A Lagrangian Relaxation Network for Graph Matching

Anand Rangarajan
Eric Mjolsness

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

A Lagrangian Relaxation Network for Graph Matching

Anand Rangarajan[†] and Eric Mjolsness[‡]

[†]Departments of Computer Science and Diagnostic Radiology, Yale University

and the Yale Center for Theoretical and Applied Neuroscience (CTAN)

E-mail: **rangarajan-anand@cs.yale.edu**

[‡]Department of Computer Science and Engineering

University of California San Diego (UCSD)

E-mail: **emj@cs.ucsd.edu**

Abstract

A Lagrangian relaxation network for graph matching is presented. The problem is formulated as follows: given graphs G and g, find a *permutation matrix M* that brings the two sets of vertices into correspondence. Permutation matrix constraints are formulated in the framework of *deterministic annealing*. Our approach is similar to a *Lagrangian decomposition* approach in that the row and column constraints are satisfied separately with Lagrange multipliers used to equate the two "solutions." Lagrange parameters also express the graph matching constraint. Due to the unavoidable symmetries involved in graph matching (resulting in multiple global minima), we add a *self-amplification* term in order to obtain a permutation matrix. A Lagrangian relaxation network is obtained. The network performs minimization with respect to the Lagrange parameters and maximization with respect to the match matrix variables. Simulation results are shown on 100 node random graphs and for a wide range of connectivities.

Index Terms: Deterministic annealing, permutation matrix, Lagrangian relaxation, graph matching, self amplification, symmetry-breaking, merit function, winner-take-all, Potts glass, softmax.

1 Introduction

Graph matching is an important part of object recognition systems in computer vision. The problem of matching structural descriptions of an object to those of a model is posed as weighted

graph matching [1]-[18]. Good, approximate solutions to inexact, weighted graph matching are usually required. Neural network approaches to graph matching [4, 5, 6, 8, 9, 16, 19, 18] share the feature of other recent approaches to graph matching [20, 21, 12, 22, 15] in that they are not restricted to looking for the right isomorphism. Instead, a measure of distance between the two graphs [23] is minimized. The focus shifts from brittle, symbolic, subgraph isomorphism distance metrics [11, 24] to an "energy" or *distance measure* between graphs which is minimized by the correct correspondence of the vertex labels of one graph with respect to the other. Subgraph isomorphism is a special case and is achieved when the distance is zero.

Exact graph isomorphism is an open problem. It is not known to be either NP-hard or have a polynomial time solution[1] [28, 29]. In contrast, subgraph isomorphism is NP-complete [28, 30]. In this paper, we are interested in exact graph isomorphism and matching as they seem to be problems intermediate in difficulty. Our approach handles isomorphism and matching in the same manner. We follow the *deterministic annealing* framework formulated in [31, 32]. This particular approach is useful since it readily generalizes to inexact, weighted graph matching. We set up our problem as follows: given graphs G and g, find a *permutation matrix M* that minimizes the distance between the two graphs. A permutation matrix is a zero-one matrix whose rows and columns sum to one. When the problem is made more difficult—subgraph isomorphism or inexact graph matching—the permutation matrix constraints get modified. The rows and columns can add up to one or zero. Permutation matrix constraints are formulated quite naturally in the framework of deterministic annealing. Essentially, the row or column constraints are *winner-take-alls* (WTAs). Our approach is similar to a *Lagrangian decomposition* approach [33, 34] in that the row and column constraints are satisfied separately and Lagrange multipliers are used to equate the two "solutions." A fixpoint preserving transformation [35] is applied to the graph matching distance, allowing us to express the combination of the graph matching distance constraint and the permutation matrix constraint using terms that are linear in the two match matrix "solutions" $M^{(1)}$ and $M^{(2)}$. Due to the unavoidable symmetries involved in graph matching (resulting in

---

[1]There is significant recent evidence indicating that graph isomorphism may not be NP-hard [25, 26, 27].

multiple global minima), we add a *self-amplification* term [36] in order to obtain a permutation matrix. The self-amplification term is similar to the hysteretic annealing performed in [37, 38] and is suited to the deterministic annealing framework employed here. Unlike hysteretic annealing, the self-amplification parameter is held fixed. The self-amplification term is reversed via another fixpoint preserving transformation. With the application of the fixpoint preserving transformations to the distance measure and self-amplification terms, the energy function becomes linear in the two match matrices. Elimination of the two match matrices considerably simplifies the energy function. Since the fixpoint preserving transformations reverse the dynamics, a saddle point of the energy function is sought. The resulting network is tested on the graph isomorphism problem with 100 node random, undirected graphs and on the weighted graph matching problem with 100 node weighted, random graphs and uniform noise added to the connections.

Section 2 sets up the deterministic annealing Lagrangian relaxation approach for graph matching. A description of the Lagrangian relaxation algorithm follows in Section 3. Experimental results for graph isomorphism and matching are presented in Section 4. Relationships to other approaches are presented in Section 5.

2 Graph matching via deterministic annealing

The graph matching problem is formulated in terms of the adjacency matrices of the two graphs. Given the adjacency matrices G_{ab} and g_{ij} of two graphs $G(V, E)$ and $g(v, e)$ respectively, the problem is to find the permutation match matrix M such that the distance between the two graphs is minimized. The adjacency matrices G_{ab} and g_{ij} of the two undirected graphs are represented as symmetric, sparse matrices with zero diagonal entries. The problem is stated as follows:

$$\min_{M} \sum_{ai} \left(\sum_{b} G_{ab} M_{bi} - \sum_{j} M_{aj} g_{ji} \right)^2 \tag{1}$$

$$\text{subject to} \quad \sum_{a} M_{ai} = 1, \quad \sum_{i} M_{ai} = 1$$

with $M_{ai} \in \{0, 1\}$. An example of isomorphic graphs is shown in Figure 1. Graph matching is not restricted to isomorphism. Equation (1) describes a distance measure between the two graphs G
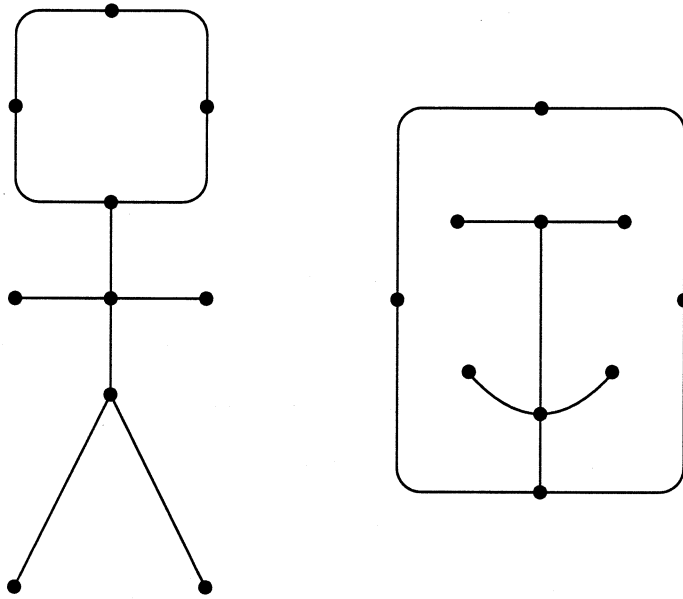
Figure 1: An example of isomorphic graphs: The person and the face are seen to be isomorphic. A link between any two nodes in one graph has a counterpart in the other graph. There's a redundancy in that the mirror image of the face (person) is also isomorphic to the person (face).

and g. For isomorphism, (1) becomes $GM = Mg$ or $g = M^T GM$ and the distance is zero, but for matching, the distance is non-zero expressing the fact that g is a noisy version of G. A distorted graph g is obtained by adding independent noise to all the edges of G.

The distance measure in (1) contains the rectangle rule of graph matching. To see this, examine the middle term of the distance measure $(-G_{ab}M_{bj}g_{ji}M_{ai})$. Restricting the edges to be either zero or one, we see that all four elements in this chain have to be "on" in order to secure a match. When this happens, vertices "a" and "b" in G are matched to vertices "j" and "i" in g respectively and edges exist between "a" and "b" and "j" and "i" in G and g respectively. This is depicted in Figure 2. The rectangle rule plays an important role in subgraph isomorphism; here, one looks for the match matrix yielding the largest number of rectangles [6].

We now turn our attention to the constraints. The distance measure contains the match matrix M which has to satisfy permutation matrix constraints. Besides the usual integrality constraint on the match variables ($M_{ai} \in \{0, 1\}$), these constraints include row and column WTAs ($\sum_a M_{ai} = 1$ and $\sum_i M_{ai} = 1$). The constraints on M for graph matching are the same as the constraints for
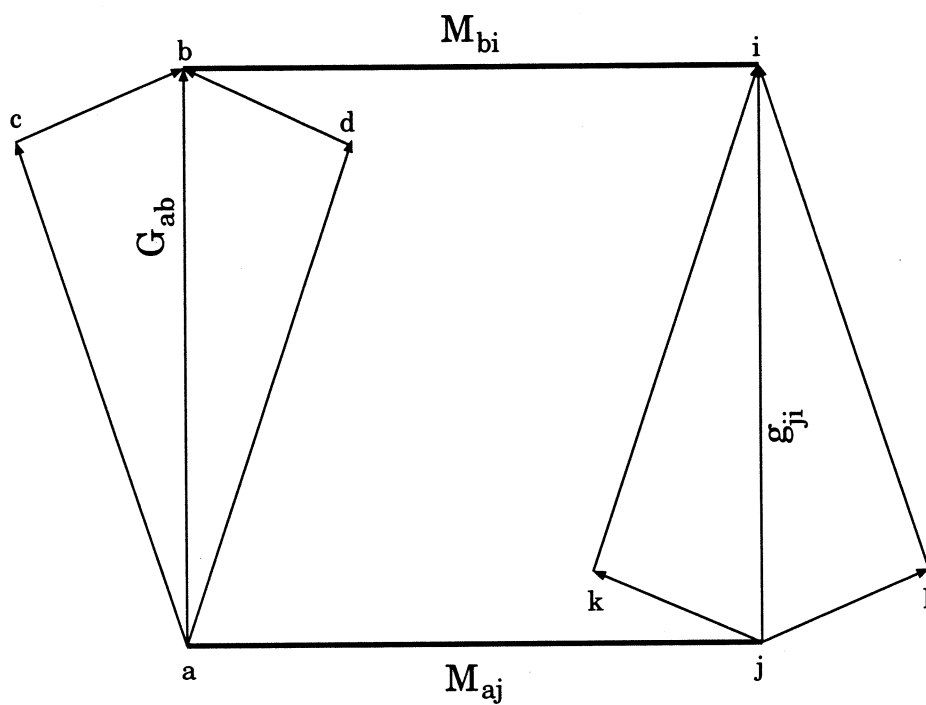
Figure 2: The rectangle rule for graph isomorphism: vertices "a" and "b" in G are matched to vertices "j" and "i" in g respectively and edges exists between "a" and "b" and "j" and "i" in G and g respectively. As a match to "i", "b" will be preferred to "c" or "d" because it has the same degree (3). The same holds for a match between "a" and "j". A rectangle is constructed from the cycle of vertices $a \xrightarrow{G_{ab}} b \xrightarrow{M_{bi}} i \xrightarrow{g_{ji}} j \xrightarrow{M_{aj}} a$.

the traveling salesman problem (TSP).

