

# Graph Matching by Graduated Assignment

Steven Gold

Department of Computer Science  
Yale University  
New Haven, CT 06520

Anand Rangarajan

Department of Diagnostic Radiology  
Yale University  
New Haven, CT 06520

## Abstract

*A new algorithm for graph matching, which uses graduated assignment is presented, along with experimental results demonstrating large improvements in speed and accuracy over previous techniques. The softassign, a novel constraint satisfaction technique, is applied to a new graph matching energy function that uses a robust, sparse distance measure between the links of the two graphs. The softassign, which has emerged out of the neural network/statistical physics framework, enforces two-way (assignment) constraints without the use of penalty terms. The algorithm's low order computational complexity [ $O(lm)$ , where  $l$  and  $m$  are the number of links in the two graphs] compares favorably with most competing approaches. The method, not restricted to any special class of graph, is applied to subgraph isomorphism, weighted graph matching, and attributed relational graph matching. Experiments on graphs generated from images and on randomly generated graphs, including benchmarks against a relaxation labeling algorithm and an algorithm employing Potts glass dynamics are reported. Over twenty-five thousand experiments were conducted. No comparable results have been reported by any other graph matching algorithm before in the research literature.*

## 1 Introduction

Because of the great representational power of graphs, graph matching has been the focus of much research, especially in areas of cognition such as computer vision [1]. Unfortunately graph matching is an intractable problem and most heuristics have resulted in algorithms that are very slow and of restricted applicability. For example branch and bound techniques are of exponential time complexity and consequently only applicable in special situations where domain knowledge is included to reduce the search space size [2, 3]. Non-linear optimization techniques such as relaxation labeling [4, 5, 6] or previous efforts in neural networks [7, 8, 9] have also suffered from problems related to speed, accuracy and restricted applicability. See [10] for a more comprehensive review.

Using newly developed techniques from statistical physics, such as the *softassign*, we introduce a new, graduated assignment graph matching algorithm which addresses many of the above problems present in previous algorithms. The algorithm works on unweighted, weighted, or attributed relational graphs

with missing or extra links and missing or extra nodes. Attributed relational graphs (ARGs) are graphs with multiple link types and multiple attribute types attached to each node. No structural restrictions are placed on the graphs in terms of planarity, fan-out, links to neighbors, etc. These graphs are unlabeled (or if labeled the algorithm is invariant under permutation of the labeling).

The algorithm combines the use of a softassign with a new formulation of a graph matching energy function that uses a robust, sparse distance measure between the links of the two graphs. Softassign is a technique for satisfying two-way (assignment) constraints without penalty terms that is analogous to the softmax which enforces a one-way constraint. The first use of the softassign was in an algorithm for the assignment problem [11]. This technique is ideally suited to the solution of matching problems which require two-way constraints, where an object in one set must be matched to at most one object to the other set and vice versa. The resulting optimization dynamics for matching problems can be described as one of graduated assignment. Graduated assignment algorithms iteratively approximately solve a succession of assignment problems in a graduated fashion, slowly becoming more exact as the temperature is lowered. The graduated assignment framework has already been applied to parametric assignment problems—point matching [12, 13, 14] and quadratic assignment problems—graph matching (equal size graphs only)[14, 15]. However the graph matching algorithm in [14, 15] was only applicable to the matching of graphs with no missing or extra links and no missing and extra nodes. Here we have formulated a new energy function, including an explicit encoding of sparsity, and the resulting new algorithm has been applied to the much more difficult problem of matching graphs with missing and extra links and missing and extra nodes.

We have included some benchmarks to analyze the impact of using softassign instead of the Potts glass dynamics (softmax and a penalty term) [16, 7] in graph matching. However, note that the Potts glass benchmark uses the new energy function, the formulation of which is also a significant, original contribution of this work.

Strong evidence is presented that this algorithm may represent a breakthrough in the development of a graph matching method that is both fast and ac-

curate. It is of low order computational complexity —  $O(lm)$  where  $l$  and  $m$  are the links of the two graphs — yet powerful enough to often work on intractable problems like subgraph isomorphism, as well as weighted graph matching and attributed relational graph matching. We have demonstrated that it performs well on a selected problem from the research literature, applied it to graphs based on real images, conducted over twenty-five thousand experiments on 100 node random graphs of varying types (graphs with only zero-one links, weighted graphs, and graphs with node attributes), and benchmarked it against a classical graph matching technique - relaxation labeling. Comparable experimental success has never been reported before for any other graph matching algorithm.

