{q,r:r\neq q} \sum_{x,y:y\neq x} S(\vec{a}_{qr}, \vec{b}_{xy})C(A_q, B_x)C(A_r, B_y),$$

$$(C, \mathcal{A}_{\mathrm{I}}C) = \frac{1}{(n + m \quad 2)} \left(\sum_{q,r:r\neq q} \sum_x C(A_q, B_x)C(A_r, B_x) + \sum_{x,y:x\neq y} \sum_q C(A_q, B_x)C(A_q, B_y) \right).$$

The meaning of the three terms is quite transparent. The external similarity part (E, C) provides a reward for the correspondence matrix having positive values in the same positions as the external similarity matrix. The excitatory term $(C, \mathcal{A}_{\mathrm{R}}C)$ is a reward for structural similarity of the two graphs as aligned by the correspondence matrix. The inhibitory term $(C, \mathcal{A}_{\mathrm{I}}C)$ is a penalty for non-orthogonality of rows or columns of the correspondence matrix.

If the net input N_t were directly added to C_t without damping (i.e., if the damping function in (6) were an identity function, $\mathrm{Damp}(N) = N$), then the ABSURDIST algorithm would become Cauchy's classic steepest descent method for the minimization of $K(C)$.[12] It can be shown that ABSURDIST, which uses damped increments, has similar optimization properties. Namely, if the learning rate is low enough, the energy functional $K(C_t)$ is monotonically non-decreasing with t. The process always converges, either to a point in Q where $N \quad \nabla K = 0$, or to a point on a boundary of Q where any non-zero components of N are directed outside of

the cube. This point is almost always a local maximum of $K(C)$ on cube Q. (A theoretically possible exception is a saddle point of $K(C)$; however, convergence to such a point is extremely unlikely in practice).

Our full proof of the convergence properties above is rather cumbersome, and, in certain special cases, not fully strict; it is reserved for a separate technical report.[13] However, it's main idea is rather simple. Given the construction of the damping function, the absolute values of the components of the vector $V_t = \mathrm{Damp}(LN_t, C_t)$ are no greater than those of LN_t; this allows us to obtain the bound

$$0 \quad \|V_t\|_2^2 \quad (V_t, N_t) \quad \|N_t\|_2^2$$

for dot products, and to use it to analyze ABSURDIST convergence in a way similar to the standard steepest descent method.

According to our report,[13] a sufficient condition for the ABSURDIST convergence properties stated above to hold is

$$0 < L < 2/\|\mathcal{A}\|, \tag{15}$$

where $\mathcal{A}$ is the standard matrix norm induced on $\mathbb{R}^{nm} \to \mathbb{R}^{nm}$ by the vector 2-norm in $\mathbb{R}^{nm}$, that is

$$\|\mathcal{A}\| = \max_{C \neq 0} \frac{\|\mathcal{A}C\|_2}{\|C\|_2}.$$

Using the properties of matrix norms, it can be shown that condition (15) will be satisfied if

$$0 < L < 2/(\beta \min(n \quad 1, m \quad 1) + 2\chi). \tag{16}$$

Of course, convergence to a local maximum of $K(\ )$ does not necessary imply convergence to the global maximum. Nor is the local —or global—maximum necessarily a permutation matrix. Our report[13] contains some observations regarding the behavior of the local and global maxima of the energy functional. In the absence of external similarity ($\alpha = 0$), the following observations seem to hold for most typical conceptual systems:

> The matrix consisting of all ones will be a local convergence point of ABSURDIST if the ratio $\chi/\beta > 0$ is sufficiently low.
> As the χ/β ratio increases, permutation matrices are increasingly likely to become local convergence points.
> For sufficiently high values of χ/β many, or—for some pairs of conceptual systems—even all, possible permutation matrices become convergence points.

It is not generally possible to set the χ/β ratio so that the "correct" permutation—the one that provides the global maximum of the energy functional—becomes the unique local maximum. But in practice, especially on systems that are isomorphous or nearly isomorphous, our algorithm often converges to local maxima that are not too divergent from the global maximum.

4.4. *Experimental parameter settings*

In the experiments described in this article, we have set L at the middle of the smaller range of the two given by the constraints (12) and (16).

The choice of β is arbitrary, as long as α and χ are scaled accordingly; so we set $\beta = 1$.

We have used the parameter ratio $\chi/\beta = n \quad 2$. The choice was made based on the fact that this value is rather "critical" in our convergence analysis: if χ/β is greater than this value, then a large number of permutation matrices become local convergence points. While this particular choice of the χ/β may not be optimal, it worked better than, for example, the rather complicated heuristic that we tried in an earlier version of this paper.[14]

In the experiments with no external similarity, α was zero. In experiments with external similarity, we used $\alpha/\beta = n \quad 1$, to ensure that the external similarity is "taken seriously" by the algorithm: for these values of the parameters, the penalty for one "missed" element of the external similarity matrix $E(q, x) = 1$ is equal to that for $n \quad 1$ completely mismatched edges.

4.5. *Exploiting Sparsity to Reduce Computational Complexity*

At every iteration step in the original ABSURDIST algorithm, the excitation $R(A_q, B_x)$ is computed for each concept pair as per Eq. (8). Directly computing the n^2 values of $R(\ ,\ )$ by this formula requires $O(n^4)$ operations at each iteration. Fortunately, the matrix or graph describing a typical conceptual system is rather sparse. The average degree D of a node is much smaller than n. Moreover, in many domains, as n grows, D does not grow as fast as $O(n)$. We will show how to use this sparsity to carry out each iteration in $O(n^2D^2)$ time instead of $O(n^4)$.