Neural net approaches (beginnning with the celebrated Hopfield-Tank network [39]) have typically expressed these constraints via penalty functions. However, contrary to the standard penalty function approach [40], the penalty function parameters are seldom varied. Regardless of the manner of constraint satisfaction, the original discrete problem is converted into a nonlinear optimization problem, with the match matrix assuming values inside the unit hypercube. The permutation matrix constraints get transformed into *doubly stochastic matrix* constraints ($\sum_a M_{ai} = 1$, $\sum_i M_{ai} = 1$ and $M_{ai} \in [0, 1]$). An $x \log(x) + (1 - x) \log(1 - x)$ barrier function constrains the match matrix to be inside the unit hypercube. Gradually changing the barrier function parameter is identical to *mean-field annealing* (MFA) [41, 42].

Beginning with Kanter and Sompolinsky [43], several authors [44, 31, 45, 46, 47] have noted that a more compact representation of the WTA constraint is available via what is known as a Potts glass. The Potts glass approach is also a mean field method. This method utilizes the fact that there are only N configurations (and not 2^N) satisfying the WTA constraint ($\sum_i s_i = 1$, $s_i \in \{0, 1\}$). The WTA constraint is kept satisfied in a soft manner by having the mean field variables sum to one ($\sum_i s_i = 1$) and is termed *softmax* [48]. However, the two WTA constraints required by the doubly stochastic matrix cannot be kept simultaneously satisfied.

The basic idea in this paper is to start with two match matrices $M_{ai}^{(1)}$ and $M_{ai}^{(2)}$ which have the respective properties

$$\sum_a M_{ai}^{(1)} = 1, \text{ and } \sum_i M_{ai}^{(2)} = 1 \tag{2}$$

always satisfied. Two separate softmax Potts glasses are used. Assuming that the claim in (2) can be established, the new objective is

$$\min_{M^{(1)}, M^{(2)}} D(M^{(1)}, M^{(2)}) = \sum_{ai} \left(\sum_b G_{ab} M_{bi}^{(1)} - \sum_j M_{aj}^{(2)} g_{ji} \right)^2 \tag{3}$$

$$\text{subject to } M_{ai}^{(1)} = M_{ai}^{(2)}. \tag{4}$$

The constraint in (4) is set up using Lagrange parameters λ. An energy function is constructed with four ingredients: (i) the graph matching distance measure, (ii) the softmaxes on $M^{(1)}$ and

6

$M^{(2)}$, (iii) equating the two "solutions" via the Lagrange parameters λ and (iv) a self-amplification term $(\frac{\gamma}{2}\sum_{ai} M_{ai}(1 - M_{ai}))$ with a parameter γ.

First, we introduce the machinery of Gibbs distributions [49]. A Gibbs probability distribution is specified by an energy function $E(V)$:

$$\Pr(\mathbf{V} = V) = \frac{1}{Z(\beta)} \exp\left(-\beta E(V)\right) \tag{5}$$

where β is the inverse temperature to be used subsequently in the deterministic annealing procedure. In (5), $Z(\beta)$ is the *partition function* which is a sum over all configurations of V $(Z(\beta) = \sum_V \exp\left[-\beta E(V)\right])$. Rewriting (5), we get

$$-\frac{1}{\beta}\log\left(Z(\beta)\right) = E(V) - \frac{1}{\beta}\log\left(\Pr(\mathbf{V} = V)\right). \tag{6}$$

Taking expectations (6) and noting that the left side remains the same, we get

$$-\frac{1}{\beta}\log\left(Z(\beta)\right) = \sum_V \Pr(\mathbf{V} = V)E(V) - \frac{1}{\beta}\sum_V \Pr(\mathbf{V} = V)\log\left(\Pr(\mathbf{V} = V)\right).$$

This is a well known statistical physics identity [50] and immediately leads to the definition of the *free energy*:

$$F(\beta) \stackrel{\text{def}}{=} -\frac{1}{\beta}\log\left(Z(\beta)\right) = \mathcal{E}_\beta\left(E(V)\right) - \frac{1}{\beta}S(\beta) \tag{7}$$

where $\mathcal{E}$ is the expectation operator and $S(\beta)$ is the entropy. As the temperature is reduced (and fewer configurations become likely) the entropy decreases to zero. Also, the expected value of the energy approaches the minimum value of the energy function. Deterministic annealing minimizes the free energy F instead of $E(V)$. The free energy (at low β) is a smooth approximation to the original, non-convex energy [51, 52, 42] and approaches the minimum of $E(V)$ as β tends to infinity.

Unfortunately, the free energy involves the logarithm of the partition function which is intractable. For a more tractable computation, the partition function (at each temperature) is approximated via the saddle-point approximation [47, 31, 53]. The partition function for graph

matching is written as follows:

$$Z(\beta) = \sum_{\left\{\begin{array}{c} V^{(1)}, V^{(2)}| \\ \sum_a V_{ai}^{(1)} = 1 \\ \sum_i V_{ai}^{(2)} = 1 \\ V_{ai}^{(1)} = V_{ai}^{(2)} \end{array}\right\}} \exp\left(-\beta\left[\frac{1}{2}\sum_{ai}\left(\sum_b G_{ab}V_{bi}^{(1)} - \sum_j V_{aj}^{(2)}g_{ji}\right)^2\right.\right.$$

$$\left.\left. +\frac{\gamma}{2}\sum_{ai}\frac{V_{ai}^{(1)}+V_{ai}^{(2)}}{2}\left(1-\frac{V_{ai}^{(1)}+V_{ai}^{(2)}}{2}\right)\right]\right). \tag{8}$$

In Appendix A, we present the derivation of the free energy arising from an application of the saddle-point approximation to the above partition function. The free energy obtained is

$$F(U^{(1)}, M^{(1)}, U^{(2)}, M^{(2)}, \lambda, \beta) = \frac{1}{2}\sum_{ai}\left(\sum_b G_{ab}M_{bi}^{(1)} - \sum_j M_{aj}^{(2)}g_{ji}\right)^2 \tag{9}$$

$$+\sum_{ai}\left[\lambda_{ai}\left(M_{ai}^{(1)} - M_{ai}^{(2)}\right) + \frac{\gamma}{2}\left(\frac{M_{ai}^{(1)}+M_{ai}^{(2)}}{2}\left[1 - \frac{M_{ai}^{(1)}+M_{ai}^2}{2}\right]\right)\right]$$

$$+\frac{1}{\beta}\left(\sum_{ai}U_{ai}^{(1)}M_{ai}^{(1)} - \sum_i \log\sum_a \exp\left\{U_{ai}^{(1)}\right\}\right) + \frac{1}{\beta}\left(\sum_{ai}U_{ai}^{(2)}M_{ai}^{(2)} - \sum_a \log\sum_i \exp\left\{U_{ai}^{(2)}\right\}\right).$$

In (9), $U^{(1)}$ and $U^{(2)}$ are dummy variables involved in the implementation of the two WTA constraints. $M^{(1)}$ and $M^{(2)}$ are continuous-valued versions of the discrete match matrices $V^{(1)}$ and $V^{(2)}$. As the temperature is reduced to zero, $M^{(1)}$ and $M^{(2)}$ converge to the corners of the unit hypercube while respecting the softmax constraints.

The free energy in (9) can be considerably simplified. We apply a fixed–point preserving transformation [35] which transforms

$$X^2 \rightarrow 2X\tau - \tau^2$$

where it is seen that

$$\max_{\tau} 2X\tau - \tau^2 = X^2.$$

Both the distance and the self-amplification terms are modified using this algebraic transformation. The distance becomes

$$D(M,\mu) = \sum_{ai}\mu_{ai}\left[\sum_b G_{ab}M_{bi} - \sum_j M_{aj}g_{ji}\right] - \frac{1}{2}\sum_{ai}\mu_{ai}^2. \tag{10}$$

With the application of the above algebraic transformation, the distance measure becomes linear in the match matrix and the dynamics are reversed. We ascend on μ as well as the Lagrange parameters λ. Introducing a new parameter μ does not significantly change the number of variables ($2N^2$) on which ascent is performed.

The self-amplification term becomes

$$-\frac{\gamma}{2} \sum_{ai} \left(\frac{M_{ai}^{(1)} + M_{ai}^2}{2} \right)^2 \rightarrow -\gamma \sum_{ai} \frac{M_{ai}^{(1)} + M_{ai}^{(2)}}{2} \sigma_{ai} + \frac{\gamma}{2} \sum_{ai} \sigma_{ai}^2. \tag{11}$$

The distance measure and the self-amplification term are transformed so that the energy function becomes linear in $M^{(1)}$ and $M^{(2)}$. Also, we have separate control via μ and σ over the amount of ascent/descent on the distance measure and self-amplification terms respectively.

