

Constrained Nets for Graph Matching and Other Quadratic Assignment Problems

Petar D. Simić

*Division of Physics, California Institute of Technology,
Pasadena, CA 91125 USA*

Some time ago Durbin and Willshaw proposed an interesting parallel algorithm (the "elastic net") for approximately solving some geometric optimization problems, such as the Traveling Salesman Problem. Recently it has been shown that their algorithm is related to neural networks of Hopfield and Tank, and that they both can be understood as the semiclassical approximation to statistical mechanics of related physical models. The main point of the elastic net algorithm is seen to be in the way one deals with the constraints when evaluating the effective cost function (free energy in the thermodynamic analogy), and not in its geometric foundation emphasized originally by Durbin and Willshaw. As a consequence, the elastic net algorithm is a special case of the more general physically based computations and can be generalized to a large class of nongeometric problems. In this paper we further elaborate on this observation, and generalize the elastic net to the quadratic assignment problem. We work out in detail its special case, the graph matching problem, because it is an important problem with many applications in computational vision and neural modeling. Simulation results on random graphs, and on structured (hand-designed) graphs of moderate size (20–100 nodes) are discussed.

1 Introduction

Recently, Durbin and Willshaw (1987) proposed an interesting and very efficient algorithm for approximately solving some geometric optimization problems, such as the Euclidean Traveling Salesman Problem (TSP). Their algorithm — they call it the "elastic net" method — naturally lends itself to implementation in parallel hardware. Although it is very different from neural network of Hopfield and Tank (1985), the two are in fact related (Simić 1990). They both can be understood as the semiclassical approximation to statistical mechanics of related physical models, and can be derived and systematically corrected (at least in principle) from the "first principles." Durbin and Willshaw considered their algorithm to be fundamentally geometric — it could be applied to many geometric optimization problems, but not to more general nongeometric problems

such as the TSP with an arbitrary matrix of distances, or the quadratic matching problems (Durbin and Willshaw 1987; Willshaw 1990). We have observed, however, that within the probabilistic approach to optimizations, the main point about the elastic net algorithm is in the way one handles the global constraints when evaluating the *effective cost* function (the free energy in thermodynamic analogy). If all the constraints are enforced softly, the neural network is obtained. On the other hand, if about half of the constraints are enforced strongly, the elastic network is obtained. As a consequence, the elastic net algorithm is a special case of a class of more general, physics-based computations — we call them *constrained neural nets* — and can be generalized to a large class of optimization problems (Simić 1990).

In related work, Peterson and Söderberg (1989) also considered the TSP problem and imposed global constraints on the Hopfield and Tank model by mapping it into a Potts glass. Although they did not establish the connection with the elastic net method, some of the formulas obtained in (Simić 1990) and reviewed here, were independently obtained by these authors. Further support for the idea of enforcing global constraints in generalizing the elastic net came from work of Yuille (1990).

In this paper we generalize the elastic net algorithm to an important class of nongeometric optimization problems — the quadratic assignment problem and its specializations, such as the graph matching problem. These are representative of a general class of *one-to-one (or many)* matching problems characterized by quadratic, as opposed to linear (Yuille 1990) dependence of the objective function on the binary variables. Their potential relevance for invariant object recognition and model matching problems in vision was recently discussed in the context of neural style optimizations (von der Malsburg and Bienenstock 1986; von der Malsburg 1988; see also Mjølness *et al.* 1989; Kree and Zippelius 1988). We discuss our experiments with the *constrained net* algorithm on random graphs, and on structured (hand-designed) graphs of moderate size (20–100 nodes), which are very encouraging.

2 Constrained Mean-Field Nets for Graph Matching and Other Quadratic Assignment Problems

The important thing about a graph is its topology, that is, the way its nodes are linked together. The link connecting two nodes of a graph could represent a physical connection between, say, hand and the shoulder — if the graph represents human body — or it could represent a conceptual relation between the two objects, parts of some higher level (cognitive) structure — for example, object–class relationships. One can do all kinds of transformations on graphs; so long as the transformation does not involve breaking of the graph links, the resulting graph is considered identical to the original.

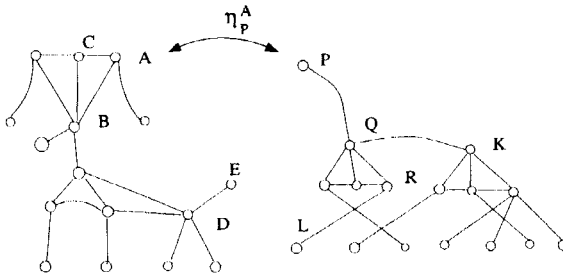


Figure 1: Two hand-designed isomorphic graphs as an example of a graph matching problem.