For simplicity, we will use the symbol A to refer to the set of all subscripts $\{1, 2, \ldots, n\}$, as well as to the conceptual systems A itself, and use the following notation:

$$C(S_1, S_2) = \sum_{q \in S_1} \sum_{r \in S_2} C(A_q, B_r);$$

$$\Omega(q) = \text{set of all indices of all nodes of } A \text{ connected to } A_q;$$

$$\Omega'(q) = \Omega'(q) \cup \{q\}; \qquad \Phi(q) = A \backslash \Omega'(q).$$

The sign $\backslash$ is used for set difference. Analogous notation is used for subsets of B.

Because $A \backslash \{q\} = \Omega(q) \cup \Phi(q)$, and $S(\vec{0}, \vec{0}) = 1$, we can decompose the excitation as follows: $(n \quad 1)R(A_q, B_x) = \sum_{r \in \Omega(q) \cup \Phi(q)} \sum_{y \in \Omega(x) \cup \Phi(x)} S(\vec{a}_{qr}, \vec{b}_{xy}) C(A_r, B_y) = Z_{11}(q, x) + Z_{12}(q, x) + Z_{21}(q, x) + Z_{22}(q, x)$, with

$$Z_{11}(q, x) \ = \sum_{r \in \Omega(q)} \sum_{y \in \Omega(x)} (S(\vec{a}_{qr}, \vec{b}_{xy}) C(A_r, B_y));$$

$$Z_{12}(q,x) = \sum_{r\in\Omega(q)} (S(\vec{a}_{qr},\vec{0})C(\{r\},\Phi(x));$$
$$Z_{21}(q,x) = \sum_{b\in\Omega(x)} (S(\vec{0},\vec{b}_{xy})C(\Phi(q),\{y\}));$$

$Z_{22}(q,x) = S(\vec{0},\vec{0})C(\Phi(a_q),\Phi(b_x)) = C(A,B) \quad C(\Omega(q),B) \quad C(\{q\},B)$ $C(A,\Omega(x))+C(\Omega(q),\Omega(x))+C(\{q\},\Omega(x)) \quad C(A,\{x\})+C(\Omega(a_q),\{x\})+C(\{q\},\{x\})$. This decomposition allows us to compute the matrix of $R(a_q,b_x)$ as follows:

(1) Each value $Z_{11}(q,x)$ is computable in $O(D^2)$ operations, making the cost for the full set of them $O(n^2D^2)$
(2) The full set of row sums $C(\{q\},B)$ for all q, column sums $C(A,\{x\})$ for all x, and then the total sum $C(A,B)$, can be computed in $O(n^2)$ operations.
(3) Since $C(\{q\},\Phi(x)) = C(\{q\},B) \quad C(\{q\},\Omega'(x))$, each of the values $C(\{q\},\Phi(x))$ can be obtained, using the pre-computed sums $C(\{q\},B)$, in $O(D)$ operations, making the cost for the entire matrix of them $O(n^2D)$.
(4) Using the pre-computed sums $C(\{q\},\Phi(x))$, each value $Z_{12}(q,x)$ can be computed in $O(D)$ operations, with the $O(n^2D)$ cost for the entire matrix. Sums $Z_{21}(q,x)$ are computed similarly.
(5) Once the row and column sums from step 2 are available, computing each of the nine added values in $Z_{22}(q,x)$ will cost no more than $O(D^2)$, making the cost of the entire matrix $O(n^2D^2)$.

None of the steps above costs more than $O(n^2D^2)$, which means that the total cost of computing the matrix of excitation values $R(A_q,B_x)$ is $O(n^2D^2)$.

The formulas above are significantly simplified in the case of an unweighted unlabeled graph (a conceptual system with a single binary-valued relation type). In such a system $S(\vec{a}_{qr},\vec{0}) = 0$ for any $r \in \Omega(q)$, and $S(\vec{a}_{qr},\vec{b}_{xy}) = 1$ for any pair ($r \in \Omega(q)$, $y \in \Omega(x)$). Therefore, $Z_{12}(q,x) = Z_{21}(q,x) = 0$, and $Z_{11}(q,x) = \sum_{r\in\Omega(q)}\sum_{y\in\Omega(x)} C(A_r,B_y)$.

We use a similar transformation to compute the inhibition matrix $I(A_q,B_x)$ in $O(n^2)$ operations at each iteration.

4.6. *Final mapping selection*

To produce a 1-to-1 mapping between the two conceptual systems based on the correspondence matrix $C(a_q,b_x)$, we use the following procedure:

1. Create sets $U_A = \emptyset$, $U_B = \emptyset$. They will be used to store indices of already mapped variables.
2. If U_A covers all concepts from A, or U_B covers all concepts from B, terminate.
3. Select the largest element $C(a_q,b_x)$ in the correspondence matrix such that q does not appear in U_A and x does not appear in U_B. Use random tie-breaking if there are several elements with the same value.
4. Map x to q. Add q to U_A; add x to U_B.

5. Go to step 2.

If the two systems have the same number of concepts ($n = m$), the above algorithm will find a 1-to-1 mapping for all concepts, providing an inexact isomorphism between the two graphs. If $n < m$, the mapping will be found for all concepts from A, providing an isomorphism between A and an n-node subgraph of B. The situation when $n > m$ is analogous.

5. Assessing mapping quality

To see how far our mapping P is from an exact isomorphism, we use the *relations mismatch measure* $\mu(A, B, P)$, which is defined, in the case of conceptual systems of the same dimension ($n = m$) as

$$\mu(A, B, P) = \frac{1}{2}\sum_{q=1}^{n}\sum_{r=1}^{n} D_{\mathrm{d}}(a_{qr}, b_{P(q)P(r)}) + \frac{\alpha}{\beta}(n \quad 1)\sum_{q=1}^{n}\sum_{x\neq P(q)} E(A_q, B_x) \quad (17)$$

In the case of an unweighted, unlabeled, undirected graph, the first term of this expression can be visualized as the "mismapped edge count", i.e. the number of edges that are present in $P(A)$ but absent in the corresponding positions in B, plus the number of edges that are present in $P(B)$ but absent in A. The second term represents a penalty for the mismatch between the permutation P and the positions of non-zeros in the external similarity matrix E.