The fixed points of the free energy w.r.t $U^{(1)}$, $U^{(2)}$, $M^{(1)}$ and $M^{(2)}$ are evaluated by direct differentiation. These variables are eliminated from F and after dropping the terms $\sum_{ai} M_{ai}^{(1)}$ and $\sum_{ai} M_{ai}^{(2)}$ and reversing the sign of F, we get

$$\begin{aligned} F(\mu, \lambda, \sigma) &= \frac{1}{2} \sum_{ai} \mu_{ai}^2 - \frac{\gamma}{2} \sum_{ai} \sigma_{ai}^2 \\ &+ \frac{1}{\beta} \sum_i \log \sum_a \exp \left(-\beta \left[\sum_b G_{ab} \mu_{bi} + \lambda_{ai} - \frac{\gamma}{2} \sigma_{ai} \right] \right) \\ &+ \frac{1}{\beta} \sum_a \log \sum_i \exp \left(+\beta \left[\sum_j g_{ij} \mu_{aj} + \lambda_{ai} + \frac{\gamma}{2} \sigma_{ai} \right] \right). \end{aligned} \tag{12}$$

This is the final form of the free energy. Note that the match variables $M^{(1)}$ and $M^{(2)}$ have been eliminated. The energy function in (12) is to be minimized w.r.t μ and λ and maximized w.r.t σ. The dynamics on (μ, λ) and σ can be separated.

When the fixed points are evaluated (partial derivatives with respect to μ, λ and σ are set to zero), we get

$$\begin{aligned} \frac{\partial F}{\partial \mu_{ai}} &= 0 \Rightarrow \mu_{ai} = \sum_b G_{ab} M_{bi}^{(1)} - \sum_j M_{aj}^{(2)} g_{ji}, \\ \frac{\partial F}{\partial \lambda_{ai}} &= 0 \Rightarrow M_{ai}^{(1)} = M_{ai}^{(2)}, \text{ and} \\ \frac{\partial F}{\partial \sigma_{ai}} &= 0 \Rightarrow \sigma_{ai} = \frac{M_{ai}^{(1)} + M_{ai}^{(2)}}{2}, \end{aligned}$$

where

$$M_{ai}^{(1)} = \frac{\exp \left(-\beta \left[\sum_c G_{ac} \mu_{ci} + \lambda_{ai} - \frac{\gamma}{2} \sigma_{ai} \right] \right)}{\sum_b \exp \left(-\beta \left[\sum_c G_{bc} \mu_{ci} + \lambda_{bi} - \frac{\gamma}{2} \sigma_{bi} \right] \right)}, \text{ and} \tag{13}$$

9

$$M_{ai}^{(2)} = \frac{\exp\left(\beta\left[\sum_k g_{ik}\mu_{ak} + \lambda_{ai} + \frac{\gamma}{2}\sigma_{ai}\right]\right)}{\sum_j \exp\left(\beta\left[\sum_k g_{jk}\mu_{ak} + \lambda_{aj} + \frac{\gamma}{2}\sigma_{aj}\right]\right)} \tag{14}$$

are the two softmaxes corresponding to $\sum_a M_{ai}^{(1)} = 1$ and $\sum_i M_{ai}^{(2)} = 1$ respectively. At its fixed point, the reversed parameter μ is equal to the distance measure. For graph isomorphism, μ converges to zero, and hence, the μ^2 term can be dropped from the free energy. The parameter μ then becomes a Lagrange parameter satisfying the graph isomorphism constraint—graph matching distance equals zero.

We have obtained a free energy which expresses the graph matching and permutation matrix constraints. In the next section, we describe the algorithm that extremizes the above energy function.

3   The Lagrangian relaxation algorithm

The energy function is minimized w.r.t μ and λ and maximized w.r.t σ. In Appendix B, we show that the energy function in (12) is weakly convex w.r.t the reversed neurons μ and the Lagrange parameters λ. Hence, local minima do not occur during minimization. It still remains to find a saddle point of the energy function. Relaxation of the constraints via μ and λ and subsequent optimization w.r.t them is referred to as *Lagrangian relaxation* [54, 34]. While minimizing the energy function, the main objective is not to find the Lagrange parameters but to satisfy the constraints in (3) and (4). To this end, we use the *merit function* [40] rather than the energy function to determine the line search parameter during each iteration of a conjugate-gradient (CG) update. The merit function for our energy function is

$$\begin{aligned}
E_{\text{merit}}(\mu, \lambda) &= \frac{1}{2}\sum_{ai}\left[\left(\frac{\partial F}{\partial \mu_{ai}}\right)^2 + \left(\frac{\partial F}{\partial \lambda_{ai}}\right)^2\right] \\
&= \frac{1}{2}\sum_{ai}\left[\left(\mu_{ai} - \sum_b G_{ab}M_{bi}^{(1)} + \sum_j M_{aj}^{(2)}g_{ji}\right)^2 + \left(M_{ai}^{(1)} - M_{ai}^{(2)}\right)^2\right]. \tag{15}
\end{aligned}$$

Using the merit function to obtain the line search parameter allows us to escape the typically slow, oscillatory convergence of the Lagrange parameters [55]. The merit function in (15) is a

combination of the graph matching distance measure and doubly stochastic matrix constraint deviation. Reducing the merit function satisfies the twin requirements of distance minimization and constraint satisfaction. The $l^{(2)}$ norm of the constraint violation is obtained from the merit function ($l^{(2)}$ norm $= \frac{\sqrt{2E_{\mathrm{merit}}}}{N}$) and used to set convergence thresholds. Since the energy function is weakly convex, the merit function (and therefore the $l^{(2)}$ norm as well) are guaranteed not to increase for gradient descent dynamics [40].

While maximizing the energy function w.r.t σ, the objective function is very slowly and gradually increased (not maximized) at each temperature. But it is always minimized w.r.t μ and λ. This approach is used since graph matching distance minimization and constraint satisfaction are more crucial than self-amplification.

THE LAGRANGIAN RELAXATION ALGORITHM

I. INITIALIZE: Set $\beta = \beta^{(0)}$ and λ, μ, σ to random values.

II. DESCEND: Run a CG algorithm w.r.t (μ, λ) in the free energy (12) until $l^{(2)}$ norm $< l^{(2)}_{\mathrm{thr}}$. Use the merit function in (15) to determine the line search parameter.

III. ASCEND: Set $\sigma_{ai} = \frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2} +$ random noise. This performs self-amplification.

IV. ANNEAL: Increase β. $\beta^{(k+1)} = (1 - \delta)\beta^{(k)} + \delta\beta_{\mathrm{max}}$. Return to step II unless all M_{ai} have converged ($\forall ai, M_{ai}(1 - M_{ai}) < \kappa$).

Note from the description of the algorithm that while the energy function is minimized w.r.t the Lagrange parameters, it is merely increased (one iteration) w.r.t σ at each temperature.

4   Simulation results

All reported simulations were executed on Silicon Graphics Indigo workstations with R4000 processors.

We are first interested in testing our method on the isomorphism problem which requires that the distance between the two graphs be equal to zero. This is achieved by dropping the second term of (10) and treating μ as a Lagrange parameter instead of as a reversed "neuron" [35] arising from the fixpoint preserving transformation.

We ran the Lagrangian relaxation algorithm on 100 node undirected, random graphs ($N = 100$) with weights in $\{0, 1\}$ and for a wide range of connectivities. It is difficult to generate random graphs with the exact number of edges as specified by the desired connectivity [56]. Instead, we generated random graphs with the number of edges approximating the desired connectivity. The chosen connectivities are $\{0.001, 0.01, 0.02, 0.03, 0.04\}$ and from 0.05 to 0.5 in steps of 0.025. Sixty examples were generated at each connectivity giving a total of 1440 examples. The results are presented in Figures 3 and 4. In all experiments, the following parameters were used: $\delta = 0.01$, $\beta^{(0)} = 1.0$ and $\beta_{\max} = 100$. The self-amplification parameter γ was set to $\frac{N}{20} = 5$. For most connectivities, a number of permutation matrices map G onto g. These different solutions correspond to multiple global minima. The self-amplification term performs the useful function of *symmetry-breaking* [57, 58]. The match matrix traverses a series of bifurcations in starting from an almost uniform doubly stochastic matrix ($M_{ai} = \frac{1}{N} +$ random noise) and ending with the final permutation matrix. The bifurcations due to symmetry-breaking start occurring at $\beta = \frac{2N}{\gamma} = 40$ for fully connected 100 node graphs but much before that for sparsely connected graphs. The $l^{(2)}$ norm threshold ($l^{(2)}_{\mathrm{thr}}$) and κ were set to 0.02 and 0.001 respectively. Our method always gave a correct permutation matrix for all the examples except for three failures at an average connectivity percentage of 1%. From Figures 3 and 4, it is seen that the number of iterations and the relaxation time are large for both extremely small connectivities and for moderate to large connectivities. The method can also be tried for connectivities larger than 0.5 but for this problem it is easier to reverse the weights on all the edges. These results are clearly superior to the results obtained by Simić [19]. Simić's deterministic annealing network could not find the right isomorphism for many connectivities in 75 node random graphs.

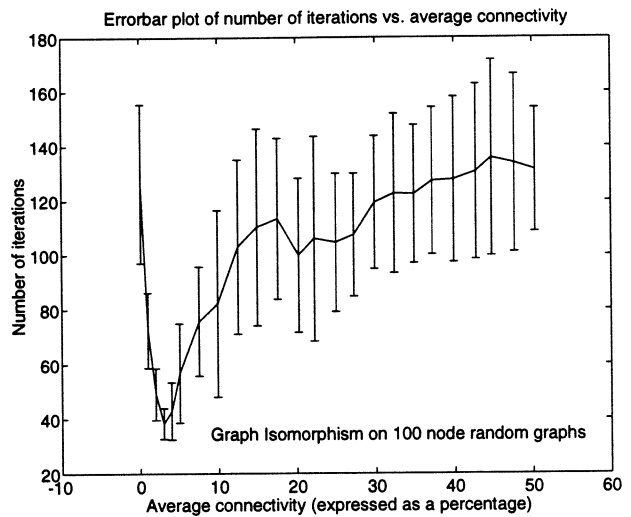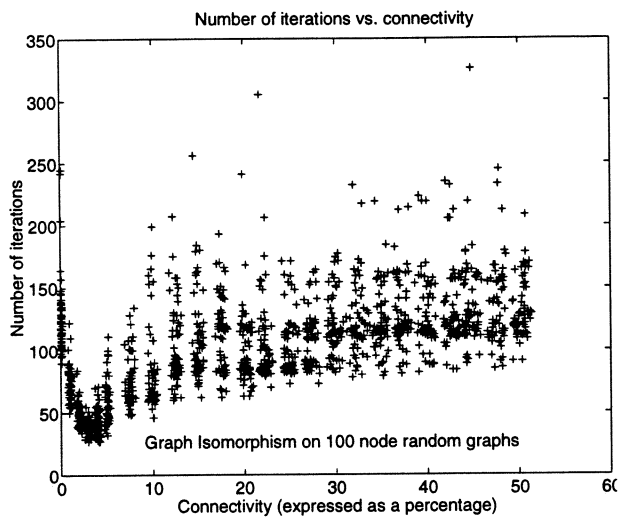Figure 5 shows the change in energy as the iterations proceed for the sixty examples of graph