## 2 Algorithm Development

### 2.1 Formulating the Objective

The graduated assignment algorithm will be described using the case of weighted graph matching. We define the problem of weighted graph matching in the following manner. Given two undirected graphs  $G$  and  $g$  which may be sparse and whose links may take values in  $R^1$ , find the match matrix  $M$  such that the following objective function is minimized.

$$E_{wg}(M) = -\frac{1}{2} \sum_{a=1}^A \sum_{i=1}^I \sum_{b=1}^A \sum_{j=1}^I M_{ai} M_{bj} C_{aibj} \quad (1)$$

subject to  $\forall a \sum_{i=1}^I M_{ai} \leq 1$ ,  $\forall i \sum_{a=1}^A M_{ai} \leq 1$ ,  $\forall ai M_{ai} \in \{0, 1\}$ .

Graphs  $G$  and  $g$  have  $A$  and  $I$  nodes respectively.  $\{C_{aibj}\}$  is defined by:

$$C_{aibj} = \begin{cases} 0 & \text{if either } G_{ab} \text{ or } g_{ij} \text{ is NULL} \\ c(G_{ab}, g_{ij}) & \text{otherwise,} \end{cases}$$

$\{G_{ab}\}$  and  $\{g_{ij}\}$  are the adjacency matrices of the graphs, whose elements may be in  $R^1$  or NULL. These matrices are symmetric with NULL elements along the diagonal. So,  $G_{ab}$  is the weight of the link between nodes  $a$  and  $b$  of graph  $G$ . The matrix  $M$  indicates which nodes in the two graphs match:

$$M_{ai} = \begin{cases} 1 & \text{if node } a \text{ in } G \text{ corresponds to node } i \text{ in } g \\ 0 & \text{otherwise,} \end{cases}$$

Note the two-way constraints on  $M$ . The function  $c(\cdot, \cdot)$  is a similarity measure between the links of the two graphs. It is positive if two links are similar and negative if two links are dissimilar. By explicitly defining  $C$  to be 0 when a link is missing (NULL) we are ensuring that  $C$  will also be sparse when the graphs are sparse. The function  $c(\cdot, \cdot)$  must be chosen such that a value of zero is appropriate for missing data. In all our experiments  $c(\cdot, \cdot)$  was defined as:  $c(G_{ab}, g_{ij}) = 1 - 3|G_{ab} - g_{ij}|$ . The function  $c$  was so chosen, in order to yield an expected value of zero when the link weights are randomly selected from a uniform distribution in the interval  $[0, 1]$ . The expected value will be zero, because two points chosen

from a uniform distribution in the unit interval will be on average  $\frac{1}{3}$  units apart. Note that the use of the  $L_1$  norm makes  $c(\cdot, \cdot)$  robust in the statistical sense.

We transform our inequality constraints in (1) into equality constraints by introducing slack variables, a standard technique from linear programming. An extra row and column are added to the match matrix  $M$  to hold our slack variables. Following the treatment in [15, 17] we employ Lagrange multipliers and an  $x \log x$  barrier function to enforce the constraints, resulting in the following objective:

$$\begin{aligned} \hat{E}_{wg}(M, \mu, \nu) = & -\frac{1}{2} \sum_{a=1}^A \sum_{i=1}^I \sum_{b=1}^A \sum_{j=1}^I M_{ai} M_{bj} C_{aibj} \\ & + \frac{1}{\beta} \sum_{a=1}^{A+1} \sum_{i=1}^{I+1} M_{ai} (\log M_{ai} - 1) \\ & + \sum_{a=1}^A \mu_a \left( \sum_{i=1}^{I+1} M_{ai} - 1 \right) + \sum_{i=1}^I \nu_i \left( \sum_{a=1}^{A+1} M_{ai} - 1 \right) \quad (2) \end{aligned}$$

In the above we are looking for a saddle point by minimizing with respect to  $M$  and maximizing with respect to  $\mu$  and  $\nu$ , the Lagrange multipliers.

The above objectives have been modified to handle ARGs as well [10].

Note the graph matching formulation in (1), with the inclusion of the robust, sparse distance measure between links, is new. Previous energy functions, as in [14], even when modified with slacks to handle missing or extra nodes, perform very badly in comparison to the algorithm presented here when matching graphs of unequal sizes.