The coefficients in Eq. (17) are chosen so that maximizing the energy functional $K(C_P)$ on the set of permutation matrices P is equivalent to minimizing the relations mismatch measure $\mu(P)$. This is so because for any permutation P,

$$K(C_P) = K_0 + \frac{\beta}{n \quad 1}\mu(A, B, P),$$

with the constant K_0

$$K_0 = \mathrm{const} = \frac{\alpha n}{2(n \quad 1)} + \alpha\sum_{q,x} E(A_q, B_x).$$

Therefore, if the ABSURDIST iterative process has converged to the global energy maximum, and that maximum was a permutation matrix C_P for some permutation P, then P is also the permutation that globally minimizes the relations mismatch measure μ.

6. Performance Evaluation of ABSURDIST II

We have implemented ABSURDIST II in Java. This section describes several experiments carried out on randomly generated graphs. In each experiment $N_R = 100$ runs was performed. In each run, a random conceptual system (the "original system", S_O) with n nodes and E edges was created. It was matched with ABSURDIST to a "noisy system" S_N generated from S_O by adding noise.

In each experiment, we run our version of ABSURDIST on the system pair (S_O, S_N) until it converged (the convergence criterion was $\|V_t\|_2^2/n^2 \quad \delta = 10^{\ 10}$), or until a fixed cut-off number of iterations ($N_{\rm it} = 20000$, except in Test 2) has been performed, whichever happens earlier.

In the experiments reported in this section, parameters α, β and χ, as well as the learning rate L, were set as described in section 4.4. All elements of the correspondence matrix were initialized with $C(A_q, B_x) = 0.5$. Most results presented here were obtained with GR2000 damping; those obtained using hard damping are labeled accordingly.

A good way to measure the quality of mapping P between S_O and S_R found with ABSURDIST II would be to compare it to the best mapping possible. That is, we could use a graph matching algorithm (e.g. presented by Messmer[5]) guaranteed to find the best possible isomorphism P between the systems with respect to the relation mismatch measure (17), and then look at the adjusted relation mismatch measure $\mu(S_O, S_N, P) \quad \mu(S_O, S_N, P\ )$. A difference of zero would mean that we have found the best possible mapping.

As finding the guaranteed best isomorphism has exponential complexity, we use a simpler approach to estimate the *base mismatch measure* $\mu \ = \mu(S_O, S_N, P\ )$ in each experiment. When noise levels are low, it is reasonable to assume that the best mapping between S_O and S_R is the identity isomorphism I. We measure the relation mismatch measure $\mu(S_O, S_N, P\ )$ for the identity isomorphism, and subtract this *base mismatch measure* from the mismatch measure of the permutation obtained by ABSURDIST II.

We then normalize the mismatch measure by dividing it by $\mu_{\max}$, which serves as the estimate for the maximum possible value of the relation mismatch measure between two graphs with the characteristics generated in a particular experiment. The resulting *normalized adjusted relation mismatch measure (NARMM)*

$$\varepsilon = \frac{\langle \mu(S_O, S_N, I) \rangle \quad \mu}{\mu_{\max}}, \tag{18}$$

averaged over the $N_R = 100$ runs, is reported in each experiment.

As the noise level increases above several percentage points, the identity permutation won't necessarily remain the best permutation: some other permutations may provide a better match between the systems, in the sense of minimizing the relation mismatch measure. Thus in some experiments NARMM may start falling as the noise level increases beyond certain point, and may eventually become negative, indicating that ABSURDIST II has found a permutation that, although perhaps not globally optimal, better fits the noisy data than the identity permutation does.

To show how different the permutation P found by ABSURDIST II is from the identity permutation, we also report for some experiments the *concept mismatch* measure, i.e. the average number of concepts in S_O that are not mapped to their counterparts in S_N, divided by the system size n.