Figure 3: Number of iterations vs. connectivity: Each "+" in the figure denotes the number of iterations taken for a 100 node graph isomorphism run. A total of 1440 graph isomorphism examples are presented. The interval $[0,5]\%$ was tested more heavily to obtain a better sampling of the decrease and then increase in the number of iterations. The apparent quantization of the iterations is not understood. The figure on the right is an errorbar plot of the data presented in the figure on the left. Note the decrease in the variance of the number of iterations in the interval $[0,5]\%$.
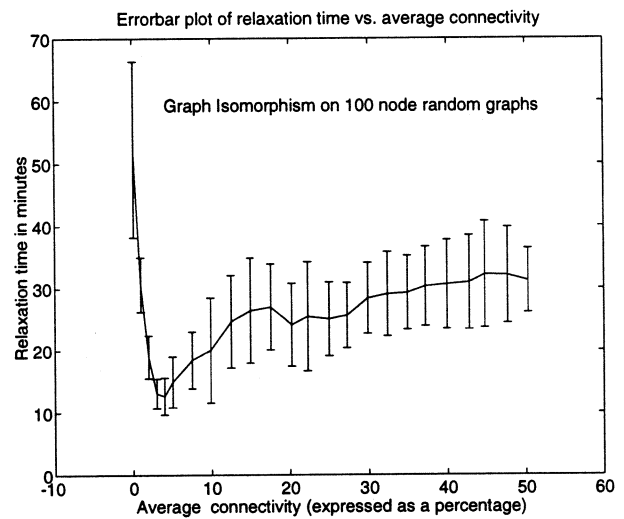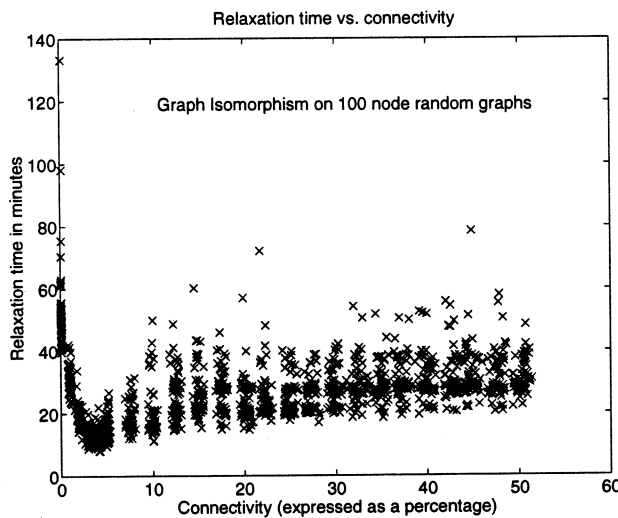

Figure 4: Relaxation time vs. connectivity: Each "x" in the figure denotes the wall clock time. The code was not optimized. The figure on the right is an errorbar plot of the data presented in the figure on the left.
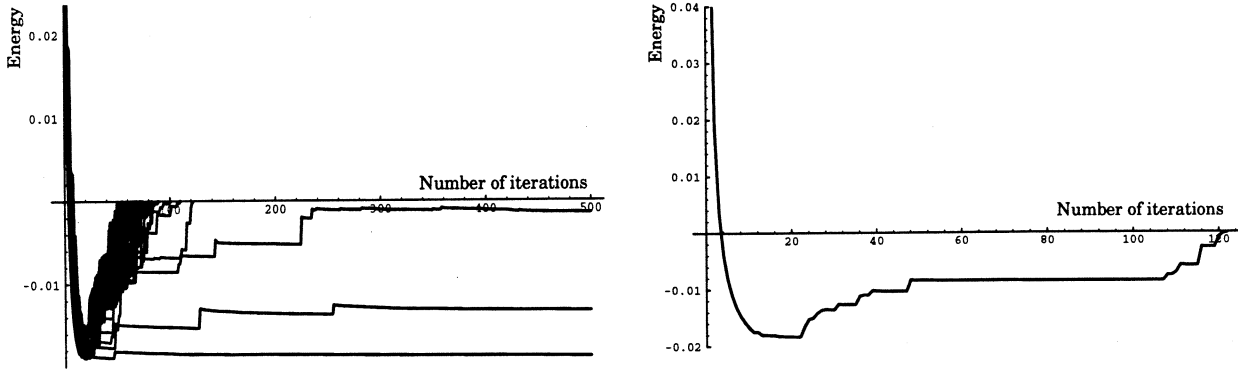
13

Figure 5: Energy vs. number of iterations: In the left figure, the approach to a saddle point is depicted for a connectivity of 1%. In all 60 cases, the energy decreases rapidly, goes through zero with the match matrix far from convergence and then slowly ascends. Three failures are indicated by flatlines. The energy does not make the ascent to zero. In the right figure, the same approach to a saddle point of the energy is depicted for one particular example.

isomorphism at a connectivity of 0.01. With the exception of the three failures (indicated by the flatlines), the pattern is the same. An isomorphism is obtained when the energy goes to zero and when all the match matrix variables have converged. This corresponds to a saddle point. The energy first decreases rapidly, passes through zero (with the match matrix very far from convergence) and then increases gradually approaching zero. The gradual progress towards integral solutions is seen in Figure 6 where the changes in the match matrix are depicted through thirty six temperatures. The black dots appearing for the first time in Figure 6(c) indicate convergence to integral solutions of the respective rows *and* columns. These stay in place and the remaining rows and columns gradually converge.

Next, we ran the Lagrangian relaxation algorithm on 100 node weighted graphs. The weights are chosen randomly in the interval $[0, 1]$. Uniform noise in the interval $[-\epsilon, \epsilon]$ is then added to all the edges. The noise variance corresponding to uniform noise is $\frac{\epsilon^2}{3}$. We chose nine noise variances ranging roughly from 0 to 1 with sixty examples at each value giving us a total of 540 examples. The results are presented in Figure 7. In all experiments, we used the following parameters: $\delta = 0.01$, $\beta^{(0)} = 1.0$ and $\beta_{\max} = 100$. The self-amplification parameter γ was set to $\frac{N}{10} = 10$.
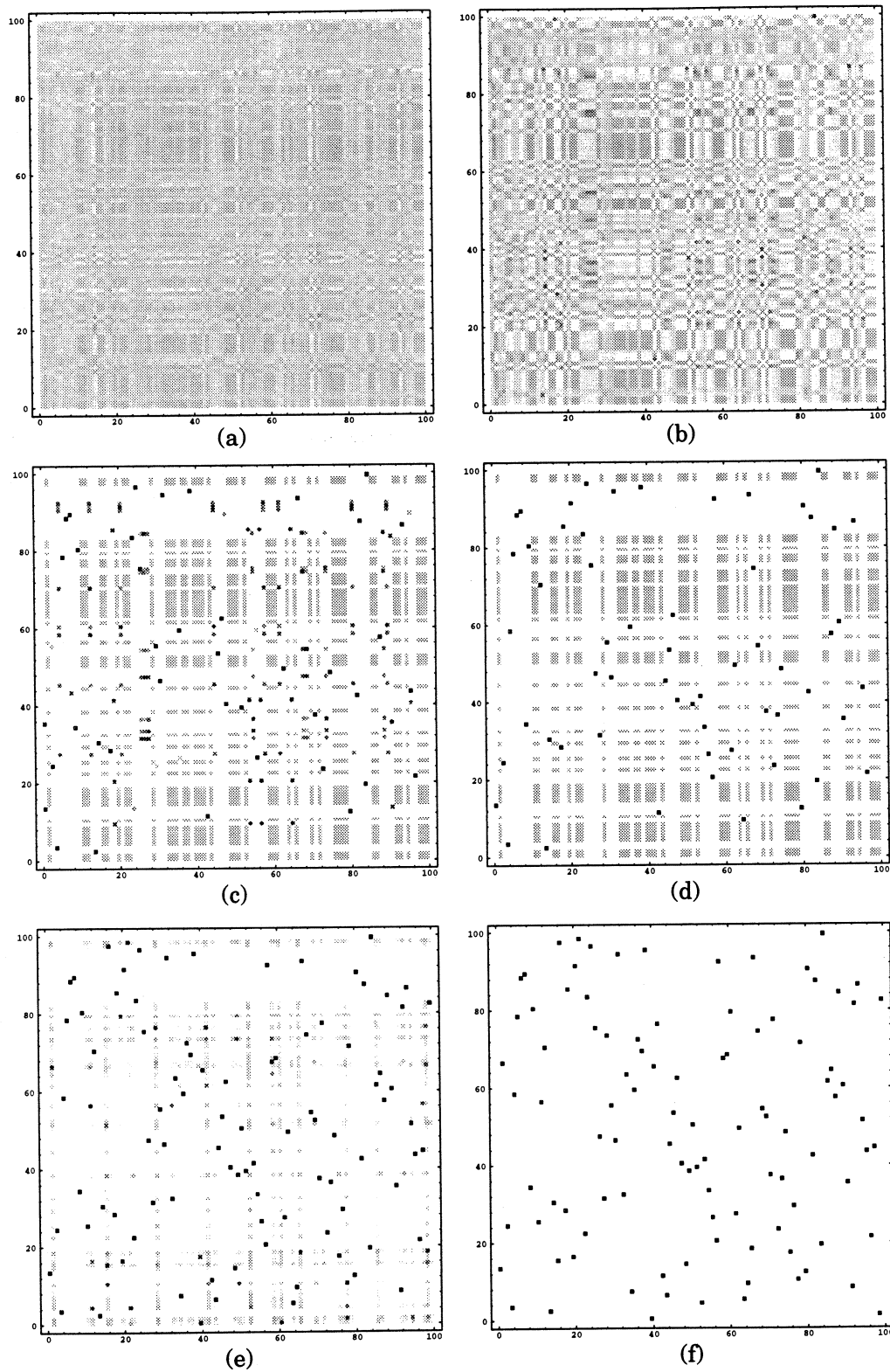
Figure 6: Match matrix at the (a) first, (b) ninth, (c) sixteenth, (d) nineteenth, (e) thirty third, and (f) thirty sixth temperatures: Gradual evolution from a uniform doubly stochastic matrix to a permutation matrix is shown.