This invariance with respect to coordinate transformations is an important characteristic of graphs, and makes them attractive for describing invariant relationships between the objects (von der Malsburg and Bienenstock 1986; von der Malsburg 1988). The graph matching problem is to determine if the two graphs have the same topology, that is, if the two graphs are in fact identical. In Figure 1, it is relatively easy to see that the two graphs are identical; they are just the different drawings of the same structure. In the case of more complicated drawings and/or graphs, this becomes very difficult. It is not known if this problem is in the NP-complete class or not, but, except for some special cases, no polynomial algorithm is known (Garey and Johnson 1987). We discuss this problem in the rest of the paper, but our algorithm is applicable to the general quadratic assignment problem.

2.1 Objective Function for Graph Matching. Graphs allow simple mathematical description. Consider the two graphs, such as the ones in the Figure 1. They can be described by the connectivity matrices, G^{AB} and D_{pq} , elements of which take binary values. We call one graph (G) the *model*, and the other (D), the *data*. If the nodes A and B are connected, then $G^{AB} = 1$, otherwise $G^{AB} = 0$. If the two graphs are isomorphic (technical term for “identical”), then there exists a permutation matrix, η_p^A , such that

$$D_{pq} = \sum_{A,B} G^{AB} \eta_p^A \eta_q^B \quad (2.1)$$

The objective of our optimization problem is to find the optimal match, η^0 , such that relation 2.1 is optimally satisfied. If the two graphs are not identical then relations 2.1 cannot be exactly obeyed. The objective

function (Kree and Zippelius 1988) that in both cases measures, in the LMS sense, the degree of the mismatch between the two graphs is given by

$$C = \frac{\lambda}{4} \sum_{p \neq q} \left(D_{pq} - \sum_{A,B} G^{AB} \eta_p^A \eta_q^B \right)^2 \quad (2.2)$$

The η_p^A are similar to the neural variables introduced by Hopfield and Tank. We call them "events" in the following, to emphasize their probabilistic interpretation, which will be developed. If node A ($A = 1, 2, \dots, M$), on the model side, is matched to node p ($p = 1, 2, \dots, N = M$), on the data side, then the event η_p^A is true, that is, $\eta_p^A = 1$, and all the other incompatible events are false, $\eta_p^B = \eta_q^A = 0$ ($B \neq A, q \neq p$). These constraints are conveniently expressed as

$$\sum_A \eta_p^A \eta_q^A = \delta_{pq} \quad (2.3)$$

$$\sum_p \eta_p^A \eta_p^B = \delta^{AB} \quad (2.4)$$

Adding the penalties for violation of the constraints 2.3–2.4, our objective function 2.2, can be written as follows:

$$C = \frac{1}{4} \sum_{A,B,p \neq q} G^{AB} (1 - 2D_{pq}) \eta_p^A \eta_q^B + \sum_{A,p,q} d_{pq} \eta_p^A \eta_q^A + \sum_{A,B,p} \gamma_{AB} \eta_p^A \eta_p^B + \sum_A \gamma_A \left(\sum_p \eta_p^A - 1 \right)^2 + \sum_p d_p \left(\sum_A \eta_p^A - 1 \right)^2 \quad (2.5)$$

where d_p, γ_A, d_{pq} ($d_{pp} = 0$), and γ_{AB} ($\gamma_{AA} = 0$) are arbitrary parameters, penalties for violation of the constraints 2.3–2.4. The active part of C is the first term, since it expresses the matching objective of the problem. When the two nodes on the data side, p and q , are connected ($D_{pq} = 1$), then C will be decreased ($\delta C = -1/2$) if the two connected nodes, A and B ($G^{AB} = 1$) on the model side, are matched to them. If, on the other hand, p and q are not connected ($D_{pq} = 0$), then there is a penalty ($\delta C = +1/2$) if the two connected nodes, A and B , are matched to them. The rest of the terms in 2.5 are penalties. We can reduce a large number of arbitrary parameters in 2.5 by choosing $\gamma_A = \gamma$, and $d_p = D$; the least biased choice since all the nodes are equal. Likewise, the problem of choosing the two-index penalty parameters (d_{pq} and γ_{AB}) can be simplified if one constructs d_{pq} and γ_{AB} using only the natural tensors of the problem, D_{pq} and G^{AB} . For example, we choose $d_{pq} = (1 - \delta_{pq}) [\alpha + (1 - \alpha) D_{pq}]$, expressing all the d_{pq} parameters in terms of just one (α) and the D_{pq} . Again, this reduction is the least biased, since all pairs of connected nodes are treated equal, and all pairs of unconnected nodes are treated equal.

