ϕ_n is constant over a signalling interval. The value of ϕ_n depends on ϕ_{n-1} during the preceding signalling interval and the symbol transmitted during that signalling interval a_{n-1}. With the further assumption that $h = p/q$, where p and q are relatively prime numbers, coupled with the modulo 2π rendition of the phase, the elements of the set Φ reduce to q distinct values. This reduction in the number of states takes place irrespective of whether p is odd or even. The q distinct states are obtained by a one-to-one mapping of the phase values ϕ_n.

When $q \geq M$, the number of phase states is greater than the number of branches leaving each state; therefore, no parallel branches exist between adjacent states. When $q < M$, the number of states is less than the number of branches leaving each state. This gives rise to parallel branches between states. For integral values of h

$$2\pi h = 4\pi h = \ldots = 2(M-1)\pi h = 0 \quad \text{modulo } 2\pi$$

In this case the set Φ has only one element, and the trellis degenerates to a single state trellis with M parallel branches. Degenerate trellises occur at all weak modulation indices [1] thus reducing the connectivity of the trellis and associated poor distance properties.

Examples of simple trellis and state diagrams for MSK and $h = 1/2$, $1/3$, $1/4$ 4-CPFSK are shown in Fig. 1. These trellis representations are simpler than those found in the literature and can be trivially extended to other CPM schemes.

We have applied this finite-state trellis representation of M-CPFSK to obtain the closed-form expression for the minimum Euclidean distance of M-CPFSK at any modulation index. It turns out to be

$$d^2_{min} = \begin{cases} 2E_s & \text{integer } h \\ \min_\gamma 4E_s(1 - \text{sinc}\,\gamma h) & q \geq M \\ 4E_s(1 - \text{sinc}\,2h) & h < 0{\cdot}301677 \quad q < M \\ 2E_s & h > 0{\cdot}301677 \quad q < M \end{cases}$$

where $\gamma = 2, 4, \ldots, 2(M-1)$.

Conclusion: The number of phase states in CPM is always equal to q, where q is the denominator of the rational value modulation index.

18th October 1991

R. Liyanapathirana (*Memorial University of Newfoundland, Faculty of Engineering and Applied Science, St. John's, Newfoundland A1B 3X5, Canada*)

N. Ekanayake (*University of Peradeniya, Department of Electrical and Electronic Engineering, Peradeniya, Sri Lanka*)

References

1 ANDERSON, J. B., AULIN, T., and SUNDBERG, C.-E. W.: 'Digital phase modulation' (Plenum Publishing Corporation, New York, NY, USA, 1986)

2 FORNEY, G. D., JUN.: 'The Viterbi algorithm', *Proc. IEEE*, 1973, **61**, pp. 268–278

LANDSCAPE RESHAPING ALGORITHM FOR ADDITIVE NEURAL NETWORKS WITH APPLICATION TO GRAPH MAPPING PROBLEMS

K. V. K. Iyer, H. Öğmen and Ç. K. Koç

Indexing terms: Neural networks, Optimisation, Graph mapping

A neural network based algorithm for problems in nonconvex optimisation is described. The algorithm restricts the search by exploiting the properties of the parameter space. It keeps the initial point fixed and reshapes the energy function so that, in the neighbourhood of an arbitrary initial condition, basins of attraction are formed corresponding to valid solutions first and then these basins are made deeper to improve the quality. This algorithm was applied to both the travelling salesman problem (TSP) and the graph mapping problem (GMP) and good results were obtained.

Introduction: Neural network (NN) approaches to nonconvex optimisation problems involve two steps:

(i) finding a representation that translates the NN activity into a possible solution

(ii) establishing a relationship between the cost and constraints of the problem and the Lyapunov (energy) function of an absolutely stable NN.

Unlike approaches that conduct the search in a fixed energy landscape [1], we propose a method that reshapes the energy landscape to obtain desirable minima in the neighbourhood of an arbitrary initial point.