Test 1 (a–b): Noise tolerance. In this test, S_O was an unlabeled, unweighted,

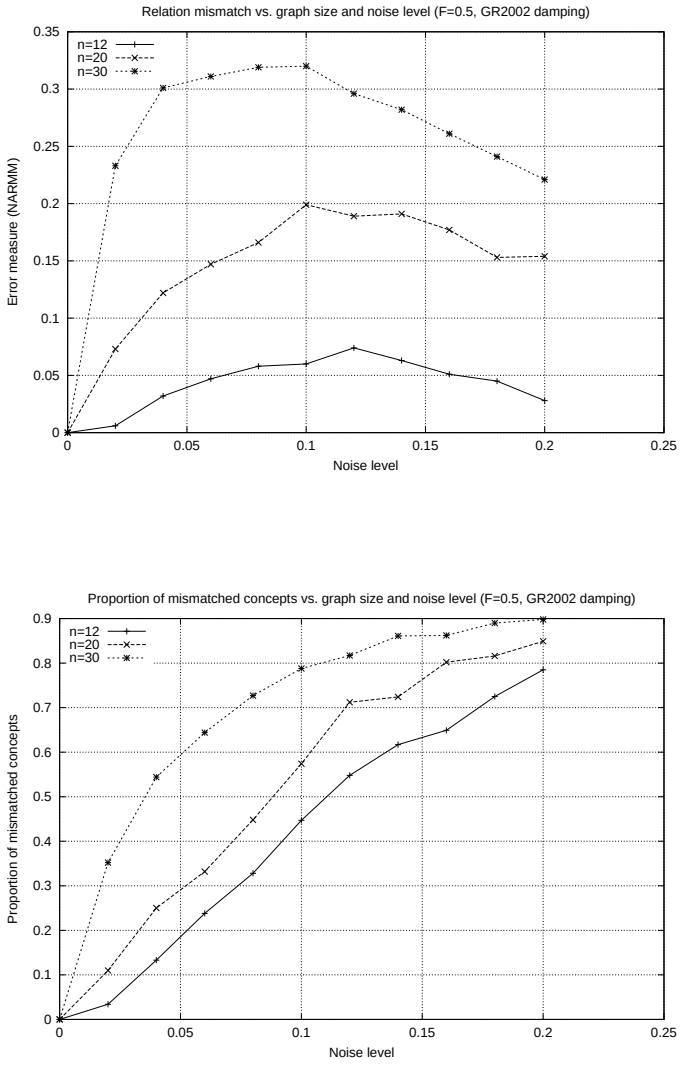

Fig. 1. Test 1(a), GR2002 damping: ε and normalized node mismatch vs. noise level ν and graph size n for fixed graph density $F = 0.5$.

undirected graph with no non-trivial homomorphisms (no degrees of symmetry) with n nodes. Each of the potential $n(n \quad 1)/2$ edges was actualized with probability F; this value F $(0 < F \quad 0.5)$ is referred to as the *density* of the graph. The noisy system S_N was created as an inexact copy of S_O, where each of $n(n \quad 1)/2$ existing or potential edges was independently switched from present to absent or vice versa

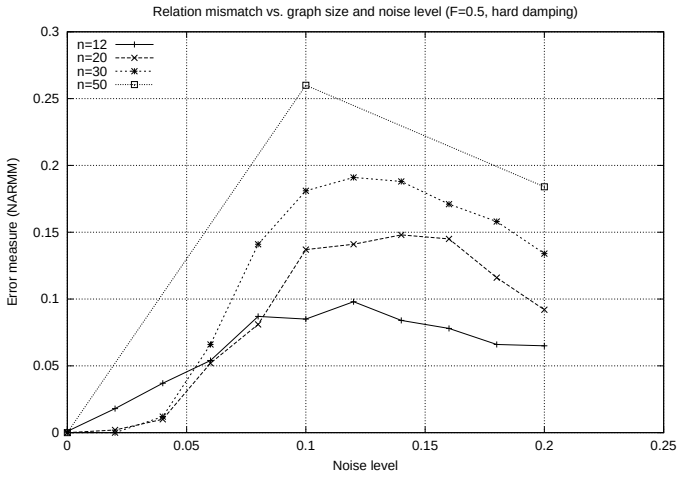

Fig. 2. Test 1(a), hard damping: ε and normalized node mismatch vs. noise level ν and graph size n for fixed graph density $F = 0.5$.

with the probability ν. There was no external similarity.

The average number of edges in these graphs was estimated as $E = Fn(n$ $1)/2$, giving maximum possible distance between graphs $\mu_{\max} = 2E$. We reported NARMM computed by (18) with this $\mu_{\max} = 2E$, as well as the concept mismatch measure and the average number of iterations performed.

This test shows that, for our parameter settings, the correct mapping is always found when the two graphs are exactly isomorphous. When matching non-isomorphous graphs, the mapping quality deteriorates with increasing graph size, given the same noise level (Figs. 1, 2). On the other hand, it improves with increasing graph density (Figs. 3, 4). Most of the results obtained with the GR2002 damping (same as in our earlier report[14]) show 1.5- to 3-fold improvement of the error measure (NARMM) for the same graph size and noise level, compared to that obtained in the experiments with our earlier χ/β and learning rate settings.[14].

Comparing Fig. 1 with 2, or Fig. 3 with 4 shows that when hard damping is used, the ABSURDIST II iterative process converges, on average, to a matrix resulting in poorer-quality permutation than the one obtained with GR2002, even though with GR2002 iterations are often stopped by the iteration number cut-off before the convergence criterion is satisfied. The difference is especially pronounced when on harder problems, i.e. lower-density graphs. We therefore carried out further experiments only with GR2002 damping.

