Combinatorial Optimization with Feedback Artificial Neural Networks*

Carsten Peterson
Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden
`carsten@thep.lu.se`

Abstract

A brief review is given for using feedback artificial neural networks (ANN) to obtain good approximate solutions to combinatorial optimization problems. The key element is the mean field approximation (MFT) The methodology, which is illustrated for the graph bisection and knapsack problems, is easily generalized to Potts systems. The latter is related to the deformable templates method, which is illustrated with the track finding problem. MFT is based on a variational principle, which also can be generalized to non-integer problems.

Introduction

Many combinatorial optimization problems are NP-complete, which require state space explorations leading to $O(a^N)$ computations for a system with N degrees of freedom. Different kinds of heuristic methods are therefore often used to find reasonably good solutions. The ANN approach falls within this category. Whereas the use of ANN for pattern recognition and prediction problems is a non-linear extension of conventional linear interpolation/extrapolation methods, ANN in the optimization domain really brings something new to the "table". In contrast to existing search and heuristics methods the ANN approach does not fully or partly explore the different possible configurations. Rather it "feels" its way in a fuzzy manner towards good solutions. This is done in a way that allows for a statistical interpretation of the results. The key element in this approach is the mean field approximation (MFT) [1, 2], which can be regarded as a variational scheme. The three basic steps involved,

$$\boxed{\text{Encode problem}} \rightarrow \boxed{\text{Linearize MFT dynamics}} \rightarrow \boxed{\text{Solve MFT equations}}$$

will be dealt with briefly using the graph bisection and knapsack problems

*Proceedings of ICANN '95 International Conference on Artificial Neural Networks, October 1995, Paris, France , eds. F. Fogelman-Soulie and P. Gallinari, EC2 & Cie (Paris 1995).

as examples. The latter requires some extra care since it contains inequality constraints. For low dimensional geometrical problems like the traveling salesman problem (TSP) and track finding a variant of the "pure" neural approach, *deformable templates*, is advantageous to use. Recent efforts to use variational methods for computing polymer configurations are also discussed. These optimization problems are not combinatorial but the techniques involved are strongly related to MFT.

Ising Neurons – the Graph Bisection Problem

The neural approach is particularly transparent for this problem due to its binary nature. Consider a set of N nodes to be partitioned into two halves such that the connectivity (cutsize) between the two halves is minimized (see fig. 1).

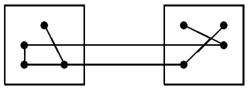

Figure 1: A graph bisection problem.

The problem is mapped onto the energy function [3]

$$E[s] = -\frac{1}{2}\sum_{ij}\omega_{ij}s_i s_j - \frac{\alpha}{2}(\sum_i s_i)^2 \tag{1}$$

where for each node a binary neuron $s_i = \pm 1$ is assigned depending on whether node i is in the left or in the right position and $\omega_{ij} = 0, 1$ encodes whether i and j are connected or not. The first term in eq. (1) minimizes the connections between partitions, whereas the second penalizes unbalanced configurations ($\sum s_i \neq 0$). The imbalance parameter α sets the relative strength between the cutsize and the balancing term.

The next step is to find an efficient procedure for minimizing eq. (1), such that local minima are avoided as much as possible. Rather than doing this with a discrete updating rule using simulated annealing [4], the MFT approximation is used. This amounts to approximate the stochastic simulated annealing procedure with iterative solutions to a set of deterministic equations

$$v_i = \tanh\left(-\frac{\partial E[v]}{\partial v_i}\frac{1}{T}\right) = \tanh\left(\sum_j(\omega_{ij} - \alpha)v_j/T\right) \tag{2}$$

These represent fixed point solutions to the circuit equations of ref. [1]. It is advantagous to know in advance the appropriate range in T – the phase transition temperature T_c. The dynamics of eq. (2) exhibits a behavior with two phases separated by $T = T_c$: at large enough temperatures ($T \to \infty$) the system has a trivial fixed point $v_i^{(0)} = 0$ and as $T \to 0$ fixed points $v_i^{(*)} = \pm 1$ emerge representing a specific solution (see fig. 2). The position of T_c, which

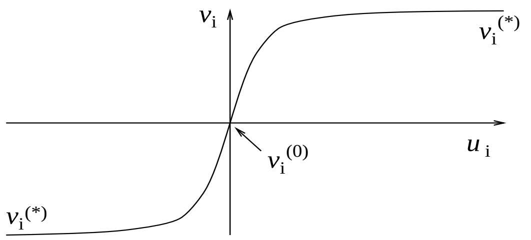

Figure 2: Fixed points in $\tanh(u_i)$.