15

The $l^{(2)}$ norm threshold ($l^{(2)}_{\text{thr}}$) and κ were set to 0.02 and 0.001 respectively. Our method always gave a correct permutation matrix for all the examples indicating that we have not "broken" our algorithm. We are currently extending this approach to inexact, weighted graph matching where the limitations of this approach can be better studied. These results are superior to the results obtained by the eigendecomposition approach [20] and the linear programming approach [22].

We have recently developed a related graph matching algorithm based on an objective function which is equivalent up to fixpoints to the free energy in (9) [59, 18]. While the objective function is similar, the optimization dynamics are very different from those of the Lagrangian relaxation algorithm. Line searches and Lagrangian relaxation are eschewed in favor of a discrete algorithm which satisfies the doubly stochastic matrix constraints via iterated row and column normalization. The method is much faster (by a factor of 10 or more) but less accurate. The Lagrangian relaxation algorithm is amenable to analog VLSI implementation and its utility may lie there. The discrete algorithm may be competitive (in terms of speed and accuracy) with other algorithms implemented on digital computers.

5   Relationship to Yuille and Kosowsky

The free energy in (9) is closely related to the approach of Yuille and Kosowsky [53][2]. Their approach, applicable to quadratic assignment problems in general, expresses the doubly stochastic matrix constraint using an $x \log(x)$ barrier function and two Lagrange parameters (enforcing WTA constraints). The barrier function parameter plays the role of deterministic annealing.

The free energy before elimination of the match matrix variables is

$$F(U^{(1)}, M^{(1)}, U^{(2)}, M^{(2)}, \lambda, \beta) = \frac{1}{2} \sum_{ai} \left(\sum_b G_{ab} M^{(1)}_{bi} - \sum_j M^{(2)}_{aj} g_{ji} \right)^2$$

$$+ \sum_{ai} \left[\lambda_{ai} \left(M^{(1)}_{ai} - M^{(2)}_{ai} \right) + \frac{\gamma}{2} \left(\frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2} \left[1 - \frac{M^{(1)}_{ai} + M^2_{ai}}{2} \right] \right) \right]$$

---

[2]We acknowledge Alan Yuille for pointing out the close relationship between our Lagrangian decomposition approach and the barrier function approach developed in [53].
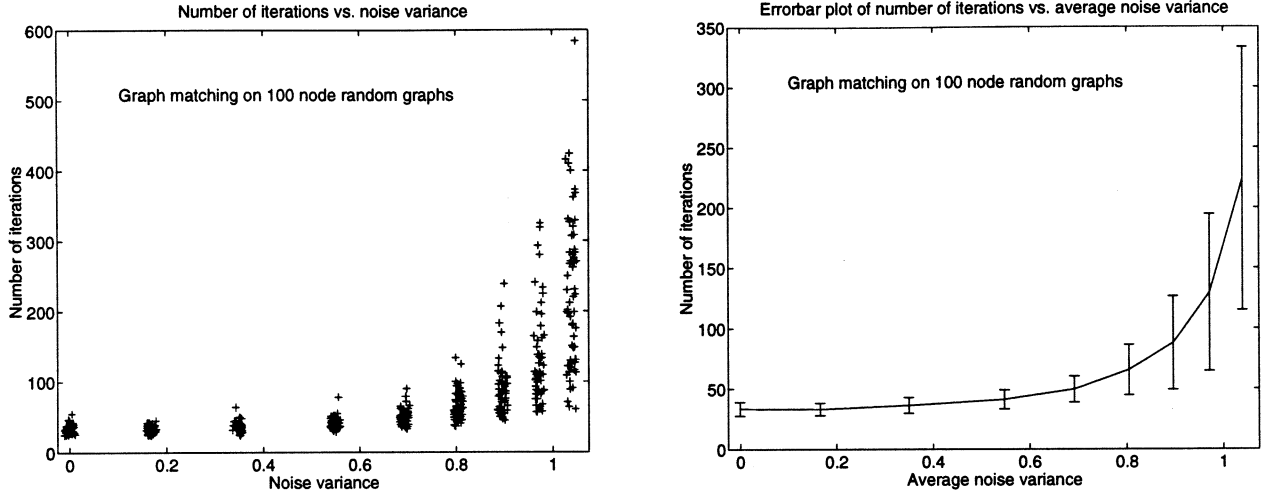
Figure 7: Number of iterations vs. noise variance: Each "+" in the figure denotes the number of iterations taken for a 100 node graph matching run. A total of 540 graph matching examples are presented. Independent, uniform noise is added to each edge. The figure on the right is an errorbar plot of the data presented in the figure on the left. Note the gradual increase in the variance of the number of iterations.

$$+\frac{1}{\beta}\left(\sum_{ai} U_{ai}^{(1)} M_{ai}^{(1)} - \sum_i \log \sum_a \exp\left\{U_{ai}^{(1)}\right\}\right) + \frac{1}{\beta}\left(\sum_{ai} U_{ai}^{(2)} M_{ai}^{(2)} - \sum_a \log \sum_i \exp\left\{U_{ai}^{(2)}\right\}\right).$$

Differentiating the above free energy w.r.t $U^{(1)}$ and equating the result to zero, we get

$$M_{ai}^{(1)} = \frac{\exp\left(U_{ai}^{(1)}\right)}{\sum_b \exp\left(U_{bi}^{(1)}\right)}. \tag{16}$$

This is rewritten to reflect the dependence of $U^{(1)}$ on $M^{(1)}$:

$$M_{ai}^{(1)} \sum_b \exp\left(U_{bi}^{(1)}\right) = \exp\left(U_{ai}^{(1)}\right)$$

$$\Rightarrow U_{ai}^{(1)} = \log\left(M_{ai}^{(1)}\right) + \log \sum_b \exp\left(U_{bi}^{(1)}\right). \tag{17}$$

Equation (17) consists of a set of self-consistent equations from which $U^{(1)}$ cannot be eliminated. Substituting (17) in the entropy term

$$S = \sum_{ai} U_{ai}^{(1)} M_{ai}^{(1)} - \sum_i \log \sum_b \exp\left(U_{bi}^{(1)}\right),$$

we get

$$S = \sum_{ai} M_{ai}^{(1)} \log\left(M_{ai}^{(1)}\right) + \sum_i \left[\log \sum_b \exp\left(U_{bi}^{(1)}\right)\right]\left(\sum_a M_{ai}^{(1)} - 1\right).$$
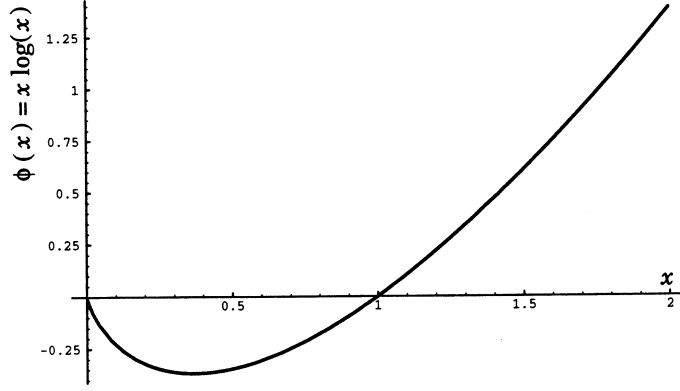
17

Figure 8: The $x \log(x)$ barrier function. The free energy resulting from the Lagrangian decomposition approach is related via a change of variables to the approach of Yuille and Kosowsky. The latter approach uses an $x \log(x)$ barrier function and two Lagrange parameters for the two WTA constraints.

The $\log \sum_a \exp \left(U_{ai}^{(1)} \right)$ term is replaced by a new variable ν_i which is a Lagrange parameter enforcing the WTA constraint $\sum_a M_{ai}^{(1)} = 1$. A new $x \log(x)$ barrier function is obtained. This was first shown by Yuille and Kosowsky [53]. Unlike traditional barrier functions [40], this barrier function (shown in Figure 8) does not go to infinity as $x \to 0$. Nevertheless, it achieves the objective of keeping the match matrix positive.

An $x \log(x)$ barrier function and a Lagrange parameter μ_a are similarly associated with the WTA constraint $\sum_i M_{ai}^{(2)} = 1$. When this is done, the new free energy takes the form

$$F(M^{(1)}, M^{(2)}, \lambda, \mu, \nu, \beta) = \frac{1}{2} \sum_{ai} \left(\sum_b G_{ab} M_{bi}^{(1)} - \sum_j M_{aj}^{(2)} g_{ji} \right)^2 \quad (18)$$

$$+ \sum_{ai} \left[\lambda_{ai} \left(M_{ai}^{(1)} - M_{ai}^{(2)} \right) + \frac{\gamma}{2} \left(\frac{M_{ai}^{(1)} + M_{ai}^{(2)}}{2} \left[1 - \frac{M_{ai}^{(1)} + M_{ai}^2}{2} \right] \right) \right]$$

$$+ \frac{1}{\beta} \left\{ \sum_{ai} M_{ai}^{(1)} \log(M_{ai}^{(1)}) + \sum_i \nu_i \left(\sum_a M_{ai}^{(1)} - 1 \right) + \sum_{ai} M_{ai}^{(2)} \log(M_{ai}^{(2)}) + \sum_a \mu_a \left(\sum_i M_{ai}^{(2)} - 1 \right) \right\}.$$

Differentiating the above free energy w.r.t λ and setting the result to zero, we get $M_{ai}^{(1)} = M_{ai}^{(2)}$. Substituting this in (18), we get

$$F(M, \mu, \nu, \beta) = \frac{1}{2} \sum_{ai} \left(\sum_b G_{ab} M_{bi} - \sum_j M_{aj} g_{ji} \right)^2 + \frac{\gamma}{2} \sum_{ai} M_{ai}(1 - M_{ai}) \quad (19)$$

$$+ \frac{1}{\beta} \left\{ \sum_{ai} 2 M_{ai} \log(M_{ai}) + \sum_i \nu_i \left(\sum_a M_{ai} - 1 \right) + \sum_a \mu_a \left(\sum_i M_{ai} - 1 \right) \right\}.$$

18

A Lagrangian relaxation network similar to the one reported in this paper can be obtained from the above free energy. At a theoretical level, the two formulations are equivalent. No penalty functions have been used to express the WTA constraints and doubly stochastic match matrices are obtained at each temperature setting when the Lagrange parameters converge.

