To appear in *The Handbook of Brain Research and Neural Networks, (2nd edition), M.A. Arbib (ed.), Bradford Books/The MIT Press.* LU TP 98-16

Neural Optimization

Carsten Peterson and Bo Söderberg

Introduction

Many combinatorial optimization problems require a more or less exhaustive search to achieve exact solutions, with the computational effort growing exponentially or worse with system size. Various kinds of heuristic methods are therefore often used to find reasonably good solutions.

The artificial neural network (ANN) approach falls within this category. In contrast to most other methods, the ANN approach does not fully or partly explore the discrete state-space. Rather, it "feels" its way in a fuzzy manner through an interpolating, continuous space towards good solutions, and allows for a probabilistic interpretation. Key elements in this approach are the *mean-field* (MF) approximation (Hopfield and Tank, 1985; Peterson and Söderberg, 1989), *annealing*, and for many problems the *Potts* formulation (Peterson and Söderberg, 1989). Recently, also *propagator* methods have proven most valuable for handling topological complications (Lagerholm, Peterson, and Söderberg, 1997; Häkkinen, Lagerholm, Peterson, and Söderberg, 1998).

The ANN method gives high-quality solutions to a variety of problems. In addition, it has the appealing feature that the MF equations can be directly implemented in VLSI (see ANALOG VLSI FOR NEURAL NETWORKS), facilitating hardware implementations.

Recurrent Networks

Recurrent networks appear in the context of associative memories (Hopfield, 1982) and difficult optimization problems (Hopfield and Tank, 1985; Peterson and Söderberg, 1989). Simple models for magnetic systems ("spin glasses") have a lot in common with recurrent networks – with an atomic spin seen as analogous to the "firing" state of a neuron – and have therefore been the source of much inspiration for neural network studies. The Hopfield model (Hopfield, 1982) is based on the energy function

$$E = -\frac{1}{2}\sum_{i\neq j} w_{ij} s_i s_j \tag{1}$$

in terms of binary variables (*Ising* neurons) $s_i{=}\pm 1$ (or $0,1$), with symmetric weights w_{ij}. With an appropriate choice of weights depending on the stored patterns, the model serves as an associative memory, with an asynchronous dynamics that locally minimizes E:

$$s_i(t+1) = \operatorname{sgn}\left(\sum_{j\neq i} w_{ij} s_j(t)\right) \tag{2}$$

(see COMPUTING WITH ATTRACTORS).

Optimization with Ising Neural Networks

The archetype of an ANN for optimization is based on a Hopfield-type energy function, adapted to a specific problem by a dedicated choice of weights. With a slightly modified, softer, MF dynamics, with sgn($\cdot$) replaced by tanh($\cdot/T$), combined with annealing ($T \searrow 0$), the resulting MF neurons will relax to a stable configuration representing a tentative solution to the problem. The key problem here is to reach the global minimum or at least a very low-lying local minimum.

Below we will discuss the binary case, illustrating all steps in some detail with the graph bisection (GB) problem. Systems with generic multi-state (Potts) spins instead of binary Ising spins can be treated in a similar way (Peterson and Söderberg, 1989), and will be discussed later.

The Graph Bisection Problem

The neural approach is particularly transparent for graph bisection because of its binary nature. The problem is defined as follows. A graph of N nodes is to be partitioned into two halves, such that the cut-size (the number of connections between the two halves) is minimal. (Figure 1A).