We use the representation proposed by Hopfield and Tank [5] for both of the optimisation problems that we consider, namely the travelling salesman problem (TSP) and the graph mapping problem (GMP) [2]. In the TSP, we want to find the shortest tour visiting each city only once and returning to the initial city. In the GMP, we want to find an adjacency preserving map between the nodes of a computational graph and those of a parallel computer. The NN consists of an array of neurons where each column and row corresponds, respectively, to a position in the tour (a node of the parallel computer architecture) and to a city (a node of the graph) for the TSP (GMP). A valid tour (mapping) corresponds to steady-state activations that form a permutation matrix. Neurons obey recurrent additive equations. The following is a Lyapunov function for the high-gain symmetric network [3, 5]:

$$E = \frac{A}{2}\sum_X\sum_i\sum_{j\neq i} V_{Xi}V_{Xj} + \frac{B}{2}\sum_i\sum_X\sum_{X\neq Y} V_{Xi}V_{Yi} + \frac{C}{2}\left(\sum_X\sum_i V_{Xi} - n\right)^2 + \frac{D}{2}\Gamma \quad (1)$$

where V_{Xi} is the output of the neuron at row X and column i, and n is the number of columns (or rows) in the network. A, B, C and D are positive constants. For the TSP, we have

$$\Gamma = \sum_X\sum_{Y\neq X}\sum_i d_{XY}V_{Xi}(V_{Yi+1} + V_{Yi-1})$$

with d_{XY} being the distance between cities X and Y. For the GMP, we choose

$$\Gamma = \sum_X\sum_{Y\neq X}\sum_i\sum_{j\neq i} d_{XiYj}V_{Xi}V_{Yj}$$

where d_{XiYj} represents the distance corresponding to the mapping of the Xth and Yth graph nodes to the ith and jth hypercube nodes, respectively. Given two adjacent nodes X and Y on the graph, if i and j are adjacent then this distance is 1. Otherwise, the distance grows with the distance between i and j.

The above form of the Lyapunov function makes explicit the constraints and the cost of the optimisation problem. The terms weighted by A and B impose the constraint 'at most one neuron active per column and row' (inhibition terms for validity); the term weighted by C imposes the constraint that a total number of n neurons should be active at steady-state (global bias for validity). These two constraints together force the network toward a valid solution (a permutation matrix). Finally, the term weighted by D represents the cost. The choice of the parameters A, B, C and D plays a crucial role in the shape of the energy landscape and thus in the ability of the network to find a good solution.

Parametric studies: We conducted extensive numerical studies to explore the properties of equilibria of the network in the

parameter space. Our results can be summarised as follows:

(i) *Validity (A, B, C) against quality (D) in the large:* In general there is a tradeoff between the quality and the percentage of valid solutions. As one can see from eqn. 1, making A, B, C much larger than D causes a domination of the validity constraint and results in a larger number of valid solutions. However, because the last term plays only a small role, the quality of the solution is in general very poor. On the other hand, a much larger D than A, B and C results in the opposite tendency.

(ii) *Inhibition (A, B) against global bias (C):* Such a study was conducted by Hegde *et al.* on some specific TSPs [4]. These authors claimed the existence of three linearly separable regions in the parameter space. We conducted a more general study on random TSPs and Fig. 1 illustrates a typical result. There are two (and not three) major regions in the space (the regions are not linearly separable): one mainly composed of overcomplete solutions (more than n neurons active) and the second mainly composed of undercomplete (less than n neurons active) and valid solutions. Moreover, as the size of the problem increases, the latter shrinks (this tendency was also found in the study by Hegde *et al.*).

(iii) *Validity against quality in the small:* To analyse local properties of the quality-validity tradeoff, we computed and compared the quality of valid solutions found in the regions identified above. Our results indicate that the region containing more valid solutions also contains solutions of better quality. Thus, as pointed out in (i) although validity and quality are in mutual conflict for extreme choices of the parameters weighing them, a 'good balance' in validity constraints leads to a synergism between validity and quality properties of the equilibria.

Landscape reshaping algorithm: We decided to integrate our findings into an algorithm. The main idea behind the algorithm is to find iteratively the parameter values A, B, C and D that yield a synergism between validity and quality. Because the parameter space contains two regions, one with overcomplete solutions and the other with a mixture of undercomplete and complete (valid) solutions, and because the latter contains solutions of between quality, the first step is to make changes in A, B, C and D to move into the latter region.