Other optimizing network approaches using penalty functions [39, 19] can be related to ours by replacing Lagrange parameters with penalty functions. For example, Simić [19] uses a Potts glass softmax and a penalty function for the two WTA constraints. The penalty functions satisfy the doubly stochastic matrix constraint asymptotically as opposed to our approach where this constraint is satisfied at each temperature.

6   Conclusions

We have formulated and tested a Lagrangian relaxation network for graph matching. A distinguishing feature of our approach is exact constraint satisfaction at each temperature within deterministic annealing. In addition, the reversal of the dynamics on the graph matching term allows us to simultaneously ascend on the matching and constraint satisfaction terms. The results for both matching and isomorphism are impressive. Perfect results are obtained for almost all instances. We are currently extending our Lagrangian relaxation approach to inexact, weighted graph matching with subsequent application to object recognition. Finally, Lagrangian relaxation networks similar to the ones presented here can be designed for other quadratic assignment problems like the traveling salesman problem.

Acknowledgements

A Derivation of the free energy

In this section, we begin with the partition function in (8) and systematically demonstrate the change of variables and other calculations leading to the final free energy in (12).

$$Z(\beta) = \sum_{\left\{ \begin{array}{c} V^{(1)}, V^{(2)}| \\ \sum_a V_{ai}^{(1)} = 1 \\ \sum_i V_{ai}^{(2)} = 1 \\ V_{ai}^{(1)} = V_{ai}^{(2)} \end{array} \right\}} \exp\left(-\beta \left[\frac{1}{2} \sum_{ai} \left(\sum_b G_{ab} V_{bi}^{(1)} - \sum_j V_{aj}^{(2)} g_{ji} \right)^2 \right. \right.$$

$$\left. \left. + \frac{\gamma}{2} \sum_{ai} \frac{V_{ai}^{(1)} + V_{ai}^{(2)}}{2} \left(1 - \frac{V_{ai}^{(1)} + V_{ai}^{(2)}}{2} \right) \right] \right). \tag{20}$$

We now illustrate the Potts glass encoding of the WTA constraint. The treatment follows [31]. Consider the following partition function arising from an energy function $E(V)$, $\sum_i V_i = 1$, $V_i \in \{0, 1\}$. Following [31], we get

$$Z(\beta) = \sum_{\left\{ \begin{array}{c} V| \\ \sum_i V_i = 1 \end{array} \right\}} \exp\left(-\beta E(V) \right)$$

$$= \sum_{\left\{ \begin{array}{c} V| \\ \sum_i V_i = 1 \end{array} \right\}} \int_R dM \, \delta(V - M) \exp\left(-\beta E(M) \right)$$

$$= \int_R dM \exp(-\beta E(M)) \int_I dU \sum_{\left\{ \begin{array}{c} V| \\ \sum_i V_i = 1 \end{array} \right\}} \exp\left[\sum_i U_i (V_i - M_i) \right]$$

$$= \int_R dM \int_I dU \exp\left[-\beta E(M) - \sum_i U_i M_i + \log\left(\sum_i \exp(U_i) \right) \right].$$

The deterministic annealing trick evaluates the above integrand at its saddle points and uses the result as an approximation to the partition function integral.

$$Z(\beta) \approx Z(\hat{M}, \hat{U}, \beta) = \exp\left[-\beta\left(E(\hat{M}) + \frac{1}{\beta}\left[\sum_i \hat{U}_i \hat{M}_i - \log\sum_i \exp(\hat{U}_i)\right]\right)\right] \stackrel{\text{def}}{=} \exp\left(-\beta F\right) \quad (21)$$

where $\hat{M}$ and $\hat{U}$ are the solutions to

$$\frac{\partial F}{\partial M_i} = 0 \Rightarrow \hat{U}_i = -\beta\frac{\partial E}{\partial M_i}, \text{ and } \frac{\partial F}{\partial U_i} = 0 \Rightarrow \hat{M}_i = \frac{\exp(U_i)}{\sum_i \exp(U_i)}. \quad (22)$$

Here, $\sum_i \hat{M}_i = 1$ and a single winner is obtained as β is increased. A continuation method is obtained from the saddle-point approximation. At each temperature setting, a relaxation network is used to obtain the pair $\hat{M}$ and $\hat{U}$. The inverse temperature β is increased and the earlier solution is used as an initial condition for the next stage.

Writing the details of the saddle-point approximation for the graph matching objective and its two WTAs is awkward. Below, we present a "bare bones" version of the derivation.

$$Z(\beta) = \int_R dM^{(1)} \int_I dU^{(1)} \int_R dM^{(2)} \int_I dU^{(2)} \int_I d\lambda \sum_{\left\{\begin{array}{c} V^{(1)}| \\ \sum_a V^{(1)}_{ai} = 1 \end{array}\right\}} \exp\left[\sum_{ai} U^{(1)}_{ai}(M^{(1)}_{ai} - V^{(1)}_{ai})\right]$$

$$\sum_{\left\{\begin{array}{c} V^{(2)}| \\ \sum_i V^{(2)}_{ai} = 1 \end{array}\right\}} \exp\left[\sum_{ai} U^{(2)}_{ai}(M^{(2)}_{ai} - V^{(2)}_{ai})\right] \exp\left(-\beta\left[\frac{1}{2}\sum_{ai}\left(\sum_b G_{ab}M^{(1)}_{bi} - \sum_j M^{(2)}_{aj}g_{ji}\right)^2\right.\right.$$

$$\left.\left. + \frac{\gamma}{2}\sum_{ai}\frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2}\left(1 - \frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2}\right) + \sum_{ai}\lambda_{ai}(M^{(1)}_{ai} - M^{(2)}_{ai})\right]\right)$$

$$\approx \exp\left(-\beta F(\hat{U^{(1)}}, \hat{M^{(1)}}, \hat{U^{(2)}}, \hat{M^{(2)}}, \hat{\lambda}, \beta)\right)$$

where

$$F(U^{(1)}, M^{(1)}, U^{(2)}, M^{(2)}, \lambda, \beta) = \frac{1}{2}\sum_{ai}\left(\sum_b G_{ab}M^{(1)}_{bi} - \sum_j M^{(2)}_{aj}g_{ji}\right)^2$$

$$+ \sum_{ai}\left[\lambda_{ai}\left(M^{(1)}_{ai} - M^{(2)}_{ai}\right) + \frac{\gamma}{2}\sum_{ai}\frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2}\left(1 - \frac{M^{(1)}_{ai} + M^{(2)}_{ai}}{2}\right)\right]$$

$$+ \frac{1}{\beta}\left(\sum_{ai} U^{(1)}_{ai}M^{(1)}_{ai} - \sum_i \log\sum_a \exp\left\{U^{(1)}_{ai}\right\}\right) + \frac{1}{\beta}\left(\sum_{ai} U^{(2)}_{ai}M^{(2)}_{ai} - \sum_i \log\sum_a \exp\left\{U^{(2)}_{ai}\right\}\right).$$

B Non-negative definiteness of the Hessian

In this section, we show that the Hessian of the objective function in (12) with respect to the reversed "neuron" μ and the Lagrange parameter λ is non-negative definite. The proof is carried out for the isomorphism case alone. Since the only difference between the energy function for graph isomorphism and graph matching is a convex $\sum_{ai} \mu_{ai}^2$ term, the proof also carries over to the graph matching energy function. For the Hessian to be non-negative definite, we require for any x and y,

$$D = \sum_{aibj} \left[x_{ai} \left(\frac{1}{\beta} \frac{\partial^2 F}{\partial \mu_{ai} \partial \mu_{bj}} \right) x_{bj} + 2 y_{ai} \left(\frac{1}{\beta} \frac{\partial^2 F}{\partial \lambda_{ai} \partial \mu_{bj}} \right) x_{bj} + y_{ai} \left(\frac{1}{\beta} \frac{\partial^2 F}{\partial \lambda_{ai} \partial \lambda_{bj}} \right) y_{bj} \right] \geq 0. \quad (23)$$

Now

$$\frac{1}{\beta} \frac{\partial^2 F}{\partial \mu_{ai} \partial \mu_{bj}} = -\frac{1}{\beta} \left(\sum_c G_{ac} \frac{\partial M_{ci}^{(1)}}{\partial \mu_{bj}} - \sum_k \frac{\partial M_{ak}^{(2)}}{\partial \mu_{bj}} g_{ki} \right), \quad (24)$$

$$\frac{1}{\beta} \frac{\partial^2 F}{\partial \lambda_{ai} \partial \lambda_{bj}} = -\frac{1}{\beta} \left(\frac{\partial M_{ai}^{(1)}}{\partial \lambda_{bj}} - \frac{\partial M_{ai}^{(2)}}{\partial \lambda_{bj}} \right), \text{ and} \quad (25)$$

$$\frac{1}{\beta} \frac{\partial^2 F}{\partial \lambda_{ai} \partial \mu_{bj}} = -\frac{1}{\beta} \left(\frac{\partial M_{ai}^{(1)}}{\partial \mu_{bj}} - \frac{\partial M_{ai}^{(2)}}{\partial \mu_{bj}} \right). \quad (26)$$

Rewriting $M_{ci}^{(1)}$ and $M_{ak}^{(2)}$, for convenience:

$$M_{ci}^{(1)} = \frac{\exp\left(-\beta \left[\sum_b G_{bc} \mu_{bi} + \lambda_{ci} \right] \right)}{\sum_d \exp\left(-\beta \left[\sum_b G_{bd} \mu_{bi} + \lambda_{di} \right] \right)}, \quad M_{ak}^{(2)} = \frac{\exp\left(\beta \left[\sum_j g_{kj} \mu_{aj} + \lambda_{ak} \right] \right)}{\sum_l \exp\left(\beta \left[\sum_j g_{lj} \mu_{aj} + \lambda_{al} \right] \right)}. \quad (27)$$

From (27), the relevant partial derivatives required in (24), (25) and (26) are evaluated.

$$\frac{\partial M_{ci}^{(1)}}{\partial \mu_{bj}} = -\beta \delta_{ij} M_{ci}^{(1)} \left(G_{bc} - \sum_d G_{bd} M_{di}^{(1)} \right), \quad (28)$$

$$\frac{\partial M_{ak}^{(2)}}{\partial \mu_{bj}} = \beta \delta_{ab} M_{ak}^{(2)} \left(g_{kj} - \sum_l g_{lj} M_{al}^{(2)} \right), \quad (29)$$

and

$$\frac{\partial M_{ai}^{(1)}}{\partial \lambda_{bj}} = -\beta \delta_{ij} M_{ai}^{(1)} \left(\delta_{ab} - M_{bi}^{(1)} \right), \quad (30)$$

$$\frac{\partial M_{ai}^{(2)}}{\partial \lambda_{bj}} = \beta \delta_{ab} M_{ai}^{(2)} \left(\delta_{ij} - M_{aj}^{(2)} \right), \quad (31)$$

where δ_{ij} and δ_{ab} are Kronecker delta functions.