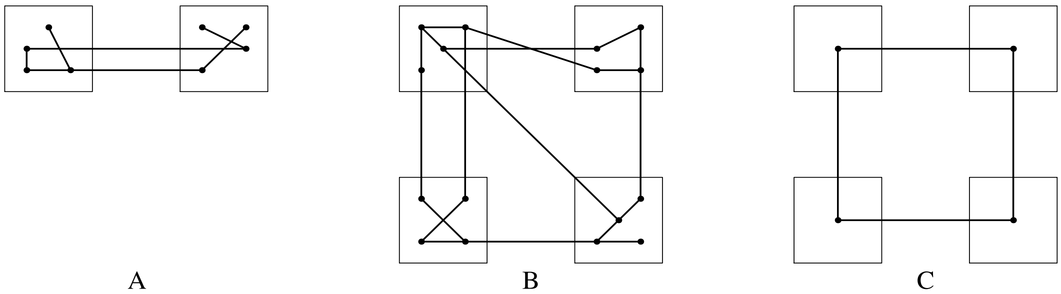

Figure 1: **A**, a graph bisection problem. **B**, a $K = 4$ graph partition problem. **C**, an $N = 4$ TSP problem.

This problem can be encoded as follows. To each node i a binary neuron $s_i = \pm 1$ is assigned, representing whether the node is in the left or right partition in terms of Figure 1A. The graph defines a connection matrix J: For each pair of vertices i, j, J_{ij} is set to 1 if they are connected, and to 0 if they are not (or if they are identical).

With this notation, $J_{ij} s_i s_j \neq 0$ only for a connected pair i, j, being 1 if they are in the same partition and -1 if not. Thus, apart from a constant, the cut-size is measured by $-1/2 \sum_{ij} J_{ij} s_i s_j$. In addition, one needs to add a *global constraint term* to the energy function that penalizes situations where the nodes are not equally partitioned. Since $\sum s_i = 0$ only for a balanced partition, a term proportional to $(\sum s_i)^2$ will subsequently do the trick. Discarding the constant diagonal part $\sum_i s_i^2 = N$, we obtain a Hopfield energy function, Equation 1, with $w_{ij} = J_{ij} - \alpha\,(1 - \delta_{ij})$:

$$E = -\frac{1}{2}\sum_{ij} J_{ij} s_i s_j + \frac{\alpha}{2}\left(\left(\sum_i s_i\right)^2 - \sum_i s_i^2\right) \tag{3}$$

where the *constraint coefficient* α sets the relative strength of the balancing term.

The generic form of Equation 3 is

$$E = \text{Cost} + \text{Global constraint} \tag{4}$$

which is typical when casting combinatorial optimization problems onto neural networks. The origin of the difficulty inherent in this kind of problem is very transparent here; the problem is frustrated because of the competition between the two terms (cost and global constraint). This often leads to the appearance of many local minima.

The next step is to find an efficient procedure for minimizing E, such that local minima are avoided as much as possible.

The Mean-Field Equations

If one attempts to minimize E according to a local optimization rule (e.g. iterating Equation 2), the system will very likely end up in some local minimum close to the starting point, which is not desired.

A better strategy is to employ a stochastic algorithm that allows for uphill moves. One such method is SIMULATED ANNEALING (SA), in which configurations are generated according to the Boltzmann distribution

$$P[s] = \frac{1}{Z} e^{-E[s]/T} \tag{5}$$

with neighborhood search methods. In Equation 5, Z is the *partition function*

$$Z = \sum_{[s]} e^{-E[s]/T} \tag{6}$$

and T is a "temperature" representing the noise level of the system. For $T \searrow 0$ the Boltzmann distribution becomes concentrated to the configuration minimizing E. If configurations are generated with a slowly decreasing T (annealing), they are less likely to get stuck in local minima than if T is set to 0 from the start. Needless to say, such a procedure can be very CPU consuming.

The MF approach aims at approximating the stochastic SA method with a set of deterministic equations. To this end, introduce for each spin s_i a new variable v_i, living in a linear space (in this case $\mathcal{R}$) containing the compact state-space (± 1) of the spin, and set it equal to the spin with a Dirac delta function. Then Z takes the form

$$Z = \sum_{[s]} \int d[v] e^{-E[v]/T} \prod_i \delta(s_i - v_i) \tag{7}$$

Fourier-expanding the delta functions in terms of conjugate variables u_i gives

$$Z \propto \sum_{[s]} \int d[v] \int d[u] e^{-E[v]/T} \prod_i e^{u_i(s_i - v_i)} \tag{8}$$

Then carry out the sum over $[s]$, and write the product as a sum in the exponent:

$$Z \propto \int d[v] \int d[u] e^{-E[v]/T - \sum_i u_i v_i + \sum_i \log\cosh u_i} \tag{9}$$

The original partition function is now rewritten entirely in terms of the new variables $[u, v]$, with an effective energy in the exponent. So far no approximation has been made.