(**Test 1 (c): Iteration count to convergence.**

Comparing formulas (10) and (13) in Section 4.2 allows one to surmise that hard

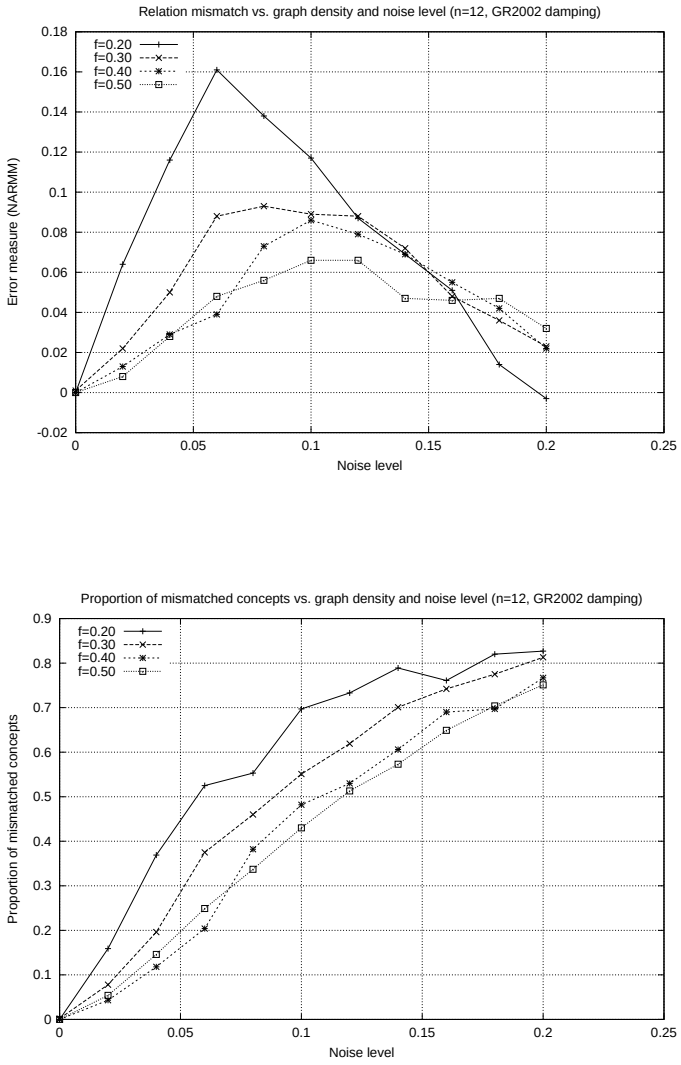

Fig. 3. Test 1(b), GR2002 damping: ε and normalized node mismatch vs. ν and F for fixed $n = 12$.

damping results in much faster convergence. Moreover, it is natural to conjecture that, if the convergence point is a matrix consisting only of 0s and 1s, an iterative process with hard damping will converge to that point in a finite number of iterations, $O(n^2)$. (This conjecture is based on the somewhat optimistic assumption that at each iteration at least one more correspondence matrix element will achieve its

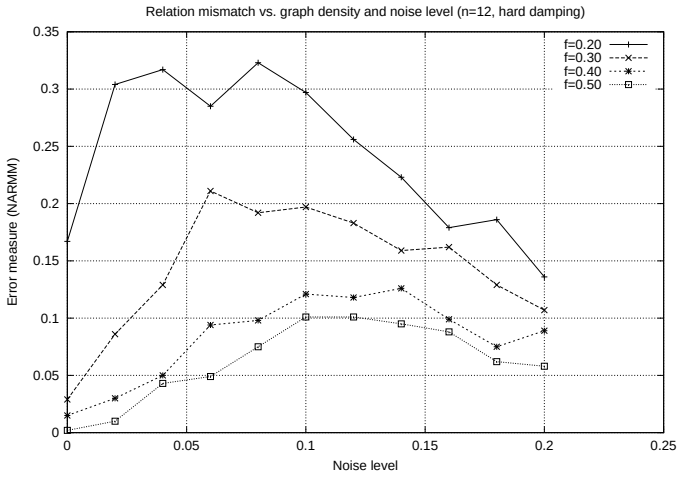

Fig. 4. Test 1(b), hard damping: ε and normalized node mismatch vs. ν and F for fixed $n = 12$.

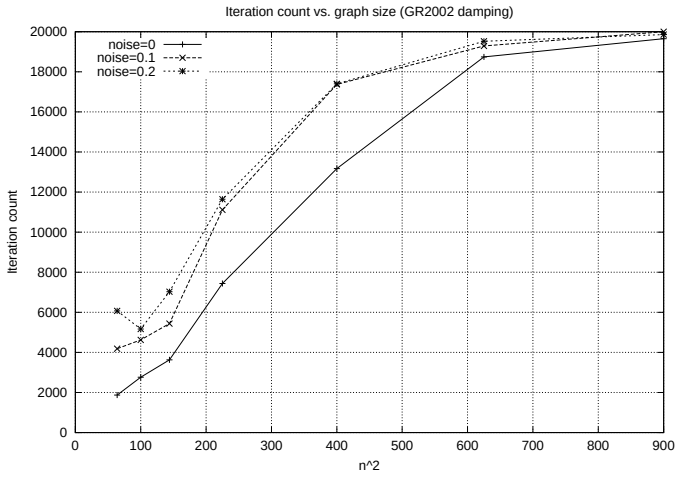

Fig. 5. Test 1(c), GR2002 damping: averaged iterations count to convergence or termination vs. noise level ν and graph size n for fixed graph density $F = 0.5$.

"final" value of 0 or 1).

This seems to be confirmed by our experiments. Figure 6 shows the average

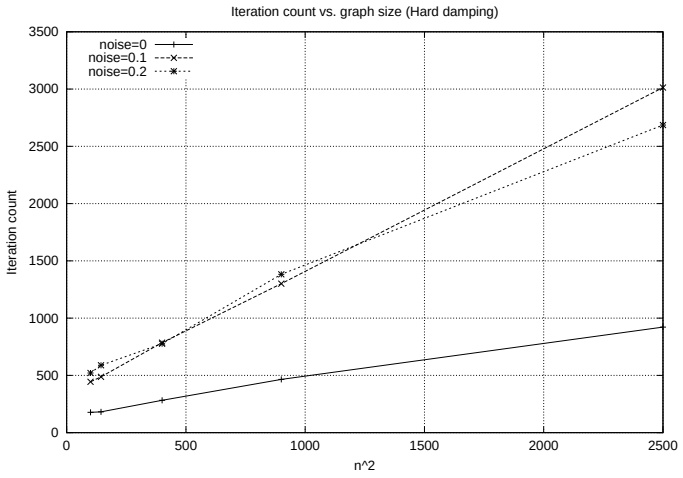

Fig. 6. Test 1(c), hard damping: averaged iterations count to convergence vs. noise level ν and graph size n for fixed graph density $F = 0.5$.