2.2 The Effective Cost Theory as an Alternative to Stochastic Annealing. The probabilistic approach to optimization is based on an analogy between optimization and statistical mechanics (Kirkpatrick *et al.* 1983; Cerny 1982). Given the objective function, one introduces the probabilistic formulation by postulating a certain probability distribution over the space of configurations — in our case, the space of matching events. A convenient choice, familiar from physics, is the Boltzmann distribution, $P_\beta[\eta] = (1/Z_\beta)e^{-\beta C[\eta]}$. The central object one wants to evaluate is the partition function,

$$Z_\beta[\mathbf{H}] = \sum_{\{\eta\}} \exp \left[-\beta \left(C[\eta] + \sum_{A,p} H_p^A \eta_p^A \right) \right] \quad (2.6)$$

where H_p^A is some matrix of external “fields.” Once the partition function is evaluated — typically using some approximations — all the relevant statistical information about the system can be deduced from it. For example, differentiating the generating functional, $-(1/\beta)\ln Z_\beta[\mathbf{H}]$ with respect to external fields about $H = 0$, one generates all correlation functions of η distribution. At zero temperature ($\beta \rightarrow \infty$) and no external fields present ($H = 0$), the partition function is dominated by the term of smallest cost, $Z_\beta \approx \exp(-\beta C[\eta^0])$. This has an important implication for optimization: as the temperature is decreased the expectation value of the matching matrix ($\langle \eta \rangle$), converges toward the optimal match (η^0) between the two graphs, that is,

$$\langle \eta \rangle \equiv \sum \eta P_\beta[\eta] \approx_{\beta \rightarrow \infty} \eta^0 + O(\exp[-\beta|\delta C|]) \quad (2.7)$$

This suggests that instead of minimizing the cost function C directly, one could attempt to calculate the thermal averages such as $\langle \eta \rangle$ and $\langle C[\eta] \rangle$, at a given temperature, and follow how these averages change as one decreases the temperature toward zero (Kirkpatrick *et al.* 1983; Cerny 1982).

It is quite interesting that the expectation values $\langle \eta \rangle$ can be obtained by minimizing an *effective cost function*, known in physics as the thermodynamic free energy. It is defined as the Legendre transformation of the generating functional ($-(1/\beta) \ln Z_\beta[\mathbf{H}]$):

$$C_\beta^{\text{eff}}[\mathbf{m}] = -\frac{1}{\beta} \ln Z_\beta[\mathbf{H}] - \sum_{A,p} H_p^A m_p^A \quad (2.8)$$

where m_p^A is the expectation value, that is, the probability of the event η_p^A in the presence of some external fields (\mathbf{H}),

$$m_p^A \equiv \langle \eta_p^A \rangle_{\beta, H} = -\frac{1}{\beta} \frac{\delta \ln Z_\beta}{\delta H_p^A} \quad (2.9)$$

Differentiating C^{eff} with respect to \mathbf{m} , one obtains

$$\frac{\delta C_{\beta}^{\text{eff}}}{\delta m_p^A} = -H_p^A + \sum_{B,q} \left(-\frac{1}{\beta} \frac{\delta}{\delta H_q^B} \ln Z_{\beta} - m_q^B \right) \frac{\delta H_q^B}{\delta m_q^A} \equiv -H_p^A \quad (2.10)$$

where the last equality follows from the definition of \mathbf{m} . We see that at $H = 0$ the $C^{\text{eff}}[\mathbf{m}]$ has local extremum at \mathbf{m} equal to the expectation value of $\boldsymbol{\eta}$ defined in 2.7. This suggests that instead of calculating $\langle \boldsymbol{\eta} \rangle_{\beta}$ numerically, one could attempt to evaluate analytically the *effective cost function* — note that it depends on the temperature — and then, while successively decreasing values of temperature, minimize it using differential equations. These *deterministic* equations describe relaxation of probabilities and stochastic averages as a function of the temperature. They provide an effective deterministic description of the system, completely equivalent, in principle, to stochastic descriptions obtained by Monte-Carlo simulations.¹ Solving them at successively decreasing temperatures — a process sometimes called “*mean-field*” *annealing* — is a way to model simulated annealing within a deterministic framework, and it is equivalent to *stochastic annealing*. From the point of view of combinatorial optimizations, it is interesting to note that the C_{β}^{eff} is a function of *continuous* variables (\mathbf{m}) and minimizing it, is a *continuous optimization* problem.

In general, $C_{\beta}^{\text{eff}} = C - 1/\beta S$, where S is the entropy of the system. The functional form of the entropy function will depend on what states are available to the system, that is, what states we sum over in evaluating the partition function in 2.6. It follows that, depending on whether we enforce *all*, *some*, or *none* of the constraints in summing over configurations of our optimization problem, we will get different functional forms of the C_{β}^{eff} . It is known (Simić 1990) that in the semiclassical approximation, *none* leads in general to neural networks of Hopfield and Tank (1985), and *some* in the case of the TSP, leads to the elastic net algorithm of Durbin and Willshaw (1987).