We next assume that Z in Equation 9 is dominated by the extremal value of the integrand, occurring for v_i satisfying the mean-field (MF) equations

$$v_i = \tanh\left(-\frac{\partial E[v]}{\partial v_i}/T\right) \equiv \tanh\left(\frac{1}{T}\sum_j w_{ij} v_j\right) \tag{10}$$

The resulting MF variables v_i can be seen as approximations to the thermal averages $\langle s_i \rangle_T$ of the original binary spins.

The MF equations 10 are solved iteratively, either synchronously or asynchronously, under annealing in T. This yields a deterministic dynamics, characteristic of a recurrent ANN. High temperatures correspond to very smooth sigmoids $\tanh(\cdot/T)$, while in the low-temperature limit the step function of Equation 2 is recovered.

Mean-Field Dynamics

The MF dynamics of such a recurrent ANN typically exhibits a behavior with two phases (Figure 2): at large temperatures, the system relaxes into a trivial fixed point $v_i^{(0)} \approx 0$. As

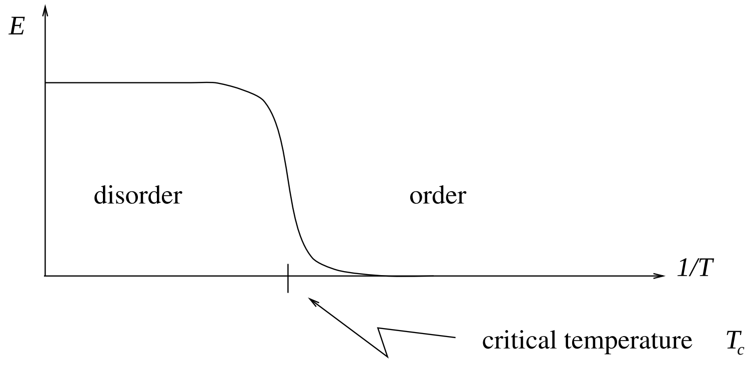

Figure 2: E as a function of T illustrating a phase transition.

the temperature is lowered a bifurcation point, T_c, is passed where $v_i^{(0)}$ becomes unstable, and as $T \to 0$ non-trivial fixed points, $v_i^{(*)} \to \pm 1$, emerge, representing a specific decision made as to the solution of the problem in question (Figure 3).

The position of T_c, which for GB depends on J_{ij} and α, can be estimated by linearizing Equation 10 around the trivial fixed point $v_i^{(0)} = 0$, i.e. by replacing the sigmoid function (tanh) by its argument. For synchronous updating it is clear that the fixed point becomes

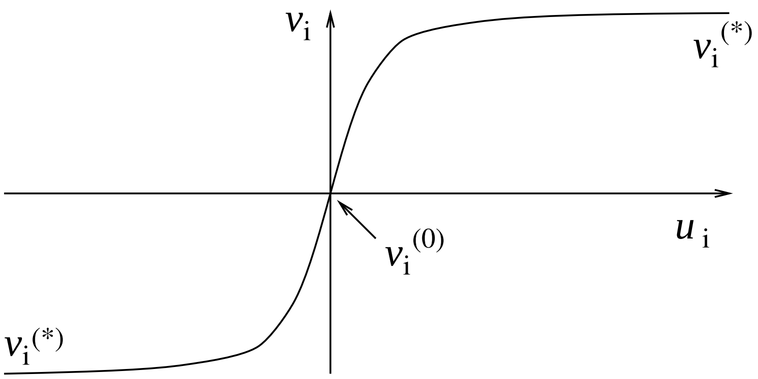

Figure 3: Fixed points in $v_i = \tanh(u_i)$.