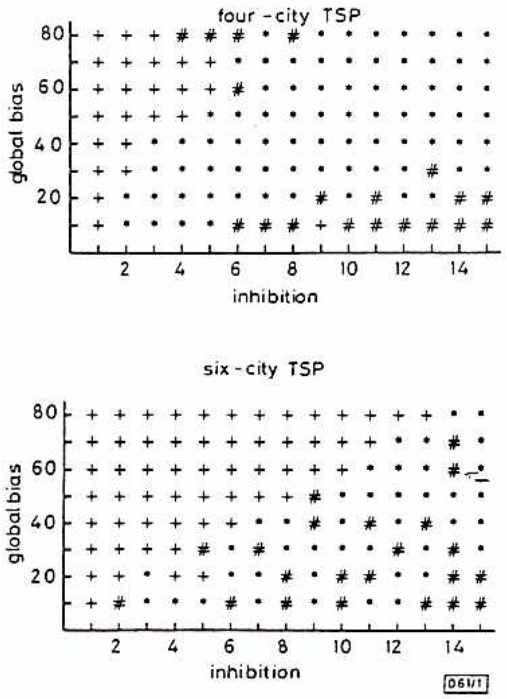


Fig. 1 *Global bias against inhibition for Hopfield-Tank data set*
+: overcomplete tours
#: complete tours
*: undercomplete tours

Once we are in this region we keep A, B and C fixed and improve the quality of the solution by increasing D. The algorithm keeps the initial point for the trajectory of the additive network fixed and reshapes the energy function so that in the neighbourhood of the initial condition we form basins of attraction corresponding to valid solutions first and then make these basins deeper to improve the quality.

Landscape reshaping algorithm:

(*a*) Select A, B, C, D, and the initial point x_0

(*b*) Repeat steps (*c*), (*d*), (i), (ii) until the first valid solution is found or the stopping criteria are met

(*c*) Run the NN from the initial point x_0 until an equilibrium point x_e is reached

(*d*)(i) If x_e is overcomplete then reduce C and D and increase A and B

(*d*)(ii) If x_e is undercomplete then reduce A, B and D and increase C

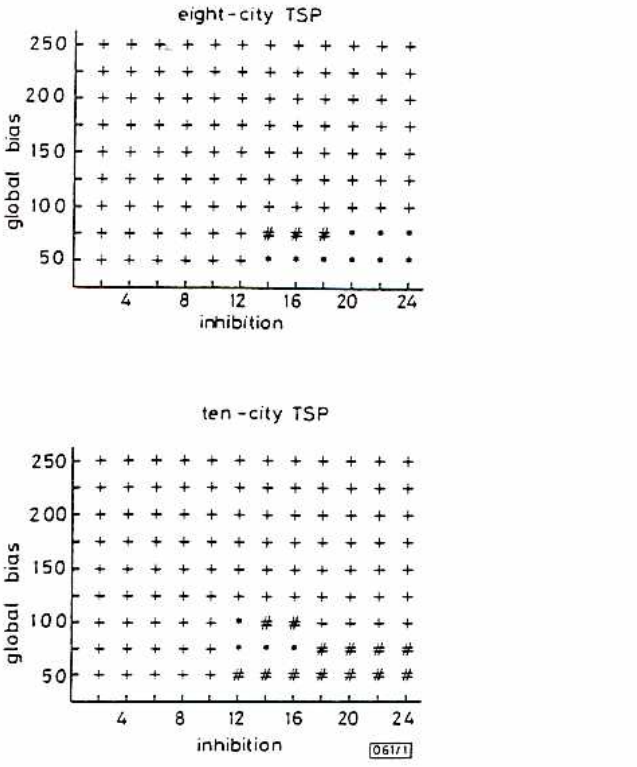
(*e*) Repeat steps (*c*), (*f*), (i), (ii) until the stopping criteria are met