2.3 The Effective Cost Function for Quadratic Assignment. To generalize the elastic net algorithm of Durbin and Willshaw (1987), along the lines proposed in Simić (1990), we should construct the entropy function enforcing a significant number of constraints 2.3–2.4 when summing over the configurations in 2.6. If we assume that the graph from the data side may have in general more nodes than the model graph, it makes sense to choose constraints 2.4 (nondiagonal part) and the diagonal part of 2.3 to be enforced, while enforcing softly, with the help of the penalty terms, the remaining constraints. These are the same constraints as in our formulation of the TSP problem, and many of the formulas we need to generalize

¹In practice, one has to use approximations in deriving the effective cost function and to this extent the effective deterministic equations will only approximate the stochastic description obtained by Monte-Carlo simulations.

the elastic net to graph matching, have been already derived there, and then specialized to the Euclidean TSP. In the following, we just outline the basic elements of this derivation, which holds for general quadratic assignment problem with an arbitrary matrix J_{pq}^{AB} , then, we specialize our results to the graph matching problem, with J_{pq}^{AB} according to equation 2.5, and discuss the temperature dependence of the penalty terms — an important ingredient of the elastic net algorithm.

For reasons that will become clear shortly, we will consider the following, regularized partition function:

$$Z_{\beta}^{(\lambda)}[\mathbf{H}] = \sum_{\{\eta\}} e^{-\beta [C(\boldsymbol{\eta}) + \sum_{A,p} \eta_p^A H_p^A - \frac{\lambda}{8} \sum_{A,p} (2\eta_p^A - 1)^2]} \tag{2.11}$$

The third term in the exponent is trivial configuration-independent constant, and its effect is just to multiply $Z_{\beta}[\mathbf{H}]$, defined in 2.6, by $e^{\beta(\lambda/8)MN}$. This clearly does not affect the η -correlation functions — they are derivatives of $\ln Z$ and are invariant under the rescaling of the partition function. This means that we can evaluate the effective cost function, and other results using the $Z_{\beta}^{(\lambda)}[\mathbf{H}]$, and at the end set $\lambda = 0$ to obtain $Z_{\beta}[\mathbf{H}]$ ($= Z_{\beta}^{(0)}[\mathbf{H}]$), and other final expressions independent of λ .

Dropping the third and the last term from the objective function in 2.5 — these are identically zero when the subset of constraints are obeyed — and introducing the “potentials” ϕ_p^A dual to neural variables (η_p^A), we can rewrite the partition function 2.11 as

$$Z_{\beta}^{(\lambda)}[\mathbf{H}] = \sum_{\{\boldsymbol{\eta}\}} \int d\phi e^{-\beta (\frac{1}{2} \sum_{A,n} \sum_{p,q} \phi_p^A ([\lambda \mathbf{1} - \mathbf{J}]^{-1})_{pq}^{AA} \phi_q^A + \sum_{A,p} \eta_p^A (\phi_p^A + H_p^A + \frac{\lambda}{2}))} \tag{2.12}$$

where $d\phi$ is the scale invariant integration measure defined in Simić (1990), and the regularization parameter λ is chosen such that the inverted matrix in the exponent has all the eigenvalues positive.² In the special case of the graph matching problem, we find from the equation 2.5,

$$J_{pq}^{AB} \equiv \frac{1}{2} \{ G^{AB} (1 - 2D_{pq}) + [\alpha + (1 - \alpha)D_{pq}] \delta^{AB} + \gamma \delta^{AB} \} (1 - \delta_{pq}) \tag{2.13}$$

Since the dependence on event variables is linear in the exponent of 2.12, we can sum over all η -configurations that obey 2.4 and the diagonal part of 2.3. Then, expanding the integral in 2.12 around its saddle point (this

²This choice is convenient and leads to the cleaner looking mathematical expressions, but is not essential because even in the $\lambda = 0$ case there is no real divergence in 2.11 — the exponential divergence of the integrand is matched with the same divergence in the denominator of the measure. This is in agreement with the previously discussed fact that for any λ , $Z_{\beta}^{(\lambda)}[\mathbf{H}] = Z_{\beta}[\mathbf{H}] \exp[\lambda(\beta/8)MN]$.