depends on ω_{ij} and α, can be estimated by expanding the sigmoid function in a power series around $v_i^{(0)} = 0$ (see fig. 2). For synchronous updating it is clear that if one of the eigenvalues of the matrix in eq. (2) is > 1 in absolute value, the fixed point becomes unstable and the solution will wander away into the nonlinear region. In the case of serial updating the philosophy is the same but the analysis slightly more complicated [2]. Finding the largest eigenvalue of the matrix is easy – multiply the matrix with itself a few times. From this procedure one reads off T_c. One can always add from an encoding point of view auxiliary diagonal terms in order to modify the eigenvalues and thereby control the dynamics (i.e. avoid cyclic behavior). This linearized dynamics analysis thus both gives us means to control dynamics and to remove one free parameter – the temperature.

The Knapsack Problem

The graph partition problem is characterized by an *equality* constraint, which is implemented with a polynomial penalty term. However, in many optimization problems, one has to deal with *inequalities*. One such problem category 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,

$$U = \sum_{i=1}^{N} c_i s_i \ , \tag{3}$$

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

$$\sum_{i=1}^{N} a_{ki} s_i \leq b_k \ , \ k = 1, M \ , \tag{4}$$

defined by load *capacities* $b_k > 0$. The encoding is in terms of binary decision variables (spins) $s_i \in \{1, 0\}$, representing whether or not item i goes into the knapsack. In the class of problems, where a_{ki} and c_i are independent uniform random numbers on the unit interval $[0, 1]$, while b_k are fixed to a common value b, the most difficult case is given by $b = N/4$. In the optimal solution to such a problem, there will be a strong correlation between the value of c_i and the probability for s_i to be 1. With a simple heuristic based on this observation near-optimal solutions are easily obtained. One therefore also consider a class of harder problems with more narrow c_i distributions – *homogeneous* problems.

The problem is mapped onto an energy function [6],

$$E = -\sum_{i=1}^{N} c_i s_i + \alpha \sum_{k=1}^{M} \Phi \left(\sum_{i=1}^{N} a_{ki} s_i - b_k \right) \quad , \tag{5}$$

where Φ is a penalty function ensuring that the constraint in eq. (4) is fulfilled. An appropriate choice of $\Phi(x)$ is $x\Theta(x)$. For alternative *slack* variable techniques to deal with inequality constraints see e.g. ref. [7].

Minimizing eq. (5) is done with the MFT equations. Due to the non-polynomial form of the constraint special care is required here. The derivative $\partial E/\partial v_i$ in eq. (2) is replaced by a difference. In fact, this trick can be used to kill self-couplings for any energy function, not only those containing inequality constraints.

$$\frac{\partial E}{\partial v_i} \rightarrow -c_i + \alpha \sum_{k=1}^{M} \left[\Phi(\sum_{j=1}^{N} a_{kj} v_j - b_k)|_{v_i=1} - \Phi(\sum_{j=1}^{N} a_{kj} v_j - b_k)|_{v_i=0} \right] \tag{6}$$

In ref. [6] this ANN approach is compared with other approaches. For small problem sizes it is feasible to use an exact algorithm, **Branch and Bound (BB)**, for comparison. This tree search technique is accelerated by ordering the c_i's according to magnitude: $c_1 > c_2 > .... > c_N$. For larger problem sizes, one is confined to other approximate methods, such as **Simulated Annealing (SA)** [4], **Greedy Heuristics (GH)** and simplex based on **Linear Programming (LP)**. With GH one also benefits from non-homogeneous problems by proceeding from larger to smaller c_i collecting every item not violating constraints. LP is not designed to solve discrete problems like the knapsack one. However, for ordered non-homogeneous knapsack problems LH gives solutions with a set of leading 1's and a set of trailing 0's, with a window in between containing real numbers. Augmented by greedy heuristics for the elements in this window, fairly good solutions emerge. In table 1 the ANN, SA, GH and LP ap-

Method	Scaling	c_i=rand[0,1]		c_i=rand[0.45,0.55]		c_i=0.5	
		Perf.	CPU	Perf.	CPU	Perf.	CPU
BB	2^N	1	1	1	1	1	1
ANN	NM	0.98	0.05	0.95	0.0005	0.97	0.0005
SA	NM	0.98	0.05	0.95	0.0005	0.96	0.0005
LP	N^2M^2	0.98	0.006	0.93	0.0002	0.93	0.0002
GH	NM	0.97	0.001	0.88	0.00001	0.85	0.00001