(*f*)(i) If x_e is valid then increase D

(*f*)(ii) If x_e is invalid then decrease D

First some random values for the parameters A, B, C, D, and the initial point x_0 for the NN activations are selected. The user has to specify the following stopping criteria: n is the maximum number of iterations, I is the improvement percentage, and *count* is the number of iterations before the improvement in quality of suboptimal solution is tested. When valid solutions are obtained, the quality of these solutions is stored in an array. If there is no improvement in the quality of the solutions beyond I% in *count* number of iterations, the program is terminated and the best quality solution and the corresponding iteration number is printed. In step (*c*) of the algorithm the neural network searches for a solution: the network is initialised to the same point x_0 and is run until it reaches an equilibrium point.* In steps (*d*)(i) and (ii) we make

* We simulated the network by using a numerical ordinary differential equation solver on a sequential workstation. An efficient implementation can be achieved by use of parallel, analogue neural network VLSI chips



changes in the parameters until we find the first valid solution. These parameter modifications follow from our observations on the Lyapunov function and the parameter space (Fig. 1). For an overcomplete solution, we reduce the global bias and D, and increase the inhibition to reach the undercomplete/complete region. Typical values used for these changes were 20% increase in A and B and 40% decrease in C and D. For an undercomplete solution, we increase the global-bias and reduce the inhibition and D to obtain more active neurons at equilibrium. Typical values used in these changes were 10% reduction in A and B, 5% increase in C and 30% reduction in D. Once we reach the region containing valid/undercomplete solutions and once we find a valid solution in that region we keep the global bias and the inhibition fixed and increase D to get better quality solutions (steps (c) and (f)(i)). The process terminates when the improvement or the iteration limits are reached. It is also possible that on increasing the value of D, we end up with an invalid solution. In that case, we reduce D in small steps until we find a valid solution again. However, if the new value of D is less than its value just before the transition from valid to invalid, then the algorithm exits because the quality of the solution cannot be improved any further. Thus, forward and backward changes in D at steps (f)(i) and (ii) are bounded with extreme points (the lower corresponding to a valid solution and the higher to an invalid solution) and the steps become progressively smaller to avoid oscillations. Typical values for these changes were 50% increase and 20% reduction.

This algorithm was implemented and tested on both the TSP and the GMP. Fig. 2 illustrates our results for a 10-city TSP and an 8-node GMP (mapping of an eight-node rectangular mesh to an eight-node hypercube). This particular GMP constitutes a good benchmark because we know its optimum solution. The quality of the solution is compared for the cases with and without the algorithm. In Fig. 2, the x-axis shows the different cases which include invalid and suboptimal solutions for the TSP and invalid, optimal and suboptimal solutions for the GMP. Along the y-axis is shown the quality of the solution. The quality was calculated as a percentage. For example, for the TSP, we ran the algorithm 100 times and the total number of iterations for all the runs put together was 1748. Out of this, 722 times (41·3%) we obtained an invalid solution, 1026 times (58·69%) we obtained valid solutions with a distance less than 3·5, 408 times (23·3%) we obtained valid solutions with a distance less than 3·0, and twice (0·01%) we obtained valid solutions with a distance less than 2·75. We obtained similar percentages for the case without the algorithm from our study of the parmaeter space (Fig. 1). We made a similar comparison for the GMP. Once again, the quality of the suboptimal solution was calculated as an inverse function of penalty. As one can see from Fig. 2, the algorithm improves simultaneously the validity and the quality of solutions even when an average over all iterations (intermediate results) is taken. Because the final solution from the algorithm is the best of all intermediate results, a significant improvement is achieved. To illustrate this, let us note that on an average, the first valid tour for the TSP was obtained after 6·7 iterations with a standard deviation of 3·42. We obtained the first valid tour for the TSP with $d < 3{\cdot}0$ in 10·73 iterations with a standard deviation of 3·87; the first valid tour with $d < 2{\cdot}75$ in 16·19 iterations with a standard deviation of 3·79. For the GMP, the global optimum was obtained after an average number of 5·91 iterations with a standard deviation of 3·41.