unstable as soon as an eigenvalue of the matrix w_{ij}/T, governing the linearized dynamics, is > 1 in absolute value. Prior estimation of the largest eigenvalues is simple to do; apart from determination of T_c, this is also important for avoiding oscillatory behavior, resulting for eigenvalues < -1. In the case of serial updating, the philosophy is the same but the analysis slightly more complicated; the main change (in the absence of diagonal terms) is that the fixed point can only loose stability with eigenvalue $+1$, barring oscillatory behavior (Peterson and Söderberg, 1989).

Optimization with Potts Neural Networks

For GB and many other optimization problems, an encoding in terms of binary elementary variables is natural. However, there are many problems where the natural elementary decisions are of the type one-of-K with $K > 2$.

Early attempts to approach such problems by neural network methods used *neuron multiplexing* (Hopfield and Tank, 1985), where for each elementary K-fold decision, a set of K binary 0/1-neurons was used, with the additional constraint that precisely one of them be on (=1). These *syntax* constraints were implemented in a soft way as penalty terms. As it turned out in the original work on the traveling salesman problem (Hopfield and Tank, 1985), as well as in subsequent investigations for the graph partition problem (Peterson and Söderberg, 1989), this approach does not yield high-quality solutions in a parameter-robust way.

An alternative encoding is to use *Potts neurons* with the syntax constraint built in. In this way the dynamics is confined to relevant parts of the solution space (Figure 4), leading to a dramatically improved performance.

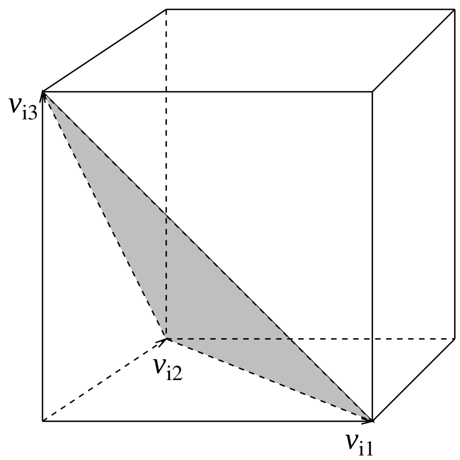

Figure 4: The volume of solutions corresponding to the neuron multiplexing encoding for $K = 3$. The shaded plane corresponds to the solution space of the corresponding Potts encoding.

Potts Spins

A K-state Potts spin is a variable that has K possible values (states). For our purposes, the best representation is in terms of a vector in the Euclidean space $\mathcal{E}_K$. Thus, denoting a spin variable by $\mathbf{s} = (s_1, s_2, \ldots, s_K)$, the ath possible state is given by the ath principal unit vector, defined by $s_a = 1$, $s_b = 0$ for $b \neq a$. These vectors point to the corners of a regular K-simplex (see Figure 4 for the case of $K = 3$). They are all normalized and mutually orthogonal, and fulfill in addition the syntax constraint $\sum_a s_a = 1$.

The MF equations for a system of Potts spins $\mathbf{s}_i$ with a given energy function $E(\mathbf{s})$ are derived following the same path as in the Ising neuron case: Rewrite the partition function as an integral over $\mathbf{u}_i$ and $\mathbf{v}_i$, and approximate it with the maximum value of the integrand. One obtains

$$u_{ia} = -\frac{\partial E(\mathbf{v})}{\partial v_{ia}}/T \tag{11}$$

$$v_{ia} = \frac{e^{u_{ia}}}{\sum_b e^{u_{ib}}} \tag{12}$$

from which it follows that the *Potts neurons* $\mathbf{v}_i$, which approximate the thermal average of $\mathbf{s}_i$, satisfy

$$v_{ia} > 0, \qquad \sum_a v_{ia} = 1 \tag{13}$$

One can think of the neuron component v_{ia} as the *probability* for the ith Potts spin to be in state a. For $K = 2$ one recovers the formalism of the Ising case in a slightly disguised form.