From (24), (28) and (29), we get

$$\frac{1}{\beta}\frac{\partial^2 F}{\partial\mu_{ai}\partial\mu_{bj}} = \delta_{ij}\left[\sum_c G_{ac}G_{bc}M_{ci}^{(1)} - \left(\sum_c G_{ac}M_{ci}^{(1)}\right)\left(\sum_d G_{bd}M_{di}^{(1)}\right)\right]$$
$$+ \delta_{ab}\left[\sum_k g_{ki}g_{kj}M_{ak}^{(2)} - \left(\sum_k g_{ki}M_{ak}^{(2)}\right)\left(\sum_l g_{lj}M_{al}^{(2)}\right)\right]. \tag{32}$$

From (25), (30) and (31), we get

$$\frac{1}{\beta}\frac{\partial^2 F}{\partial\lambda_{ai}\partial\lambda_{bj}} = \delta_{ij}M_{ai}^{(1)}\left(\delta_{ab} - M_{bi}^{(1)}\right) + \delta_{ab}M_{ai}^{(2)}\left(\delta_{ij} - M_{aj}^{(2)}\right). \tag{33}$$

And finally, from (26), (28) and (29), we get

$$\frac{1}{\beta}\frac{\partial^2 F}{\partial\lambda_{ai}\partial\mu_{bj}} = \delta_{ij}M_{ai}^{(1)}\left(G_{ba} - \sum_c G_{bc}M_{ci}^{(1)}\right) + \delta_{ab}M_{ai}^{(2)}\left(g_{ij} - \sum_k g_{kj}M_{ak}^{(2)}\right). \tag{34}$$

Using (32), (33) and (34), we evaluate D in (23):

$$D = \sum_{aibj} x_{ai}\delta_{ij}\left[\sum_c G_{ac}G_{bc}M_{ci}^{(1)} - \left(\sum_c G_{ac}M_{ci}^{(1)}\right)\left(\sum_d G_{bd}M_{di}^{(1)}\right)\right]x_{bj}$$
$$+ \sum_{aibj} x_{ai}\delta_{ab}\left[\sum_k g_{ki}g_{kj}M_{ak}^{(2)} - \left(\sum_k g_{ki}M_{ak}^{(2)}\right)\left(\sum_l g_{lj}M_{al}^{(2)}\right)\right]x_{bj}$$
$$+ 2\sum_{aibj} y_{ai}\left[\delta_{ij}M_{ai}^{(1)}\left(G_{ba} - \sum_c G_{bc}M_{ci}^{(1)}\right)x_{bj} + \delta_{ab}M_{ai}^{(2)}\left(g_{ij} - \sum_k g_{kj}M_{ak}^{(2)}\right)x_{bj}\right]$$
$$+ \sum_{aibj} y_{ai}\left[\delta_{ij}M_{ai}^{(1)}\left(\delta_{ab} - M_{bi}^{(1)}\right) + \delta_{ab}M_{ai}^{(2)}\left(\delta_{ij} - M_{aj}^{(2)}\right)\right]y_{bj}. \tag{35}$$

This is rearranged to give

$$D = \sum_{i,c}\left(\sum_a G_{ac}x_{ai}\right)\left(\sum_b G_{bc}x_{bi}\right)M_{ci}^{(1)} - \sum_i\left[\sum_c\left(\sum_a G_{ac}x_{ai}\right)M_{ci}^{(1)}\right]\left[\sum_d\left(\sum_b G_{bd}x_{bi}\right)M_{di}^{(1)}\right]$$
$$+ \sum_{a,k}\left(\sum_i g_{ki}x_{ai}\right)\left(\sum_j g_{kj}x_{aj}\right)M_{ak}^{(2)} - \sum_a\left[\sum_k\left(\sum_i g_{ki}x_{ai}\right)M_{ak}^{(2)}\right]\left[\sum_l\left(\sum_j g_{lj}x_{aj}\right)M_{al}^{(2)}\right]$$
$$+ 2\sum_{i,a} y_{ai}\left(\sum_b G_{ba}x_{bi}\right)M_{ai}^{(1)} - 2\sum_{i,a} y_{ai}\left[\sum_c\left(\sum_b G_{bc}x_{bi}\right)M_{ci}^{(1)}\right]$$
$$+ 2\sum_{a,i} y_{ai}\left(\sum_j g_{ij}x_{aj}\right)M_{ai}^{(2)} - 2\sum_{a,i} y_{ai}\left[\sum_k\left(\sum_j g_{kj}x_{aj}\right)M_{ak}^{(2)}\right]$$
$$+ \sum_{i,a} y_{ai}^2 M_{ai}^{(1)} - \sum_i\left(\sum_a y_{ai}M_{ai}^{(1)}\right)\left(\sum_b y_{bi}M_{bi}^{(1)}\right)$$
$$+ \sum_{a,i} y_{ai}^2 M_{ai}^{(2)} - \sum_a\left(\sum_i y_{ai}M_{ai}^{(2)}\right)\left(\sum_j y_{aj}M_{aj}^{(2)}\right).$$

The above expression is simplified as follows. Define

$$u_{ci} \stackrel{\text{def}}{=} \sum_a G_{ac} x_{ai}, \quad v_{ak} \stackrel{\text{def}}{=} \sum_i g_{ki} x_{ai}. \tag{36}$$

D is rewritten in a more compact form:

$$
\begin{aligned}
D &= \sum_i \left[\sum_a u_{ai}^2 M_{ai}^{(1)} - \left(\sum_a u_{ai} M_{ai}^{(1)} \right)^2 \right] + \sum_a \left[\sum_i v_{ai}^2 M_{ai}^{(2)} - \left(\sum_i v_{ai} M_{ai}^{(2)} \right)^2 \right] \\
&+ 2 \sum_i \left[\sum_a y_{ai} u_{ai} M_{ai}^{(1)} - \left(\sum_a y_{ai} M_{ai}^{(1)} \right) \left(\sum_a u_{ai} M_{ai}^{(1)} \right) \right] \\
&+ 2 \sum_a \left[\sum_i y_{ai} v_{ai} M_{ai}^{(2)} - \left(\sum_i y_{ai} M_{ai}^{(2)} \right) \left(\sum_i v_{ai} M_{ai}^{(2)} \right) \right] \\
&+ \sum_i \left[\sum_a y_{ai}^2 M_{ai}^{(1)} - \left(\sum_a y_{ai} M_{ai}^{(1)} \right)^2 \right] + \sum_a \left[\sum_i y_{ai}^2 M_{ai}^{(2)} - \left(\sum_i y_{ai} M_{ai}^{(2)} \right)^2 \right].
\end{aligned}
\tag{37}
$$

And after further simplification,

$$
\begin{aligned}
D &= \sum_{i,a} \left[u_{ai}^2 M_{ai}^{(1)} + 2 y_{ai} u_{ai} M_{ai}^{(1)} + y_{ai}^2 M_{ai}^{(1)} \right] + \sum_{a,i} \left[v_{ai}^2 M_{ai}^{(2)} + 2 y_{ai} v_{ai} M_{ai}^{(2)} + y_{ai}^2 M_{ai}^{(2)} \right] \\
&- \sum_i \left[\left(\sum_a u_{ai} M_{ai}^{(1)} \right)^2 + 2 \left(\sum_a y_{ai} M_{ai}^{(1)} \right) \left(\sum_a u_{ai} M_{ai}^{(1)} \right) + \left(\sum_a y_{ai} M_{ai}^{(1)} \right)^2 \right] \\
&- \sum_a \left[\left(\sum_i v_{ai} M_{ai}^{(2)} \right)^2 + 2 \left(\sum_i y_{ai} M_{ai}^{(2)} \right) \left(\sum_i v_{ai} M_{ai}^{(2)} \right) + \left(\sum_i y_{ai} M_{ai}^{(2)} \right)^2 \right].
\end{aligned}
$$

A second round of definitions completes the "compactification" of D.

$$w_{ai} \stackrel{\text{def}}{=} u_{ai} + y_{ai}, \quad z_{ai} \stackrel{\text{def}}{=} v_{ai} + y_{ai}. \tag{38}$$

And finally, D is shown to be

$$D = \sum_i \left[\sum_a w_{ai}^2 M_{ai}^{(1)} - \left(\sum_a w_{ai} M_{ai}^{(1)} \right)^2 \right] + \sum_a \left[\sum_i z_{ai}^2 M_{ai}^{(2)} - \left(\sum_i z_{ai} M_{ai}^{(2)} \right)^2 \right]. \tag{39}$$

For any i, setting $f_a = w_{ai}\sqrt{M_{ai}^{(1)}}$ and $g_a = \sqrt{M_{ai}^{(1)}}$ and using the Cauchy-Schwartz inequality

$$\sum_a f_a^2 \sum_a g_a^2 \geq \left(\sum_a f_a g_a \right)^2$$

with equality occurring when $f_a = h g_a$ and h is an arbitrary non-zero constant, we get [32]

$$\sum_a w_{ai}^2 M_{ai}^{(1)} \sum_a M_{ai}^{(1)} \geq \left(\sum_a w_{ai} M_{ai}^{(1)} \right)^2$$

$$\Rightarrow \sum_a w_{ai}^2 M_{ai}^{(1)} \geq \left(\sum_a w_{ai} M_{ai}^{(1)} \right)^2 \tag{40}$$

24

using the fact that $\sum_a M_{ai}^{(1)} = 1$ is always satisfied.