expansion is induced by letting $\phi = \phi^{cl} + (1/\sqrt{N})\delta\phi$, and neglecting the terms quadratic and higher in fluctuations, we obtain the following approximate expression for the partition function³:

$$\begin{aligned} Z_{\beta}^{(\lambda)} &\approx \int d\phi e^{-\beta E[\phi]} \approx e^{-\beta E[\phi^{cl}]} \\ E[\phi] &= \frac{1}{2} \sum_{A,B,p,q} \phi_p^A ([\lambda \mathbf{1} - \mathbf{J}]^{-1})_{pq}^{AB} \phi_q^B \\ &\quad - \frac{1}{\beta} \sum_p \ln \sum_A e^{-\beta(\phi_p^A + H_p^A + \frac{1}{2})} \end{aligned} \quad (2.14)$$

Differentiating the $\ln Z_{\beta}^{(\lambda)}[\mathbf{H}]$ with respect to H , and setting $\lambda = 0$, one finds the following semiclassical relationship between \mathbf{m} and $\tilde{\phi} = \phi^{cl} + H$:

$$m_p^A(\tilde{\phi}) \approx \frac{\exp(-\beta \tilde{\phi}_p^A)}{\sum_C \exp(-\beta \tilde{\phi}_p^C)} \quad (2.15)$$

This relationship was written up previously in Peterson and Söderberg (1989) and also in Simić (1990). On the other hand, from the saddle point equation $\delta E/\delta\phi = 0$ obeyed by the semiclassical field ϕ^{cl} , and equation 2.15, one obtains,

$$\phi^{cl} = \mathbf{Jm}(\tilde{\phi}) \quad (2.16)$$

which justifies the name ‘‘potentials’’ for ϕ -variables. This expression is the mean-field version of the generally valid relationship, $\langle \phi \rangle \equiv \int d\phi \phi e^{-\beta E[\phi]} = \mathbf{Jm}$, which can be proved from 2.12, and it shows the meaning of the semiclassical field ϕ^{cl} .

We now have all elements to derive the semiclassical approximation of the *effective cost* function. Using $E[\phi^{cl}]$, and the defining relation 2.8, we find (at $\lambda = 0$):

$$C_{\beta}^{\text{eff}}[\mathbf{m}] \approx \frac{1}{2} \sum_{A,B,p,q} J_{pq}^{AB} m_p^A m_q^B - \frac{1}{\beta} \left[\sum_{A,p} \beta \tilde{\phi}_p^A m_p^A + \sum_p \ln \sum_A e^{-\beta \tilde{\phi}_p^A} \right] \quad (2.17)$$

where we have used equation 2.16 and the previously defined $\tilde{\phi} = \phi^{cl} + H$. Since $\tilde{\phi}$ is an implicit function of m , one should consider C^{eff} to be a function of m ; note that H can be obtained from $C^{\text{eff}}[\mathbf{m}]$ using equation 2.10.

³This is the well-known semiclassical or (‘‘naive’’) mean-field approximation. It is the first term in a systematic expansion of the partition function based on order-by-order inclusion of the effects of thermal fluctuations, and it is formally an expansion in powers of $1/N$. Unfortunately, for spin-glass and similar optimization problems this approximation does not become exact in the thermodynamic (large N) limit. (Thouless *et al.* 1977). The exact result is obtained (in the spin-glass case) by including the next term (quadratic in fluctuations). It is not clear, in general, how important these corrections are from the neural computation point of view, where the degree of improvement in the quality of approximate solutions should be measured relative to the computational complexity of the ‘‘corrected’’ algorithms.

The C^{eff} has the expected $C - TS$ form. The first term is just the original cost function 2.5, while the term in the brackets is the entropy function of the constrained net.⁴ One should note the second term in the brackets. Its sum-log-sum form is characteristic of the elastic net (Durbin and Willshaw 1987). In the elastic net case (originally applied to the TSP), a similar term was shown to be the consequence of enforcing strongly the constraints that each city be visited by one and only one node of the discretized elastic string (Simić 1990).⁵ Here, it is a consequence of enforcing strongly the constraint that each data node should be matched to one and only one model node.

One could follow the analogy with Hopfield's equations (Hopfield 1984), and minimize the C_{β}^{eff} by postulating the relaxation equations for $\phi -$ potentials, $\dot{\phi} = -\phi + \mathbf{Jm}(\phi)$, with $m_p^A(\phi)$ given by relation 2.15. Instead, we will write down the relaxation equations directly for $m_p^A(t)$.

$$\delta m_p^A(t) = -\delta t \left(m_p^A(t) - \frac{\exp(-\beta \sum_{B,q} J_{pq}^{AB} m_q^B(t))}{\sum_B \exp(-\beta \sum_{C,q} J_{pq}^{BC} m_q^C(t))} \right) \tag{2.18}$$