As in the binary case, one can make a linear stability analysis to obtain an estimate for the critical temperature T_c.

The Graph Partition Problem

A generalization of graph bisection is *graph partitioning* (GP): An N-node graph, defined by a symmetric connection matrix $J_{ij} = 0, 1$, $i \neq j = 1, \ldots, N$, is to be partitioned into K subsets of N/K nodes each, while minimizing the cut-size, i.e. the number of connected node pairs winding up in different subsets (see Figure 1B).

This problem is naturally encoded in terms of Potts spins as follows: For each node $i = 1, \ldots, N$, a K-state Potts spin, $\mathbf{s}_i = (s_{i1}, \ldots, s_{iK})$, is assigned, with the component s_{ia} taking the value 1 or 0 depending on whether node i belongs to subset a or not. A suitable energy function is then given by (cf. Equation 3)

$$E = -\frac{1}{2} \sum_{i,j=1}^{N} J_{ij} \mathbf{s}_i \cdot \mathbf{s}_j + \frac{\alpha}{2} \left(\left(\sum_{i=1}^{N} \mathbf{s}_i \right)^2 - \sum_{i=1}^{N} \mathbf{s}_i^2 \right) \tag{14}$$

where the first term is a cost term (cut-size), while the second is a penalty term with a minimum when the nodes are equally partitioned into the K subsets. Note that the diagonal contributions are subtracted in the second term; these are irrelevant for the encoding but affect the dynamics.

The Traveling Salesman Problem

In the traveling salesman problem (TSP) the coordinates $\mathbf{x}_i \in \mathcal{R}^2$ of a set of N cities are given. A closed tour of minimal total length is to be chosen such that each city is visited exactly once. This problem is somewhat reminiscent of (the trivial) $K = N$ graph partition (see Figure 1C). We define an N-state Potts neuron $\mathbf{s}_i$ for each city $i = 1, \ldots, N$, such that the component s_{ia} ($a = 1, \ldots, N$) is 1 if city i has the tour number a and 0 otherwise. Let d_{ij} be the distance between cities i and j. Then a suitable energy function is given by

$$E = \sum_{i,j=1}^{N} d_{ij} \sum_{a=1}^{N} s_{ia} s_{j(a+1)} + \frac{\alpha}{2} \left(\left(\sum_{i=1}^{N} \mathbf{s}_i \right)^2 - \sum_{i=1}^{N} \mathbf{s}_i^2 \right) \tag{15}$$

where the first term is a cost term, and the second a soft constraint term penalizing configurations where two cities are assigned the same tour number.

Scheduling Problems

Scheduling problems have a natural formulation in terms of Potts neurons. In its purest form, a scheduling problem has the simple structure: For a given set of events, a time-slot and a location are to be chosen, each from a set of allowed possibilities, such that no clashes occur. Such a problem consists entirely in fulfilling a set of basic no-clash constraints, each of which can be encoded as a *penalty term* that will vanish when the constraint is obeyed.

In many realistic scheduling applications, however, there exist additional preferences within the set of legal schedules, that lead to the appearance also of *cost terms*. In (Gislén, Peterson, and Söderberg, 1992b), a set of real-world scheduling problems was successfully dealt with using a straight-forward MF Potts formalism.

Refinements and Generalizations

In this section, we will discuss possible complications that often arise in optimization applications and require special care in one way or another.

Non-Quadratic Energy Functions

Not all optimization problems can be formulated in terms of a quadratic energy function, even though the state-space can be encoded in terms of a set of Potts neurons. This presents no principal difficulty, and one can still use Equation 11. However, a possible practical problem arises from the induced self-couplings in the energy function (terms with non-linearities in a single spin), that might affect performance (Peterson and Söderberg, 1989).

With a quadratic E self-couplings can be avoided by removing all diagonal terms, $s_{ia}s_{ib} \rightarrow \delta_{ab}s_{ia}$. Such a procedure can be generalized to any polynomial E. Although in principle any energy-function of a finite number N of Potts spins can be rewritten as a polynomial of at most degree N, this can be difficult in practice for large N.