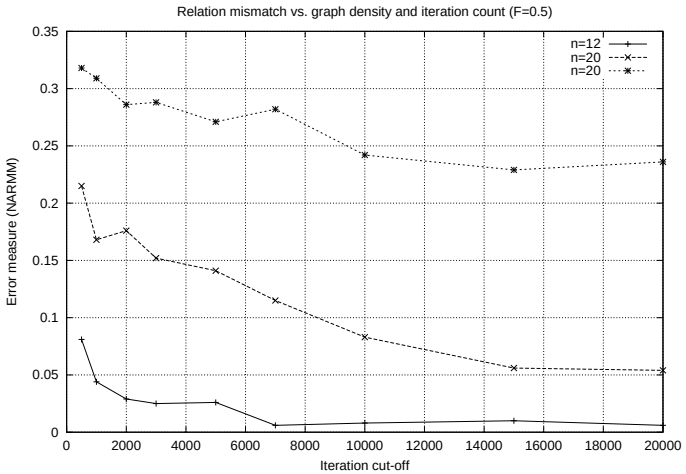

Fig. 7. Test 2(a): ε vs. iteration cut-off number and graph size for fixed graph density $F = 0.5$ and noise level $\nu = 0.02$. GR2002 damping.

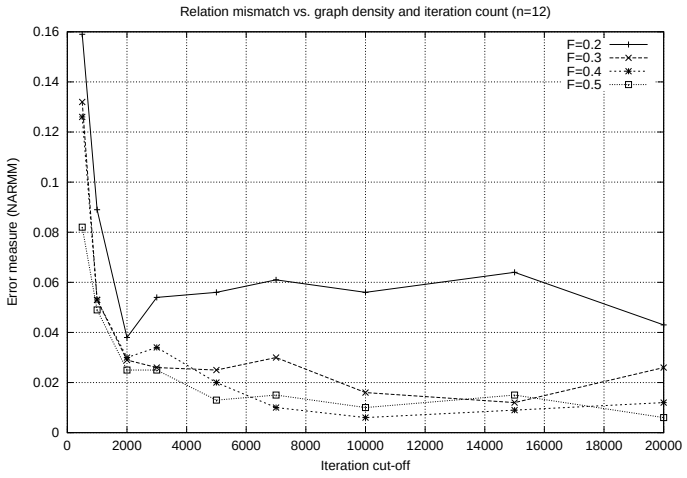

Fig. 8. Test 2(b): ε vs. iteration cut-off number and graph density for fixed graph size $n = 12$ and noise level $\nu = 0.02$. GR2002 damping.

iteration count to convergence in Test 1 with hard damping for $n = 10$ through 50. When two identical systems are matched (no noise), the process converges in approximately $140 + 0.36n^2$ iterations; when noisy systems are matched ($\nu = 0.1$ or 0.2), the iteration count to convergence can be approximated as $400 + 1.08n^2$.

On the other hand, using the GR2002 damping procedure results in a much slower convergence. Already at n =20–25, it takes more than 20,000 iterations (our iteration number cut-off) for ABSURDIST II with GR2002 damping to converge within the same termination criterion, $\|V_t\|_2^2/n^2 \quad \delta$. The iteration count for Test 1 with GR2002 damping shown in Figure 5 is based on averaging all the counts from all N_R runs, both those that converged in fewer than N_{it} iterations, and those that were terminated after reaching that cut-off number; thus it only provides a lower bound on the number of iterations that would be needed to achieve convergence.

Test 2: Convergence speed. In this test (Figs. 7,8), we observed the convergence behavior of our algorithm. Each point in the graphs corresponds to a series of $N_R = 100$ runs performed with a given cut-off iteration number N_{it}. Pairs of unlabeled, unweighted, undirected graphs were generated as in Test 1, with noise level $\nu = 0.02$.

The apparent non-monotonicity of some curves is explained by the fairly high standard error of the performance numbers (e.g., the standard error for NARMM in the experiments with $n = 12$, for $N_R = 100$, was around 0.003 0.01). We don't show error bars to avoid cluttering the graphs.

Test 3: Coverage noise and intensity noise on weighted graphs. In

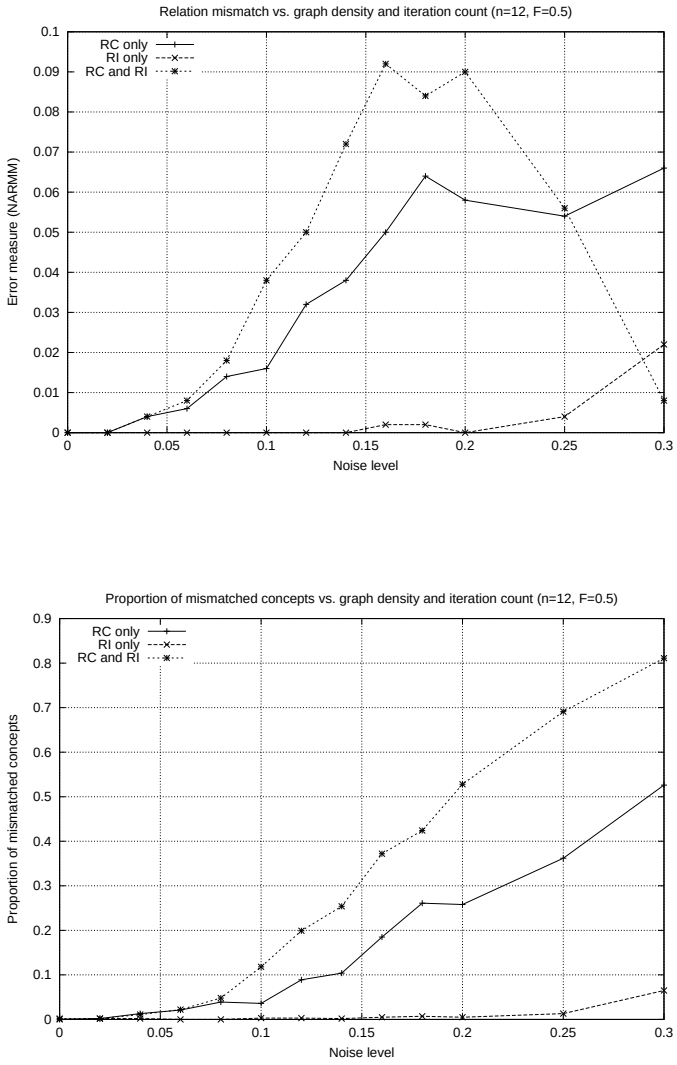

Fig. 9. Test 3: ε and concept mismatch vs. noise level for coverage noise RC, intensity noise RI, and their combination. $n = 12$, $F = 0.5$. GR2002 damping.