Table 1: Comparison of relative performance and CPU time consumption for the different algorithms for N=M=30 problems.

proaches are compared with the exact BB for N=M=30 non-homogeneous and homogeneous problems. As expected LP and in particular GH benefits from non-homogeneity both quality- and CPU-wise, while for homogeneous problems the ANN algorithm performs very well. When comparing the different approximate approaches for larger problem sizes (50 to 500) the conclusions are the same – the real strength in the ANN approach is best exploited for more homogeneous problem. Similar conclusions were drawn in [8], where the knapsack problem was generalized to the *General Assignment Problem*, which contains more than one knapsack and hence involves Potts spins. Also in ref. [8] a hybrid scheme, where the ANN method is used to resolve the "twilight zone" that is not populated by integer s_i with LP, is found to be quite successful.

Potts Neurons

For problems that are not binary, e.g. graph partition, TSP and scheduling problems, the Ising neuron procedure above is easily generalized to Potts neurons [2, 5]. The MFT equations for Potts neurons s_{ia} satisfying $\sum_a s_{ia} = 1$ reads

$$v_{ia} = \frac{e^{u_{ia}}}{\sum_b e^{u_{ib}}} \tag{7}$$

where $u_{ia} = 1/T(\partial E/\partial v_{ia})$. Very competitive solution qualities have been obtained using Potts neurons for realistic scheduling problems [5].

Probability Propagators

For resource allocation problems with non-trivial topologies type airline crew scheduling, straightforward encodings do not contain a fixed number of neuronic (probability) factors as e.g. in the graph bisection case (eq. (1)). It is then convenient to work with propagators P_{ij} in terms of Potts neurons v_{ij} [9]

$$P_{ij} = \left[\frac{\mathbf{1}}{\mathbf{1}-\mathbf{v}}\right]_{ij} = 1 + v_{ij} + \sum_k v_{ik}v_{kj} + \sum_{kl} v_{ik}v_{kl}v_{lj} + ... \tag{8}$$

where e.g. v_{ij} is the probability for a crew connecting flight i with j.

Rotor Neurons

A binary (Ising) neuron can be considered as a vector living on a "sphere" in one dimension. The MFT approach can be generalized to variables defined on spheres in higher dimensions. Such *rotor* neurons may be used in geometrical optimization problems with angular variables ([10]).

Consider the general problem of minimizing an energy function $E(\vec{s}_1, \ldots, \vec{s}_N)$ with respect to a set of N D-dimensional unit vectors (rotors), $\vec{s}_i \in \mathcal{R}^D$, $|\vec{s}_i| = 1$, $i = 1, \ldots, N$. Along the same lines as in the binary (Ising) and Potts cases previously discussed, one derives the corresponding MFT equations

$$\vec{v}_i = \mathbf{g}(\vec{u}_i) \equiv \hat{u}_i g\left(\frac{1}{T}\nabla_i E(\vec{v})\right) \tag{9}$$

where $\hat{u_i} = \vec{u}_i/u_i$, and g is a modified sigmoid

$$g(u) = \frac{\log I_{D/2}(u)}{\log I_{(D-2)/2}(u)} \tag{10}$$

I_n is the modified Bessel function of order n. For $D = 1$ the standard sigmoid (cf. eq. (2)) is recovered. Again eq. (9) is iterated by annealing in the temperature T. At high temperature the system is in a symmetric phase, characterized by a stable trivial fixpoint $\vec{v}_i \approx 0$. At lower T, this becomes unstable, and the mean fields $\vec{v}_i$ will be repelled by the origin. For low enough temperature they will stabilize close to the sphere $\vec{v}_i^2 = 1$.

The dynamics is thus very different from that of conventional methods, where moves typically take place on the surface. The ability of exploring an "off-shell" interpolating state space also here provides an additional dimension through which spurious local minima can be esccess from. The method was successfully to configurational problems in ref. [10].

Deformable Templates

For low-dimensional geometric optimization problems it is often advantageous to use the deformable templates approach, thereby reducing the number of degrees of freedom. This approach has been successfully applied to TSP [11] and the particle physics track finding problem [12, 13]. The latter is used here for illustration.

Track finding is the problem of assigning signal points to smooth curves. In a pure neural approach one would use Potts neurons $s_{ij} = \pm 1$ to encode whether signals i and j are connected or not and construct $E(s_{ij})$ such that smooth tracks with no bifurcations correspond to minima. In its basic form with no heuristics this approach requires N^2 degrees of freedom for N signals. Even though this approach [14, 15] works well it may not be the optimal way to proceed. The method is more general than what is needed for this problem since the parametric form of the tracks are known in advance – straight lines or helices. One should take advantage of this prior knowledge as a human being would do. This is illustrated in fig. 3 with signal points generated from 15 straight tracks with 100% noise. One detects the tracks in the noisy environment by inspecting the image in fig. 3 by holding it up looking for straight lines. Furthermore, for a problem with N signals and M tracks one should only need O(M) degrees of freedom.