An efficient and general method for avoiding self-couplings altogether is to replace the derivative in Equation 11 by a difference:

$$u_{ia} = -\frac{1}{T}\left(E(\mathbf{v})|_{\mathbf{v}_i=\mathbf{e}_a} - E(\mathbf{v})|_{\mathbf{v}_i=\mathbf{0}}\right) \tag{16}$$

where $\mathbf{v} = (v_{i1}, \ldots, v_{iK})$, and $\mathbf{e}_a$ is the principal unit-vector in the a-direction. Whenever E is free of self-couplings, Equations 11 and 16 are equivalent.

Inequality Constraints

In the problems mentioned in the previous section, the constraints considered were all of the *equality* type, $f(s) = 0$, that could be implemented with quadratic penalty terms $\propto f(s)^2$. However, in many optimization problems, in particular those of resource allocation type, one has to deal with *inequalities*. An inequality constraint, $g(s) \leq 0$, can be implemented with a penalty term e.g. proportional to

$$\Phi(g) = g\Theta(g) \tag{17}$$

with Θ the Heaviside step function: $\Theta(x) = 1$ if $x > 0$ and 0 otherwise. Of course, such a non-polynomial term in the energy function must be handled using Equation 16.

A problem category with inequality constraints is the *knapsack problem*, where one has a set of N *items* i with associated *utilities* c_i and *loads* a_{ki}. The goal is to fill a "knapsack" with a subset of the items such that their total utility,

$$U = \sum_{i=1}^{N} c_i s_i \tag{18}$$

is maximized, subject to a set of M load constraints,

$$\sum_{i=1}^{N} a_{ki} s_i \leq b_k, \qquad k = 1, \ldots, M \tag{19}$$

defined by load *capacities* $b_k > 0$. In (Ohlsson, Peterson, and Söderberg, 1993), a set of difficult random knapsack problems were approached with a neural method, based on the energy function

$$E = -\sum_{i=1}^{N} c_i s_i + \alpha \sum_{k=1}^{M} \Phi \left(\sum_{i=1}^{N} a_{ki} s_i - b_k \right) \tag{20}$$

in terms of binary spins $s_i \in \{1, 0\}$, representing whether or not item i goes into the knapsack. In (Ohlsson and Pi, 1997) this approach was extended to multiple knapsacks and generalized assignment problems, using Potts MF neurons.

Routing Problems

Many *network-routing* problems can be conveniently handled in a Potts MF approach. We will briefly illustrate the method on the simple shortest-path problem: Given a set of N cities connected by a network of roads, find the shortest path from city a to city b.

Regarding the cities as nodes in a network, and denoting by d_{ij} the length of a direct path, an *arc*, between cities i and j, the problem is equivalent to finding the shortest sequence of arcs leading from node a to b.

This problem can be solved in polynomial time using e.g. the Bellman-Ford (BF) algorithm (Bellman, 1958), where every node i estimates its distance D_{ib} to b, minimized with respect to the choice of a continuation node j among its neighbors (nodes directly connected to i via an arc):

$$D_{ib} = \min_j \left(d_{ij} + D_{jb} \right), \quad i \neq b \tag{21}$$

while $D_{bb} = 0$. Iteration of Equation 21 gives convergence in less than N steps, and D_{ab} can be read off.

By introducing a Potts MF neuron to handle the neighbor choice for each node, we obtain a softer version of the BF algorithm,

$$D_{ib} = \sum_j v_{ij} \left(d_{ij} + D_{jb} \right), \qquad \text{with} \qquad v_{ij} = e^{-\left(d_{ij} + D_{jb}\right)/T} \tag{22}$$

In the limit of $T \to 0$, a strict minimum is obtained, and BF is recovered.