Similarly, for any a, setting $f_i = z_{ai}\sqrt{M_{ai}^{(2)}}$ and $g_i = \sqrt{M_{ai}^{(2)}}$ and using the Cauchy-Schwartz inequality, we get

$$\sum_i z_{ai}^2 M_{ai}^{(2)} \sum_i M_{ai}^{(2)} \geq \left(\sum_i z_{ai} M_{ai}^{(2)}\right)^2$$

$$\Rightarrow \sum_i z_{ai}^2 M_{ai}^{(2)} \geq \left(\sum_i z_{ai} M_{ai}^{(2)}\right)^2 \tag{41}$$

using the fact that $\sum_i M_{ai}^{(2)} = 1$ is always satisfied.

Therefore $D \geq 0$ with equality occurring for those vectors $w_{ai} = s_i$, $\forall\, a$ and $z_{ai} = t_a$, $\forall\, i$, where s and t are arbitrary vectors.

References

[1] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 3, pp. 504–519, Sept. 1981.

[2] 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.

[3] C. von der Malsburg and E. Bienenstock, "Statistical coding and short-term synaptic plasticity: A scheme for knowledge representation in the brain", in E. Bienenstock, F. Fogelman-Soulié, and G. Weisbuch, editors, *Disordered Systems and Biological Organization*, pp. 247–252. Springer–Verlag, Berlin, 1986.

[4] C. von der Malsburg, "Pattern recognition by labeled graph matching", *Neural Networks*, vol. 1, pp. 141–148, 1988.

[5] R. Kree and A. Zippelius, "Recognition of topological features of graphs and images in neural networks", *J. Phys. A*, vol. 21, pp. L813–L818, 1988.

[6] E. Mjolsness, G. Gindi, and P. Anandan, "Optimization in model matching and perceptual organization", *Neural Computation*, vol. 1, pp. 218–229, 1989.

[7] J. Segen, "Model learning and recognition of nonrigid objects", *in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 597–602. IEEE Press, 1989.

[8] W. Li and N. M. Nasrabadi, "Object recognition based on graph matching implemented by a Hopfield–style neural network", *in Intl. Joint Conf. on Neural Networks (IJCNN)*, vol. 2, pp. 287–290. IEEE Press, 1989.

[9] V. Tresp, "A neural network approach for three–dimensional object recognition", in R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pp. 306–312. Morgan Kaufmann, San Mateo, CA, 1991.

[10] E. Mjolsness, "Bayesian inference on visual grammars by neural nets that optimize", Technical Report YALEU/DCS/TR-854, Department of Computer Science, Yale University, May 1991.

[11] E. Wong, "Model matching in robot vision by subgraph isomorphism", *Pattern Recognition*, vol. 25, pp. 287–303, 1992.

[12] S. Z. Li, "Matching: invariant to translations, rotations and scale changes", *Pattern Recognition*, vol. 25, pp. 583–594, 1992.

[13] R. Bergevin and M. D. Levine, "Generic object recognition: Building and matching coarse descriptions from line drawings", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, pp. 19–36, Jan. 1993.

[14] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic–link architecture", *IEEE Trans. Computers*, vol. 42, pp. 300–311, Mar. 1993.

[15] J. Kittler, W. J. Christmas, and M. Petrou, "Probabilistic relaxation for matching problems in computer vision", *in Fourth Intl. Conf. on Computer Vision*, pp. 666–673. IEEE Press, 1993.

[16] 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.

[17] A. Pearce, T. Caelli, and W. F. Bischof, "Rulegraphs for graph matching in pattern recognition", *Pattern Recognition*, vol. 27, pp. 1231–1247, 1994.

[18] S. Gold, A. Rangarajan, and E. Mjolsness, "Learning with preknowledge: clustering with point- and graph-matching distance measures", Technical Report YALEU/DCS/RR-1037, Department of Computer Science, Yale University, May 1994, To appear in NIPS 7.

[19] P. D. Simić, "Constrained nets for graph matching and other quadratic assignment problems", *Neural Computation*, vol. 3, pp. 268–281, 1991.

[20] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 10, pp. 695–703, Sept. 1988.

[21] H. A. Almohamad, "A polynomial transform for matching pairs of weighted graphs", *J. Applied Math. Modeling*, vol. 15, April 1991.

[22] 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.

[23] A. K. C. Wong and M. You, "Entropy and distance of random graphs with applications to structural pattern recognition", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 7, pp. 599–609, Sept. 1985.

[24] W.-H. Tsai and K.-S. Fu, "Error–correcting isomorphisms of attributed relational graphs for pattern analysis", *IEEE Trans. Syst. Man, Cyber.*, vol. 9, pp. 757–768, Dec. 1979.

[25] R. Motwani, "personal communication", Sept. 1994.

[26] D. S. Johnson, "The NP–Completeness column: An ongoing guide", *Journal of Algorithms*, vol. 13, pp. 502–524, 1992.

[27] D. S. Johnson, "The NP–Completeness column: An ongoing guide", *Journal of Algorithms*, vol. 9, pp. 426–444, 1988.

[28] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP–completeness*, W. H. Freeman, San Francisco, CA, 1979.

[29] D. G. Corneil and C. C. Gotlieb, "An efficient algorithm for graph isomorphism", *J. Ass. Comput. Mach.*, vol. 17, pp. 51–64, Jan. 1970.

[30] J. R. Ullman, "An algorithm for subgraph isomorphism", *J. Ass. Comput. Mach.*, vol. 23, pp. 31–42, Jan. 1976.

[31] C. Peterson and B. Söderberg, "A new method for mapping optimization problems onto neural networks", *Intl. Journal of Neural Systems*, vol. 1, pp. 3–22, 1989.

[32] 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.

[33] M. Guignard and S. Kim, "Lagrangean decomposition: A model yielding stronger Lagrangean bounds", *Mathematical Programming*, vol. 39, pp. 215–228, 1987.

[34] K. Jörnsten and M. Näsberg, "A new Lagrangian relaxation approach to the generalized assignment problem", *European Journal of Operational Research*, vol. 27, pp. 313–323, 1986.

[35] E. Mjolsness and C. Garrett, "Algebraic transformations of objective functions", *Neural Networks*, vol. 3, pp. 651–669, 1990.

[36] C. von der Malsburg, "Network self-organization", in S. F. Zornetzer, J. L. Davis, and C. Lau, editors, *An Introduction to Neural and Electronic Networks*, pp. 421–432. Academic Press, San Diego, CA, 1990.

[37] S. V. B. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield model", *IEEE Trans. Neural Networks*, vol. 1, pp. 204–215, 1990.

[38] A. H. Gee and R. W. Prager, "Polyhedral combinatorics and neural networks", *Neural Computation*, vol. 6, pp. 161–180, Jan. 1994.

[39] J. J. Hopfield and D. Tank, "'Neural' computation of decisions in optimization problems", *Biol. Cyber.*, vol. 52, pp. 141–152, 1985.

[40] D. Luenberger, *Linear and Nonlinear Programming*, Addison–Wesley, Reading, MA, 1984.

[41] G. L. Bilbro, W. E. Snyder, R. Mann, D. Van den Bout, T. Miller, and M. White, "Optimization by mean field annealing", in D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pp. 91–98. Morgan Kaufmann, San Mateo, CA, 1988.

[42] G. L. Bilbro, W. E. Snyder, S. J. Garnier, and J. W. Gault, "Mean field annealing: A formalism for constructing GNC-like algorithms", *IEEE Trans. Neural Networks*, vol. 3, pp. 131–138, Jan. 1992.

[43] I. Kanter and H. Sompolinsky, "Graph optimisation problems and the Potts glass", *J. Phys. A*, vol. 20, pp. L673–L677, 1987.

[44] D. E. Van den Bout and T. K. Miller III, "Improving the performance of the Hopfield–Tank neural network through normalization and annealing", *Biol. Cyber.*, vol. 62, pp. 129–139, 1989.

[45] D. Geiger and A. L. Yuille, "A common framework for image segmentation", *Intl. Journal of Computer Vision*, vol. 6, pp. 227–243, Aug. 1991.

[46] D. E. Van den Bout and T. K. Miller III, "Graph partitioning using annealed networks", *IEEE Trans. Neural Networks*, vol. 1, pp. 192–203, June 1990.

[47] P. D. Simić, "Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations", *Network*, vol. 1, pp. 89–103, 1990.

[48] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters", in D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp. 211–217. Morgan Kaufmann, San Mateo, CA, 1990.

[49] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[50] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, vol. 1 of *A Lecture Notes volume in the Santa Fe Institute Studies in the Sciences of Complexity*, Addison–Wesley, New York, 1991.

[51] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRFs: Surface reconstruction", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, pp. 401–412, May 1991.

[52] A. Rangarajan and R. Chellappa, "A continuation method for image estimation and segmentation", Technical Report CAR-TR-586, Center for Automation Research, University of Maryland, Oct. 1991.

[53] A. L. Yuille and J. J. Kosowsky, "Statistical physics algorithms that converge", *Neural Computation*, vol. 6, pp. 341–356, May 1994.

[54] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems", *Management Science*, vol. 27, pp. 1–18, 1981.

[55] J. C. Platt and A. H. Barr, "Constraint methods for flexible models", *Computer Graphics*, vol. 22, pp. 279–288, Aug. 1988, SIGGRAPH '88.

[56] S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Addison–Wesley, Reading, Mass., 1990.

[57] H. Haken, *Synergetics–An Introduction*, vol. 1 of *Springer Series in Synergetics*, Springer–Verlag, New York, 3rd edition, 1983.

[58] H. Haken, *Advanced Synergetics*, vol. 20 of *Springer Series in Synergetics*, Springer–Verlag, New York, 2nd edition, 1987.

[59] A. Rangarajan, S. Gold, and E. Mjolsness, "A novel optimizing network architecture with applications", Technical Report YALEU/DCS/RR-1036, Department of Computer Science, Yale University, May 1994.

[60] A. Rangarajan and E. Mjolsness, "A Lagrangian relaxation network for graph matching", *in IEEE Intl. Conf. on Neural Networks (ICNN)*, vol. 7, pp. 4629–4634. IEEE Press, 1994.