It can be shown that this time evolution converges toward the minima of the *effective cost* function 2.17, that is, $\mathbf{m}(t) \approx_{t \rightarrow \infty} \langle \boldsymbol{\eta} \rangle$. To prove this, consider how the C^{eff} change with time under the time evolution specified by 2.18. Using the update equation 2.18, and the $m(\phi)$ relationship 2.15, we find,

$$\begin{aligned} \frac{\delta C_{\beta}^{\text{eff}}}{\delta t} &= -\sum_{p,A} \left(\tilde{\phi}_p^A - \sum_{B,q} J_{pq}^{AB} m_q^B \right) \frac{\delta m_p^A}{\delta t} \\ &= \sum_{A,B,p} \frac{1}{z_p y_p} \left(\mathbf{Jm} - \tilde{\phi} \right)_p^A \left(e^{-\beta(\mathbf{Jm})_p^A} e^{-\beta \tilde{\phi}_p^A} - e^{-\beta(\mathbf{Jm})_p^B} e^{-\beta \tilde{\phi}_p^B} \right) \\ &= \sum_{p,A \neq B} c_p^{AB} \left[\left(\mathbf{Jm} - \tilde{\phi} \right)_p^B - \left(\mathbf{Jm} - \tilde{\phi} \right)_p^A \right] \\ &\quad \left[e^{-\beta(\mathbf{Jm} - \tilde{\phi})_p^B} - e^{-\beta(\mathbf{Jm} - \tilde{\phi})_p^A} \right] \leq 0 \end{aligned} \tag{2.19}$$

where z_p , y_p , and c_p^{AB} are manifestly positive expressions.⁶ Since the constrained net entropy function, given by the expression in the brackets

⁴Had we enforced all the constraints softly (by penalty terms in 2.5), and summed over all possible configurations of the binary event variables, we would obtain the familiar neural network energy function (Hopfield 1984), and the bracket term could be reduced to the more familiar, information-theoretical entropy form (Simić 1990).

⁵In the TSP case, one could reabsorb neural variables (\mathbf{ms}) and express this term in the original (Durbin and Willshaw 1987) form. This is not possible in the case of the general quadratic assignment problem, where \mathbf{J} does not have necessarily some (low-dimensional) geometric interpretation. It is the form of the effective cost function, which "knows" about global constraints, that is in our view the essential ingredient of the elastic net algorithm, and not the ability to further eliminate neural variables (\mathbf{ms}), which rests on the presence of some underlying, low-dimensional geometric interpretation.

⁶ $z_p = \sum_C e^{-\beta \tilde{\phi}_p^C}$, $y_p = \sum_C e^{-\beta(\mathbf{Jm})_p^C}$, and $c_p^{AB} \equiv z_p^{-1} y_p^{-1} e^{-\beta(\tilde{\phi}_p^A + \tilde{\phi}_p^B)}$.

of equation 2.17, is bounded — it is minimum for configurations obeying the constraints 2.3–2.4, and it is close to its maximal value $\alpha N \ln N$ when the $\tilde{\phi}$ are all the same — equation 2.19 shows that the time evolution defined by 2.18 is converging toward the minima of C^{eff} .

It is useful to summarize what has been achieved in this section. We have formulated the quadratic assignment problem probabilistically, using the partition function formalism. The allowed configurations obey the constraints 2.3–2.4, and we have enforced half of these constraints by summing only over the configurations that obey them. This was possible with help of the $\tilde{\phi}$ fields, which are related to the matching probabilities ($m_p^A = \langle \eta_p^A \rangle$) by equations 2.15 and 2.16, and lead to an effective cost function 2.14 depending on $\tilde{\phi}$ s and some external fields (H). Then by the Legendre transformation, we have eliminated the explicit H -dependence, and have constructed the *effective cost* function 2.17. Its minima correspond to the most probable matching events, and 2.18 is a simple relaxation formula without any $\tilde{\phi}$ s that minimizes it. In the following, we will refer to relaxation equations 2.18, as the *constrained net* equations. They are applicable to the most general quadratic assignment problem, which is characterized by an arbitrary symmetric matrix of couplings, J_{pq}^{AB} , and the matching constraints given by equations 2.3–2.4.

2.4 Graph Matching Algorithm and Simulation Results. The final step in generalizing the elastic net algorithm is in choosing the temperature dependence of the penalty parameters, associated with the softly enforced constraints. Specializing here to the graph matching problem, with \mathbf{J} given by equation 2.13, and after some experimentation, we decided for the following T -dependence of the penalty parameters: $\alpha(T) \approx 2\sqrt{\beta}$ and $\gamma(T) \approx 0.1\beta$. This purely phenomenological choice, guarantees that at high T there will be a small penalty for violation of the constraints (2.4, diagonal part) and (2.3, nondiagonal part), but as the optimization temperature decreases toward zero, these penalties gradually increase, eventually becoming infinite, and all the constraints become exactly satisfied — an important characteristic of the elastic net method.