In a similar way also more complex routing problems can be formulated in terms of an optimal local neighbor choice, that naturally lends itself to a MF recast. A nice feature of such an approach is the *locality* inherited from BF: The information needed for the neighbor choice is local to the node and its neighbors.

In (Häkkinen, Lagerholm, Peterson, and Söderberg, 1998) a set of communication routing problems, based on various combinations of unicast and multicast in a network with arcs of finite capacity, were approached along these lines.

The Propagator

Typical of the type of approach sketched above, is the use of *node-node* MF Potts neurons v_{ij} for the local direction of a fuzzy path from a node to its neighbors.

As a convenient tool for monitoring the global topological properties of fuzzy paths, a propagator formalism has been devised (Häkkinen, Lagerholm, Peterson, and Söderberg, 1998), defined in terms of a propagator matrix $\mathbf{P} = (\mathbf{1} - \mathbf{v})^{-1}$, or

$$P_{ij} = \delta_{ij} + v_{ij} + \sum_k v_{ik} v_{kj} + \sum_{kl} v_{ik} v_{kl} v_{lj} + \ldots \tag{23}$$

In the absence of loops, P_{ij} will be 1 if there exists a path from i to j, and 0 if not. For the case of fuzzy paths, the propagator has an obvious probabilistic interpretation. In addition, the propagator will diverge in the presence of loops, and can therefore be used to avoid loop formation.

By using suitable methods for doing the matrix-inversion required for the computation of the propagator elements in a clever way, the increase in computational demand can be kept manageable.

As an example of the use of an analogous formalism in a somewhat different setting, see (Lagerholm, Peterson, and Söderberg, 1997), where a set of semi-realistic airline-crew scheduling problems was treated, with v_{ij} in Equation 23 controlling the transfer of a crew from a flight i to a connecting flight j.

Hybrid Approaches

A large class of optimization problems can be viewed as *parametric assignment* problems, containing elements of both discrete assignment and parametric fitting to given data, e.g. using templates with a known structure. The assignment part can then be encoded in terms of Potts neurons, while the template part may be formulated in terms of a set of continuous, adjustable parameters. This leads to a kind of hybrid approach, *deformable templates*.

Track finding in high energy physics, where circular or spiral-shaped tracks are to be extracted from a set of data points, is an example where this approach is suitable (Ohlsson, Peterson, and Yuille, 1992; Gyulassy and Harlander, 1991). In addition, certain pure assignment problems with a well-defined geometric structure can be recast in this form; an example is given by TSP (see CONSTRAINED OPTIMIZATION AND THE ELASTIC NET).

Results and Summary

For a large class of combinatorial optimization problems, a straight-forward ANN approach can be used, with the following basic steps:

- Map the problem onto a neural network by a suitable encoding of the solution space and an appropriate choice of energy function. Where applicable, use a Potts encoding to avoid Ising neuron multiplexing.
- Compute a suitable starting temperature, e.g. by doing a linear stability analysis of the MF dynamics.

- While annealing, solve the MF equations iteratively.
- When the system has settled, the solutions are checked with respect to constraint satisfaction. If needed, one may perform a simple corrective post-processing, or rerun the system (possibly with modified constraint coefficients).

This very general approach has been numerically explored for all problem types discussed above: Graph bisection (Peterson and Anderson, 1988), graph partitioning and TSP (Peterson and Söderberg, 1989, Peterson, 1990), high school scheduling (Gislén, Peterson, and Söderberg, 1992b), knapsack problems (Ohlsson, Peterson, and Yuille, 1992; Ohlsson and Pi, 1997), crew scheduling (Lagerholm, Peterson, and Söderberg, 1997) and routing (Häkkinen, Lagerholm, Peterson, and Söderberg, 1998), in the last two cases in combination with a propagator formalism. Whenever possible, test beds have been used where comparisons with other approximative approaches are feasible.

From these comparisons the following general picture emerges: For a variety of problem types and sizes, the MF method consistently performs, without excessive fine-tuning, roughly in parity with the comparison methods, which are typically more mature and have often been designed to perform well for a particular problem class. With a prior estimate of T_c, convergence is consistently achieved after a very modest number (typically 50–100) of iterations, independently of problem size.