### 2.2 The Algorithm

In graph matching we have a two-way constraint: A node in graph  $G$  must correspond to only one node in graph  $g$  and vice versa. These two constraints can be satisfied using a result from [18]. In [18] it is proven that any square matrix whose elements are all positive will converge to a doubly stochastic matrix just by the iterative process of alternatively normalizing the rows and columns. (A doubly stochastic matrix is a matrix whose elements are all positive and whose rows and columns all add up to one—it may roughly be thought of as the continuous analog of a permutation matrix).

Figure 1 depicts the algorithm used to minimize the objective in (2). Notice the contrast between softassign and softmax. In softmax, a one-way constraint is strictly enforced by normalizing over a vector.

The softassign simply employs Sinkhorn's technique within a deterministic annealing context. In [11], the softassign was used to solve the assignment problem, i.e. the minimization of:  $-\sum_{a=1}^A \sum_{i=1}^I M_{ai} Q_{ai}$ . For the quadratic assignment problem, being solved here, by doing a simple regrouping of terms in (1),  $-\frac{1}{2} \sum_{a=1}^A \sum_{i=1}^I M_{ai} (\sum_{b=1}^A \sum_{j=1}^I M_{bj} C_{aibj})$  and using the values of  $M$  from the previous iteration, we can at each iteration produce a new

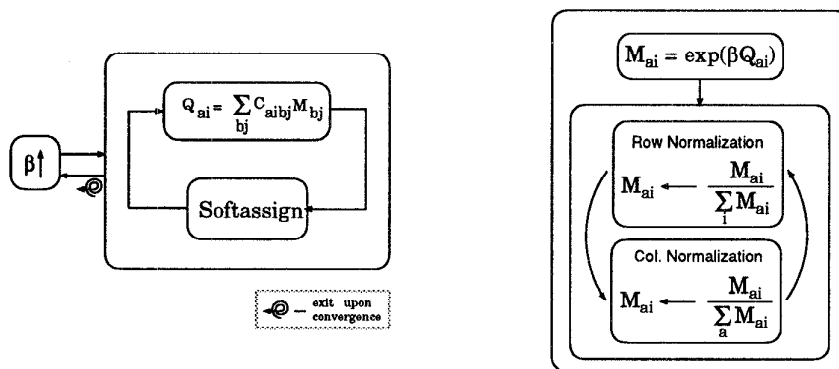


Figure 1: Left: Graduated assignment graph matching architecture.  $C_{ai bj}$  is a sparse distance measure between the two graphs and  $M_{ai}$  is the match matrix. Right: Softassign.

assignment problem for which the softassign then returns a doubly stochastic matrix. As the temperature is lowered a series of assignment problems are generated, along with the corresponding doubly stochastic matrices returned by each softassign, until a permutation matrix is reached. Hence the name graduated assignment.

The dynamics in Figure 1 may be obtained by evaluating the saddle points of the objective in (2). Sinkhorn's method finds the saddle points for the Lagrange parameters.

Note the algorithm actually operates on non-square match matrices, with an extra row and column of slack variables to handle missing and extra nodes. This does not result in any major complications in the above.

### 3 Experimental Results

Experiments conducted include: i) Repeating an experiment with attributed relational graphs first constructed by Eshera and Fu [3] who used a branch and bound method to perform the matching. ii) Hand designing attributed relational graphs from real images and matching them. iii) Generating twenty-five thousand random graphs of different types (unweighted graphs, weighted graphs and weighted graphs with binary attributes) and testing them under varying conditions of noise. iv) Running benchmarks against a relaxation labeling algorithm and an algorithm employing Potts glass dynamics. A sample follows; see [10] for the rest of the experiments.

#### 3.1 Graphs from Images

In one experiment, we hand designed graphs from two real images, Figure 2 and Figure 3, representing a model, and representing the scene in which we wish to locate the model, respectively. We assumed a low-level image processing sub-system capable of edge detection and curve grouping. Notice the large difference in scale, the occlusion, the large amount of clutter in the scene and even a small perspective difference. Three curves were created for the model and seventeen curves for the scene (curves are not shown). Four types of features were then marked for points on these curves, corresponding to whether they were points on



Figure 2: Image of scotch tape roll with features hand labeled

straight lines, convex curves, non-convex curves or discontinuities (break points) such as the end of the curve or at an inflection point. 44 (model) and 110 (scene) feature points were produced in the two images.