The *constrained net algorithm* works as follows. One starts at some sufficiently high temperature — in our experiments $T \approx 0.2$ – 0.3 was a good starting point — from a random configuration $\mathbf{m}(0)$, uniformly distributed within a small neighborhood around $1/N$, and then iterates equations 2.18 until the equilibrium sets in; we take about 10–20 steps. Then, one decreases the temperature, typically 1–2%, and continues with iterations for another 10–20 steps. This process is repeated until the network converges to its final low-temperature equilibrium state in which all the constraints 2.3–2.4 are satisfied.

The algorithm was applied to both structured (hand-designed graphs such as the one in Fig. 1), and random graphs with up to 100 nodes. In all the graphs we have been studying, the constrained net was capable of

solving the problem *exactly* in a significant fraction of the total number of trials. In general, the optimal solution is discovered very quickly, say in the first 50–100 steps, or the algorithm runs many iterations (few thousands) before converging to a local minimum, often far from the optimal solution. Figure 1 presents two isomorphic graphs, typical of the hand-designed graphs we have studied. Probability of successful match was 0.39, and the convergence was achieved, in such cases, in less than 60 iteration steps. We observed that part of the strategy our algorithm is using is based on searching for the *notable* nodes, that is, nodes with the highest or the lowest connectivity, and matching them first; if this is done successfully, the network would rapidly converge to the final, optimal match state.

In another set of experiments we generated an ensemble of graphs with given number of nodes (N) and some expected connectivity (ρ). For example, in the experiments with $N = 50$ and $\rho = 0.1$ we generate 512 different (pairs of isomorphic) graphs, by randomly generating links between the nodes with probability 0.1 (the expected number of links per node is in this case 5, but the actual number is randomly distributed about this value). For each pair of graphs, we run the algorithm once and record if the algorithm has found the optimal solution or not. At the end, we count how many times the exact solution has been found, and divide this number by the total number of runs. This gives us a probabilistic estimate, $P(N, \rho)$, of how successful in the average, is our algorithm on randomly generated (N, ρ) graphs. Figure 2 summarizes our results for $N = 20, 30, 40, 50,$ and 75 ; for each N , ρ takes number of values between 0.05 and 0.60. Notice the sigmoid-shaped curve indicating essentially 100% success on any size graph with connectivity within interval 0.4–0.6, and decreasing rate of success (but always significantly higher than zero) as the connectivity becomes smaller (or larger). An almost 100% matching success rate is somewhat surprising. It can be partially understood by noticing that the dispersion of the connectivity is proportional to $N\rho(1 - \rho)$ and takes the largest value around $\rho \approx 0.5$. This means that the distribution of the connectivities is rather broad in this case, with few nodes on the extreme ends of its tail having widely different connectivities. An interesting matching heuristic would be to try to match these nodes first. As we have already mentioned, this appears to be the strategy adapted by our network; typically after just 20 to 30 iterations, the *notable* nodes are identified and matched; all the remaining nodes would then be rapidly matched and the algorithm would converge.

This strategy appears very useful in matching similar graphs. We find that if we start with the two isomorphic graphs with expected connectivity around 0.5, and distort one of them by deleting randomly up to 10% of its links, there is no significant degradation in matching performance — the success rate is close to 1. As distortion become larger, at some point the locally stable states appear (corresponding to the wrong

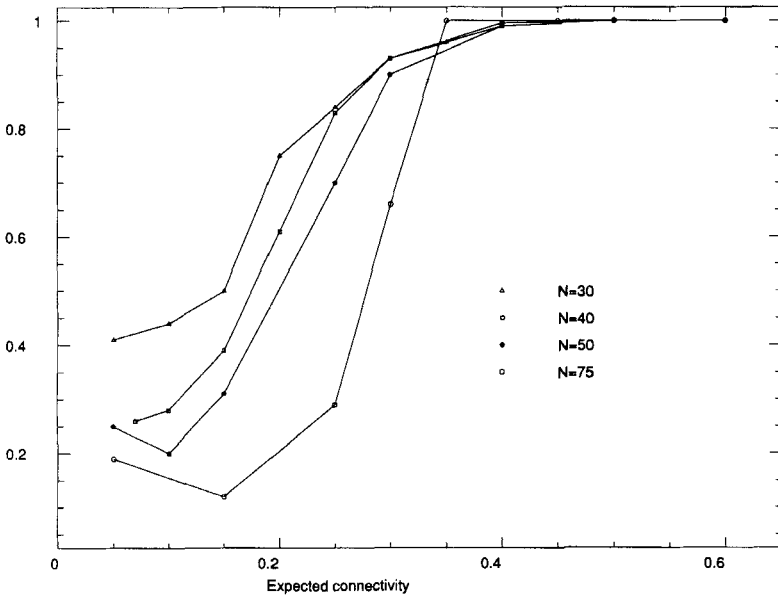


Figure 2: Probability that the constrained net finds the correct match between the two isomorphic graphs with $N = 30, 40, 50,$ and 75 nodes, versus the expectation value of the (random) connectivity per node. (Expected connectivity normalized to one.)