In the *deformable templates* approach [12] [13] the observed points are matched to simple parametrized models, templates, the form of which reflects the *a priori* knowledge about the possible track shapes – helices passing through the origin (collision point). In addition, the formalism allows for the data points corresponding to noise to be unmatched. The mechanism involved is closely related to redescending M-estimators used in Robust Statistics. A Hough transform is used to give initial conditions for the templates and to specify the number of templates required. Then this set of M templates defined by the parameters $\vec{\pi}_a$ are processed with the elastic arms method that

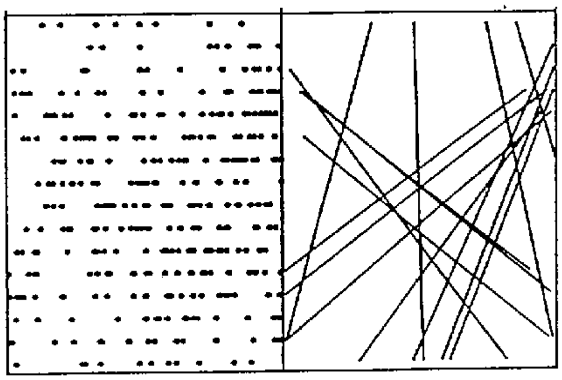

Figure 3: (a) Signal points from 15 generated straight tracks with 100% noise. (b) The corresponding solution. From ref. [13].

resolves ambiguities and makes detailed fits to the signals by minimizing

$$E\left[v_{ia};\vec{\pi}_a\right]=\sum_{i,a}(M_{ia}-\lambda)v_{ia} \tag{11}$$

where $\vec{x}_i$ is the position of the i:th signal, M_{ia} is a measure of the (squared) distance between the i:th signal and the a:th template. In eq. (11) v_{ia} is 1 if the i:th point is assigned to the a:th arm, and 0 otherwise, and subject to constraint $\sum_a v_{ia}=0,1$. Given i there should be *at most* one a such that $v_{ia}=1$. This allows for noise signals not being assigned to any track.

$E[v_{ia},\vec{\pi}_a]$ is minimized with respect to $\vec{\pi}_a$ subject to the modified Potts constraint $\sum_a v_{ia}=0,1$. A Boltzmann distribution is introduced in order to avoid local minima. Summing over allowed configurations for v_{ia}, one obtains the marginal probability

$$P_M[\vec{\pi}_a]=\frac{1}{Z}e^{-\beta E_{eff}[\vec{\pi}_a]} \tag{12}$$

Gradient descent on E_{eff} gives for $\Delta\pi_a^{(k)}$ (k=1,2,3,..)

$$\Delta\pi_a^{(k)}=-\eta\frac{\partial E_{eff}}{\partial\pi_a^{(k)}}=-\eta\sum_i\hat{V}_{ia}\frac{\partial M_{ia}}{\partial\pi_a^{(k)}} \tag{13}$$

$$\hat{V}_{ia}=\frac{e^{-\beta M_{ia}}}{e^{-\beta\lambda}+\sum_b e^{-\beta M_{ib}}} \tag{14}$$

In the $T\rightarrow 0$ limit these equations are nothing but the k-means clustering updating equations (apart from the noise factor).

How does this algorithm work? Initially the template arms are placed according to the Hough transform. The templates compete for the signals by means of Gaussian distributions of width $T=1/\beta$ centered around the arm positions. At first each arm can attract many signals. The relative importance of the different signals is measured by the Potts factor (eq. (14)). As the temperature is lowered the different arms are attracted more exclusively only to nearby signals. The constant λ governs the amount of noise points or *outliers* the algorithm allows for. It enters the Potts factor (eq. (14)) like an extra "null" component contributing to the denominator. The domain of attraction of the arms is cut off (for small T) at a distance $\sqrt{\lambda}$.

The performance of the Elastic Arms algorithm is competitive as compared to conventional tracking routines.