Attributed relational graphs of 44 and 110 nodes each were created from these sets of features. Each node had four binary valued attribute types, corresponding to straight line, convex curve, non-convex curve or break point features. Eight different link types between nodes within a graph were created. The first link type was binary valued and set to 1 between any two nodes corresponding to feature points on the same curve, NULL otherwise. The remaining link types were dependent on various parameters, such as the normalized distance between feature points within the same curve, the tangent angles to feature points on the curve, and the relationship between the curves the feature points lied on. The algorithm returned a match matrix with an assignment between the two graphs that was essentially correct except for

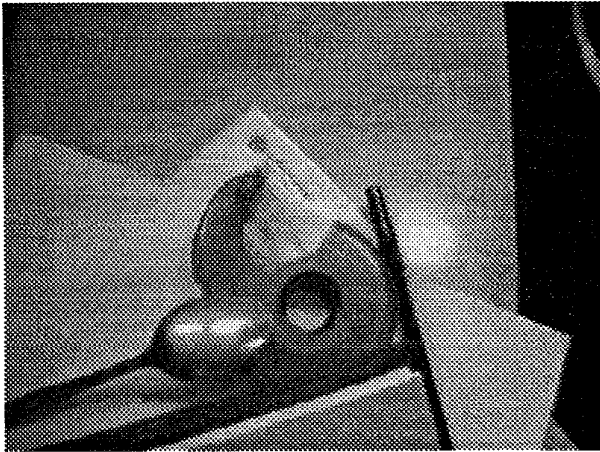


Figure 3: Image of a table top scene with features hand labeled

a small shift in the matching of the circular curve in the scotch tape in the model and the circular curve in the scotch tape in the scene. That is instead of matching  $1 \leftrightarrow 1, 2 \leftrightarrow 2, 3 \leftrightarrow 3, 4 \leftrightarrow 4...$  it matched up as  $1 \leftrightarrow 2, 2 \leftrightarrow 3, 3 \leftrightarrow 4, 4 \leftrightarrow 5, ...$

### 3.2 Benchmarks using Randomly Generated Graphs

Relaxation labeling [4] was chosen as a benchmark, because it is the only widely used graph matching method in the computer vision community, of low enough order computational complexity, to handle the size graphs we are examining. Branch and bound techniques, in contrast, are very slow and only suitable to either very small ( $< 20$  nodes) or very problem specific graphs. For example, the technique in [3] is still of approximately  $O(l^3m^2)$  complexity (where  $l$  and  $m$  are the number of links in the two graphs), even after the search space is pruned, and would run many orders of magnitude slower than our  $O(lm)$  graduated assigned algorithm on the 1000 link graphs we are testing. We did however, as noted above, repeat a specific branch and bound graph matching experiment from the literature [3, 10]. In neural networks, computer vision, operations research, and theoretical computer science we have found no literature that reports experiments on algorithms that have produced comparable results on the size of the graphs we are working with. For example, using a neural network technique [7] obtained very poor results ( $< 25\%$  accuracy) on graph isomorphism with 75 node graphs of less than 30% connectivity. On a much harder 100 to 98 node subgraph isomorphism problem (much harder because the algorithm included slack variables to handle potentially missing or extra nodes) the graduated assignment algorithm was almost 100% accurate on all graphs of greater than 4% connectivity. (This experiment is not shown, a harder 100 to 90 node experiment is shown instead in Figure 4, for which the results are still excellent.) Other traditional computer science techniques,

such as linear programming also fair very poorly in comparison [19].

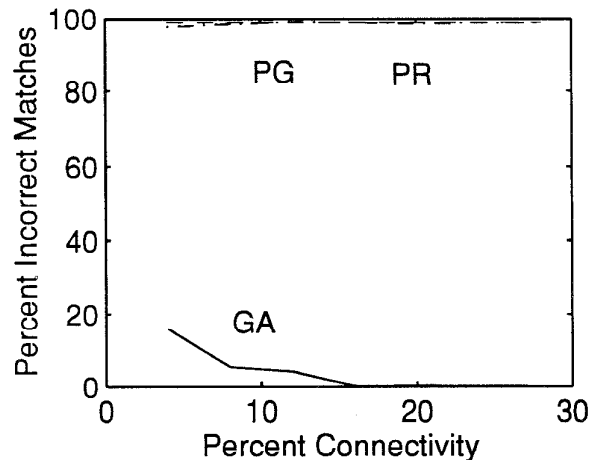


Figure 4: Comparisons between graduated assignment (GA), relaxation labeling (PR), and Potts glass (PG) **Subgraph isomorphism**. 90 node graphs of varying connectivity run against 100 node graphs. GA - 700 trials. PR - 70 trials. PG - 700 trials.