match), and the probability of success decreases rapidly. Basically, we believe that as long as the connectivity of the *notable* nodes is not considerably decreased (in the case of the high connectivity nodes) or increased (in the case of the low connectivity nodes) relative to the other nodes, our algorithm successfully converges towards the correct match. This appears to be the case in many of the examples we studied.

3 Concluding Remarks

The essential ingredients for generalizing the elastic net algorithm to a broad range of constrained optimization problems are (Simić 1990): (1) the probabilistic formulation (formulation based on statistical mechanics), and (2) the explicit enforcement of a significant part of the global constraints when evaluating the *effective cost* function. In this work we have applied these ideas to the quadratic assignment problem. Its important special case, the graph matching problem, is studied in some

detail. Our algorithm appears to be very efficient for matching isomorphic graphs as well as for matching similar graphs (in this case one graph is a simplified version of the other in that some fraction of its links are deleted). It spontaneously develops an interesting matching strategy, based on looking for the most *notable* nodes and matching them first. This strategy seems to be the type of approach that would be useful to humans, and that they might take, in solving similar model matching problems.

Many problems in pattern recognition and computer vision, such as classification and clustering and model and graph matching, can be formulated as assignment problems quadratic in binary variables. The *constrained neural networks* should be applied to these problems since they are "natural" and computationally efficient generalization of both, the elastic net algorithm of Durbin and Willshaw, and the Hopfield and Tank neural networks. One could simply add the constrained net entropy to the original cost function, and choose the penalties for softly enforced constraints to be increasing functions of the inverse temperature. Then the "neural style" relaxation equations 2.18, or any other iterative method, can be used at a sequence of successively decreasing temperatures to find the locally optimal configurations.

Acknowledgments

I thank J. Hopfield, and S. Judd for useful conversations. The indirect inspiration for this paper came also from some conversations with D. Willshaw. This work was supported in part by the DOE under Grants DE-AC03-81ER40050 and DE-FG03-85ER25009, the Office of the Program Manager of the Joint Tactical Fusion Office, and the NSF under Grant EET-8700064.

References

- Cerny, V. 1982. A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Inst. Phys. & Biophys.*, preprint, Comenius Univ., Bratislava.
- Durbin, R., and Willshaw, D. 1987. An analogue approach to the traveling salesman problem using an elastic net method. *Nature (London)* **326**, 689–691.
- Garey, M. R., and Johnson, D. S. 1987. *Computers and Intractability*. W. H. Freeman, New York.
- Hopfield, J. J. 1984. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.* **81**, 3088–3092.
- Hopfield, J. J., and Tank, D. W. 1985. Neural computation of decisions in optimization problems. *Biol. Cybernet.* **52**, 141–152.

- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* **220**, 671.
- Kree, R., and Zippelius, A. 1988. Recognition of topological features of graphs and images in neural networks. *J. Phys. A: Math. Gen.* **21**, L813-L818.
- Mjolsness, E., Gindi, G., and Anandan, P. 1989. Optimization in model matching and perceptual organization. *Neural Comp.* **1**, 218-229.
- Peterson, C., and Söderberg, B. 1989. A new method for mapping optimization problems onto neural networks. *Int. J. Neural Syst.* **1**, 3-22.
- Simić, P. D. 1990. Statistical mechanics as the underlying theory of "elastic" and "neural" optimizations. *Network* **1**, 89-104.
- Thouless, D. J., Anderson, P. W., and Palmer, R. G. 1977. Solution of 'solvable model of a spin glass'. *Phil. Mag.* **35**, 593.
- von der Malsburg, C. 1988. Pattern recognition by labeled graph matching. *Neural Networks* **1**, 141-148.
- von der Malsburg, C., and Bienenstock, E. 1986. *Statistical Coding and Short-Term Synaptic Plasticity: A Scheme for Knowledge Representation in the Brain*, pp. 247-252. Springer-Verlag, Berlin.
- Willshaw, D. 1990. Personal communication.
- Yuille, A. 1990. Generalized deformable templates, statistical physics and matching problems. *Neural Comp.* **2**, 1-24.