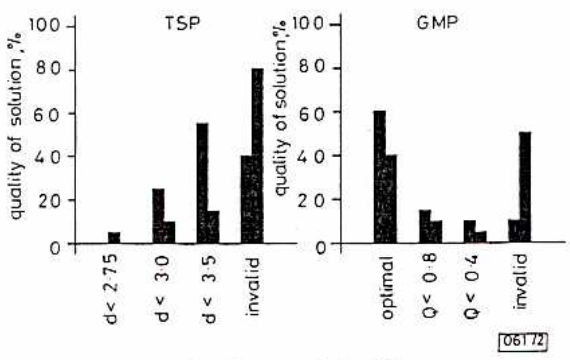


Fig. 2 *Comparison of performance of algorithm*
■ with algorithm
▨ without algorithm

Conclusions: We presented an algorithm which, instead of conducting a stochastic search on different regions of a fixed energy landscape, conducts a search around the neighbourhood of a fixed initial point by iteratively reshaping the energy landscape. The reshaping is achieved by changing the parameters in the energy function based on our knowledge of the parameter space. This selective change enables us to bring basins of attractions of valid solutions to the neighbourhood of the fixed point, and to make these basins deeper to improve the quality. The overall scheme results in an efficient optimisation approach. This efficiency is further increased by the fact that the neural network is massively parallel and is implementable in commercially available analogue neural network VLSI chips.

30th September 1991

K. V. K. Iyer, H. Öğmen and Ç. K. Koç (*Department of Electrical Engineering, University of Houston, Houston, TX 77204-4793, USA*)

References

1 AARTS, E. H. L., and KORST, J. H. M.: 'Simulated annealing and Boltzmann machines' (John Wiley and Sons, New York, 1989)
2 BERMAN, F., and SNYDER, L.: 'On mapping algorithms into parallel architectures', *J. Parallel Distrib. Comput.*, 1987, **4**, pp. 439–458
3 COHEN, M. A., and GROSSBERG, S.: 'Absolute stability of global pattern formation and parallel memory storage by competitive neural networks', *IEEE Trans.*, 1983, **SMC-13**, pp. 815–826
4 HEGDE, S. U., SWEET, J. L., and LEVY, W. B.: 'Determination of parametrs in a Hopfield/Tank computational network'. Proc. IEEE Int. Conf. Neural Networks, 1990, pp. II-291-298
5 HOPFIELD, J. J., and TANK, D. W.: 'Neural computation of decisions in optimization problems', *Biol. Cybern.*, 1985, **52**, pp. 141–152

TUNABLE GREEN UPCONVERSION ERBIUM FIBRE LASER

J. Y. Allain, M. Monerie and H. Poignant

Indexing terms: Lasers, Optical fibres

50 mW of green power and slope efficiency against launched pump power of 15% have been obtained at room temperature in an erbium doped fluorozirconate fibre pumped at 0·97 μm. Lasing tunability domain, pump wavelength tunability and the role of colasing at 1·55 μm are described.

Introduction: The first demonstration of room temperature CW upconversion lasing at 546 nm in an erbium doped fluorozirconate fibre has been reported recently [1]. A green light output power of 23 mW and a maximum slope efficiency against absorbed pump power of 11% were observed. The fibre was pumped around 0·80 μm. The authors noted a strong competition of lasing at 850 nm, leading to saturation of green power with increasing pump power.

We report on experiments on the same system pumped around 0·97 μm with a Ti : sapphire laser. Shifting the pump wavelength from 0·80 to 0·97 μm, where laser diodes are also commercially available, offers some advantages. Mainly, it avoids the drawback of favouring oscillation at 0·85 μm following the scheme $^4I_{13/2} \rightarrow {}^2H_{11/2}$ by pump excited-state absorption (pump ESA) at 0·80 μm, $^2H_{11/2} \rightarrow {}^4S_{3/2}$ by non-radiative relaxation, and $^4S_{3/2} \rightarrow {}^4I_{13/2}$ by laser oscillation at 0·85 μm (refer to Fig. 1 of Reference 1). This pathway tends to squeeze the $^4I_{15/2}$ ground-state level and favours laser oscillation at 0·85 μm. Pumping at 0·97 μm removes this difficulty