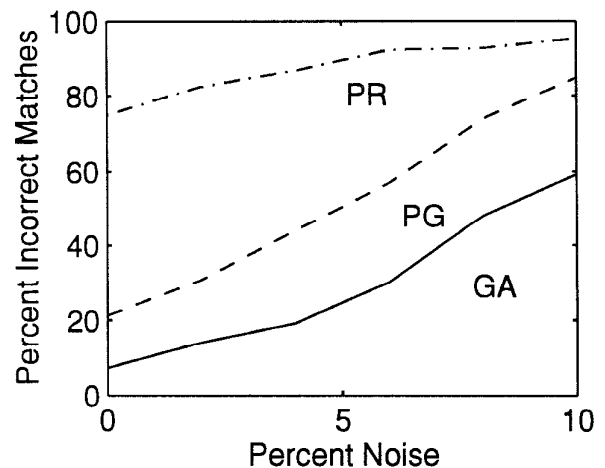


Figure 5: Comparisons between graduated assignment (GA), relaxation labeling (PR), and Potts glass (PG) **Weighted graph matching**. 60 node graphs, 15% connectivity, 5% deleted links, 5% spurious links, run against 100 node graphs. GA - 600 trials. PR - 60 trials. PG - 600 trials.

The second benchmark was conducted against an algorithm which used the Potts glass dynamics [16] to minimize the objective in Section 2.1 with an additional penalty term to enforce one constraint. Note this second algorithm is also original, since the formulation in Section 2.1 is new. It was designed to measure the performance of the softassign against the

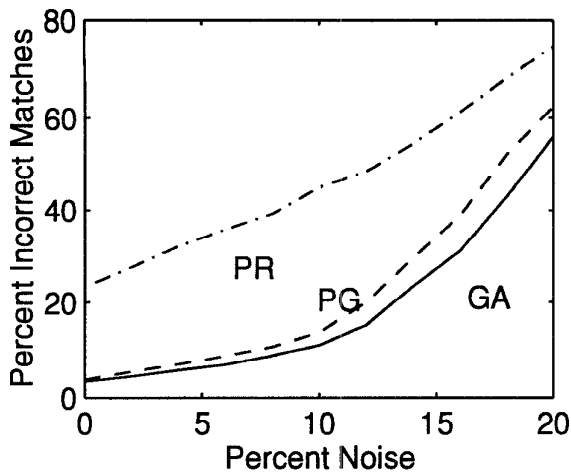


Figure 6: Comparisons between graduated assignment (GA), relaxation labeling (PR), and Potts glass (PG) **Attributed relational graph matching**. 40 node graphs, 10% connectivity, 3 binary attributes, 5% deleted links, 5% spurious links, 5% mislabeled attributes, run against 100 node graphs. GA - 1100 trials. PR - 110 trials. PG - 1100 trials.

softmax with a penalty term, which is one facet of the improvements offered by the new algorithm.

Figures 4, 5 and 6 display the results of these experiments on subgraph isomorphism, weighted graph matching, and attributed relational graph matching. In subgraph isomorphism, the x-axis displays the percent connectivity of the random graphs generated. In the other experiments the x-axis displays the percent noise added to the link weights in the randomly generated graphs. Link weights were randomly chosen from a uniform distribution in the interval  $[0, 1]$ . Then uniform noise was added to the link weights in the intervals  $[0, 0.1]$  and  $[0, 0.2]$  for the weighted graphs and the ARGs respectively. As noted in Figures 4, 5 and 6 various size graphs were matched and other sources of noise were included. Running times of the graduated assignment and Potts glass algorithms in these experiments range between 20 seconds and 2 minutes on a SGI Indigo workstation except for some of the subgraph isomorphism experiments involving graphs of higher connectivity. The relaxation labeling algorithm was between 5 and 15 times slower.

Graduated assignment performs best by a wide margin.