For parametric assignment problems like track-finding, a deformable templates method is preferable; this hybrid approach can be applied also to certain low-dimensional geometrical assignment problems, like TSP.

A binary (Ising) neuron can be considered as a vector living on a "sphere" in one dimension. The MF approach can be generalized to variables defined on spheres in higher dimensions. Such *rotor* neurons can be used in geometrical optimization problems involving angular variables (Gislén, Peterson, and Söderberg, 1992a).

The MF approximation can be viewed as a variational approach – the true energy $E(s_i)$ is approximated by a trial one $E_0(s_i; u_i)$, which is optimized with respect to the variational parameters u_i. This procedure can be used in a wide range of situations not necessarily confined to discrete optimization problems (Jönsson, Peterson, and Söderberg, 1993).

References

[Bellman, 1958] Bellman, R., 1958, On a routing problem, *Quarterly of Appl. Math.*, 16:87–90.

[Gislén, Peterson, and Söderberg, 1992a] Gislén, L., Peterson, C., and Söderberg, B., 1992, Rotor neurons: Basic formalism and dynamics, *Neural Computat.*, 4:737–745.

[Gislén, Peterson, and Söderberg, 1992b] Gislén, L., Peterson, C., and Söderberg, B., 1992, Complex scheduling with Potts neural networks, *Neural Computat.*, 4:805–831.

[Gyulassy and Harlander, 1991] Gyulassy, M., and Harlander, H., 1991, Elastic tracking and neural network algorithms for complex pattern recognition, *Comput. Phys. Commun.*, 66:31–46.

[Häkkinen, Lagerholm, Peterson, and Söderberg, 1998] Häkkinen, J., Lagerholm, M., Peterson, C., and Söderberg, B., 1998, A Potts neuron approach to communication routing, *Neur. Computat.*, 10:1587-1599.

[Hopfield, 1982] Hopfield, J.J., 1982, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79:2554–2558.

[Hopfield and Tank, 1985] Hopfield, J.J., and Tank, D.W., 1985, Neural computation of decisions in optimization problems, *Biol. Cybern.*, 52:141–152.

[Jönsson, Peterson, and Söderberg, 1993] Jönsson, B., Peterson, C., and Söderberg, B., 1993, A variational approach to correlations in polymers, *Phys. Rev. Lett.*, 71:376–379.

[Lagerholm, Peterson, and Söderberg, 1997] Lagerholm, M., Peterson, C., and Söderberg, B., 1997, Airline crew scheduling with Potts neurons, *Neur. Computat.*, 9:1589–1599.

[Ohlsson, Peterson, and Söderberg, 1993] Ohlsson, M., Peterson, C., and Söderberg, B., 1993, Neural networks for optimization problems with inequality constraints: The knapsack problem, *Neural Computat.*, 5:331–339.

[Ohlsson, Peterson, and Yuille, 1992] Ohlsson, M., Peterson, C., and Yuille, A., 1992, Track finding with deformable templates: The elastic arms approach, *Comput. Phys. Commun.*, 71:77–98.

[Ohlsson and Pi, 1997] Ohlsson, M., and Pi, H., 1997, A study of the Mean Field Approach to Knapsack Problems, *Neur. Netw.*, 10:263–271.

[Peterson, 1990] C. Peterson, 1990, Parallel distributed approaches to combinatorial optimization problems: Benchmark studies on TSP, *Neural Computat.*, 2:261–269.

[Peterson and Anderson, 1988] Peterson, C., and Anderson, J. R., 1988, Neural Networks and NP-complete Optimization Problems: A Performance Study on the Graph Partition Problem, *Compl. Syst.*, 2:59–89.

[Peterson and Söderberg, 1989] Peterson, C., and Söderberg, B., 1989, A new method for mapping optimization problems onto neural networks, *Int. J. Neural Syst.*, 1:3–22.