Non-integer Optimization Problems

The MFT approximation is a variational approach – one approximates the true energy E with a trial one, $E_0(u_i)$, and minimizes the *free energy* with respect to the variational parameters u_i. In the Ising neuron case the latter are identified with the local fields $\sum_j \omega_{ij} v_j$ using the linear ansatz $E_0 = \sum_i u_i v_i$. This procedure can be generalized to non-integer problems with appropriate assumptions for the form of E_0. In particular harmonic trials have turned out to be profitable. In [16] a variational algorithm for computing correlations in polymers was developed. For an N-atom polymer with Coulombic self-repulsion one makes the variational ansatz

$$E_0 = \frac{1}{2} \sum_{ij} G^{-1}{}_{ij} (\vec{x}_i - \vec{a}_i) \cdot (\vec{x}_j - \vec{a}_j) \tag{15}$$

with $\vec{a}_i$ and G_{ij}^{-1} as variational parameters. The results for the important end-to-end correlations are impressive when compared to the CPU demanding Monte Carlo method. This method is quite general and can be used in a variety of situations including Mixed Integer Programming problems.

Summary

Using feedback neural networks to find good solutions to combinatorial optimization problems is an approach which in its nature is very different from other approximate methods. Appetizing features include automatic prior setting of parameters and probability interpretation of the resulting MFT neuronic values. With respect to quality of the solutions the ANN methods give rise to very competitive results as compared to those of other approximate schemes. The only exception being TSP, where "classical" algorithms like Lin-Kernighan are hard to beat [17]. On the other hand TSP may not be a very typical resource allocation problem. The extensive comparisons with other approaches for the knapsack problem [6] are particularly interesting since they illuminate under what circumstances (lack of structure in the problem) the ANN method is a winner.

References

[1] J.J. Hopfield and D.W.Tank, "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics* **52**, 141 (1985).

[2] C. Peterson and B. Söderberg, "A New Method for Mapping Optimization Problems onto Neural Networks", *International Journal of Neural Systems* **1**, 3 (1989).

[3] C. Peterson and J.R. Anderson, "Neural Networks and NP-complete Optimization Problems; A Performance Study on the Graph Bisection Problem", *Complex Systems* **2**, 59 (1988).

[4] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by Simulated Annealing", *Science* **220**, 671 (1983).

[5] L. Gislén, B. Söderberg and C. Peterson, "Complex Scheduling with Potts Neural Networks", *Neural Computation* **4**, 805 (1992).

[6] M. Ohlsson, C. Peterson and B. Söderberg, "Neural Networks for Optimization Problems with Inequality Constraints - the Knapsack Problem", *Neural Computation* **5**, 331 (1993).

[7] A.H. Gee and W. Prager" Polyhedral Combinatorics and Neural Networks", *Neural Computation* **6**, 161 (1993).

[8] H. Pi and M. Ohlsson, "A Study of the Mean Field Approach to Knapsack Problems", *LU TP 95-7* (submitted to *Neural Networks*).

[9] M. Lagerholm, C. Peterson and B. Söderberg, in press.

[10] L. Gislén, B. Söderberg and C. Peterson, "Rotor Neurons - Basic Formalism and Dynamics", *Neural Computation* **4**, 727 (1992); C. Peterson, O. Sommelius and B. Söderberg, "A Variational Approach for Minimizing Lennard-Jones Energies", *LU TP 94-4* (submitted to *Physical Review* **E**).

[11] R. Durbin and D. Willshaw, "An Analog Approach to the Traveling Salesman Problem using an Elastic Net Method". *Nature* **326**, 689 (1987).

[12] A. Yuille, K. Honda and C. Peterson, "Particle Tracking by Deformable Templates", *Proceedings of 1991 IEEE INNS International Joint Conference on Neural Networks*, Vol. 1, pp 7-12, Seattle, WA (July 1991);
M. Ohlsson, C. Peterson and A. Yuille, "Track Finding with Deformable Templates – The Elastic Arms Approach", *Computer Physics Communications* **71**, 77 (1992).

[13] M. Gyulassy and H. Harlander, "Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition", *Computer Physics Communications* **66**, 31 (1991).

[14] B. Denby, "Neural Networks and Cellular Automata in Experimental High Energy Physics", *Computer Physics Communications* **49**, 429 (1988).

[15] C. Peterson, "Track Finding with Neural Networks", *Nuclear Instruments and Methods* **A279**, 537 (1989).

[16] B. Jönsson, C. Peterson and B. Söderberg, "A Variational Approach to Correlations in Polymers", *Physical Review Letters* **71**, 376 (1993).

[17] D.S. Johnson and L.A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization", to appear in *Local Search in Combinatorial Optimization*, eds. E.H.L. Aarts and J.K. Lenstra, New York 1992: John Wiley & Sons).