#### 4 Conclusion

A fundamentally new algorithm for inexact graph matching, applicable to any type of graph has been developed. By combining neural network techniques from statistical physics with a new energy function that explicitly encodes for sparsity large improvements in accuracy and speed are achieved over conventional techniques. A recently developed technique for constraint satisfaction, softassign, removes the need for penalty terms or Lagrange multipliers within the non-

linear objective used to formulate the problem. The result is a remarkably simple, yet effective two-step iterative procedure for graph matching.

Powerful evidence has been provided of the algorithm's performance, including experimental evidence on a scale never before provided for any graph matching technique. We have demonstrated that it will work on a problem from the research literature [3, 10], applied it to graphs from real images, tested it on a wide variety of graphs under conditions of noise, and benchmarked it against a standard method. The method is universal—it is applicable to any type of graph. It has low order computational complexity ( $O(lm)$ ). And it is accurate—it will work on problems such as subgraph isomorphism which have proved difficult for non-linear methods. Graduated assignment graph matching holds enormous promise.

#### Acknowledgements

We would like to thank Eric Mjolsness for his encouragement and support. This work was partially supported by ARPA/ONR grant N00014-92-J-4048 and the Yale Neuroengineering and Neuroscience Center.

#### References

- [1] E. Mjolsness, G. Gindi, and P. Anandan, "Optimization in model matching and perceptual organization", *Neural Computation*, vol. 1, pp. 218-229, 1989.
- [2] K. S. Fu, "A step towards unification of syntactic and statistical pattern recognition", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 5, pp. 200-205, March 1983.
- [3] M. A. Eshera and K. S. Fu, "An image understanding system using attributed symbolic representation and inexact graph-matching", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 8, pp. 604-619, Sept. 1986.
- [4] A. Rosenfeld and A. Kak, *Digital Picture Processing*, vol. 2, Academic Press, Inc., Orlando, Fl., 1982.
- [5] R. Hummel and S. Zucker, "On the foundations of relaxation labeling processes", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 5, pp. 267-287, May 1983.
- [6] S. Z. Li, "Matching: invariant to translations, rotations and scale changes", *Pattern Recognition*, vol. 25, pp. 583-594, 1992.
- [7] P. D. Simic, "Constrained nets for graph matching and other quadratic assignment problems", *Neural Computation*, vol. 3, pp. 268-281, 1991.
- [8] S.-S. Yu and W.-H. Tsai, "Relaxation by the Hopfield neural network", *Pattern Recognition*, vol. 25, pp. 197-208, Feb. 1992.

- [9] T.-W. Chen and W.-C. Lin, "A neural network approach to CSG-based 3-D object recognition", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, pp. 719-725, July 1994.
- [10] S. Gold, *Matching and Learning Structural and Spatial Representations with Neural Networks*, PhD thesis, Yale University, 1995.
- [11] J. J. Kosowsky and A. L. Yuille, "The invisible hand algorithm: Solving the assignment problem with statistical physics", *Neural Networks*, vol. 7, pp. 477-490, 1994.
- [12] S. Gold, E. Mjolsness, and A. Rangarajan, "Clustering with a domain specific distance measure", in J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pp. 96-103. Morgan Kaufmann, San Francisco, CA, 1994.
- [13] S. Gold, C. P. Lu, A. Rangarajan, Suguna Pappu, and E. Mjolsness, "New algorithms for 2-D and 3-D point matching: pose estimation and correspondence", in G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pp. 957-964. MIT Press, Cambridge, MA, 1995.
- [14] S. Gold, A. Rangarajan, and E. Mjolsness, "Learning with preknowledge: clustering with point and graph matching distance measures", *Neural Computation*, vol. 8, pp. 787-804, 1996.
- [15] A. Rangarajan, S. Gold, and E. Mjolsness, "A novel optimizing network architecture with applications", *Neural Computation*, vol. 8, 1996, (in press).
- [16] C. Peterson and B. Soderberg, "A new method for mapping optimization problems onto neural networks", *Intl. Journal of Neural Systems*, vol. 1, pp. 3-22, 1989.
- [17] A. L. Yuille and J. J. Kosowsky, "Statistical physics algorithms that converge", *Neural Computation*, vol. 6, pp. 341-356, May 1994.
- [18] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices", *Ann. Math. Statist.*, vol. 35, pp. 876-879, 1964.
- [19] H. A. Almohamad and S. O. Duffuaa, "A linear programming approach for the weighted graph matching problem", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, pp. 522-525, May 1993.