this test we generated random, undirected weighted graphs with one relation type. The weight of each relation was uniformly distributed on the $[0; 1]$ interval. Each edge was actualized, as in Test 1, with probability $F = 0.5$, giving the average of $E = Fn(n \quad 1)$ edges per graph.

The maximum possible relation mismatch $\mu_{\max}$ between two such weight-

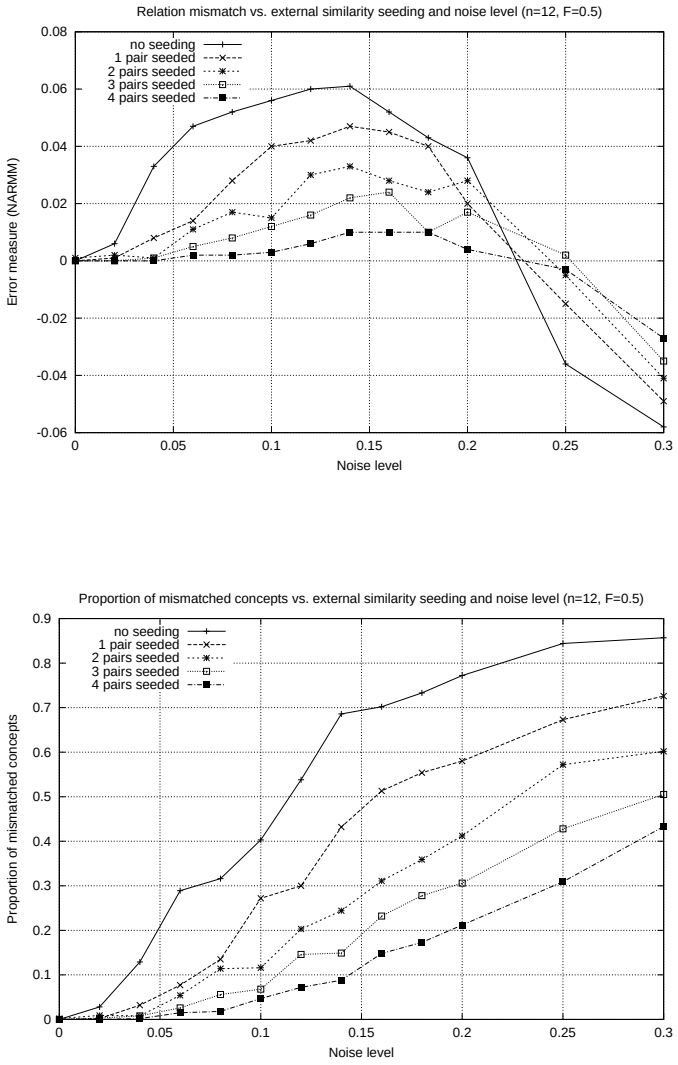

Fig. 10. Test 4: ε and normalized node mismatch vs. noise level for various amount of external seeding. 0 n_S 4, $n = 12$, $F = 0.5$. GR2002 damping.

ed graphs, to be used in Eq. (18) for computing NARMM, was estimated as $\mu_{\max} = E = Fn(n \quad 1)$.

Two types of noise were tested. In the experiments with *coverage noise* (RC), the difference between the noisy system S_N and the original system S_O consisted in some edges created or destroyed according to the same rules as in Test 1. The

creation/destruction probability for each existing or potential edge was ν_C.

In the experiments with the *intensity noise* (RI), S_O and S_N had the same topology, but the intensity of each edge present in S_O was changed in S_N. The difference between the weights of the edges in the two graphs was, for each edge, a random value distributed uniformly on the $[\ \nu_C; \nu_C]$ range. However, the intensity change was limited so as to never take the resulting weight outside of the $[0; 1]$ interval.

Finally, we carried out experiments with the two types of noise combined. Coverage noise was applied first, and then intensity noise was applied to those edges that existed in S_O and remained in S_N. In the experiments presented here, the two noise levels were equal, $\nu_C = \nu_I$.

The results shown in Figure 9 are quite interesting. For a pair of system with intensity noise only, the NARMM is very close to zero for the noise level up to 0.2. This indicates that ABSURDIST II is quite tolerant to random variation to the values of weighted edges, matching graphs correctly as long as the topological structure is preserved. This finding may be of importance in applications where the weights of the relations come from real-world data and, by their nature, are imprecise.

The NARMM for coverage-noise-only weighted-graphs experiments (the "RC" curve in Figure 9) is slightly better than that obtained in similar experiment with an unweighted graph with the same size and density (the $n = 12$ curves in Figure 3, Test 1(a)); the concept mismatch measure is significantly better. Our explanation is that for the same coverage noise level, in a pair of weighted graph the identity permutation is more likely to remain the optimal permutation. This is so because when unweighted graphs are matched, if a non-empty edge in one graph is aligned with any other edge in the other graph, this alignment brings no contribution to the mismatch measure (17); in contrast, when weighted graphs are matched, aligning an edge in one graph with a different-weight edge in the other graph would give a non-zero contribution to (17). Thus varying weights of different edges work as labels, ensuring the optimality of the identity permutation under higher coverage noise levels, and helping ABSURDIST II match edges (and, therefore, concepts) correctly.

Test 4: External similarity seeding. In this test we explored the effect of external similarity input on ABSURDIST II mapping quality. We used pairs of unlabeled, unweighted graphs generated as in Test 1, with $n = 12$ nodes, density $F = 0.5$, and the noise level ν from 0 to 0.3. Unlike other tests, external similarity was used along with the excitation and inhibition in the net input at each step. The external similarity matrix contained ones in n_{S} randomly chosen diagonal positions, and zeros elsewhere. Thus it provided external "seeding", tying n_{S} concepts in the original system to their counterparts in the noisy system.

The results, shown in Figure 10 for $n_{\mathrm{S}} = 0$ through 4 show that the presence of even one external seeding point improves mapping quality significantly compared

to the results obtained with internal similarity only. Adding extra seeding points improves the mapping quality further, but the marginal gains diminish as more correspondences are externally seeded. The improvements in mapping garnered by seeding correspondences far outstrips the gains predicted if only the seeded correspondence itself was correctly mapped. This is due to the ripples of influence that one seeded correspondence has on fixing other correspondences.

7. Conclusion and Future Work

We introduced the attributed graph representation of conceptual systems, extending the ABSURDIST framework to handle systems with multiple relation types, including weighted and directed relations. We exploited sparsity of the relation graphs to improve efficiency and scalability of the algorithm.

We illustrated the behavior of ABSURDIST II on some simple models. We have shown that our iterative approach inspired by constraint-propagation neural networks can be used to match concept systems represented by purely topological structures (unweighted graphs). When noise is introduced in the form of adding and deleting edges, the resulting translation is considerably poorer than when noise is added to the weights. That is, distorting a system by "flipping" X% of the edges is more damaging than distorting a system by altering weights of all the edges by X%.

Future work may include applying the algorithm to conceptual systems from real-world domains: ontologies, dictionaries, database schemas. We have already achieved some success using ABSURDIST II for matching hundreds of terms in bilingual texts. We can investigate modifying the damping scheme to speed up convergence without jeopardizing the quality of the resulting permutation; or simply replacing our steepest-descent-based optimization process with some other non-convex optimization techniques known in quadratic programming.

It would be useful to compare the effectiveness and efficiency of our approach to those of already implemented classical methods guaranteed to find the best mapping with respect to the target mismatch function, such as those described by Messmer.[15]

Still, in terms of real-world translation tasks, ABSURDIST II already offers benefits over many existing graph matching algorithms: when 1-to-1 mapping is not enforced, it allows multiple concepts from a richer conceptual system to map onto a single concept from a more impoverished system; it provides a natural way to incorporate both internal and external sources of information; and it accommodates several different graph structures. More generally, it offers the promise of showing how the meaning of a concept can be based simultaneously on its role within its conceptual system and on its external grounding, and that these two sources of information cooperate rather than compete.

Acknowledgments

This research was supported by a grant from the Lockheed Martin Corporation. We would like to thank Todd Hughes, Brian Rogosky, and Ben Ashpole for helpful comments.

Preliminary work on ABSURDIST II was presented as a paper [14] at the FLAIRS 2004 conference in Miami Beach in May 2004.

References

1. J. Fodor. *Concepts: where cognitive science went wrong*. Clarendon Press, 1998.
2. D.B. Lenat and E.A. Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47:185–250, 1991.
3. Robert L. Goldstone and Brian J. Rogosky. Using relations within conceptual systems to translate across conceptual systems. *Cognition*, pages 295–320, 2002.
4. Robert L. Goldstone, Ying Feng, and Brian J. Rogosky. Connecting concepts to each other and the world. In Diane Pecher and Rolf Zwaan, editors, *Grounding cognition: The role of perception and action in memory, language, and thinking*. Cambridge University Press, 2005.
5. Bruno T. Messmer. *Efficient Graph Matching Algorithms*. PhD thesis, Universität Bern, 1995.
6. Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.
7. A. Rangarajan and E. Mjolsness. A Lagrangian relaxation network for graph matching. In *Proc. Int. Conf. Neural Networks*, volume 7, pages 4629–4634. Inst. Electrical & Electronics Engineers, 1994.
8. Marcello Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11:1933–1955, 1999.
9. Kristina Schadler and Fritz Wysotzki. A connectionist approach to structural simiarity determination as a basis of clustering, classification and feature detection. In *Principles of Data Mining and Knowledge Discovery*, pages 254–264, 1997.
10. Kristina Schädler and Fritz Wysotzki. Comparing structures using a Hopfield-style neural network. *Applied Intelligence*, 11:15–30, 1999.
11. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *18th International Conference on Data Engineering (ICDE)*, pages 117–128, 2002.
12. Michael T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, New York, 2002.
13. Ying Feng, Robert L. Goldstone, and Vladimir Menkov. The ABSURDIST algorithm for matching concept systems as a converging optimization method, 2004. Preliminary version. Available at http://www.cs.indiana.edu/~ yingfeng/ABSURDIST/.
14. Ying Feng, Robert L. Goldstone, and Vladimir Menkov. ABSURDIST II: A graph matching algorithm and its application to conceptual system translation. In Valerie Barr and Zdravko Markov, editors, *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Symposium Conference, Miami Beach, Fla., USA*, pages 640–645. AAAI Press, 2004.
15. B. Messmer and H. Bunke. Fast error-correcting graph isomorphism based on model precompilation. Technical report, Institut fur Informatik, Universität Bern, 1996.