# Compact analogue neural network: a new paradigm for neural based combinatorial optimisation

Jayadeva, S.C.Dutta Roy and A.Chaudhary

**Abstract:** The authors present a new approach to neural based optimisation, to be termed as the compact analogue neural network (CANN), which requires substantially fewer neurons and interconnection weights as compared to the Hopfield net. They demonstrate that the graph colouring problem can be solved by using the CANN, with only *O(N)* neurons and $O(N^2)$ interconnections, where *N* is the number of nodes. In contrast, a Hopfield net would require $N^2$ neurons and *0{N*)* interconnection weights. A novel scheme for realising the CANN in hardware form is discussed, in which each neuron consists of a modified phase locked loop (PLL), whose output frequency represents the colour of the relevant node in a graph. Interactions between coupled neurons cause the PLLs to equilibrate to frequencies corresponding to a valid colouring. Computer simulations and experimental results using hardware bear out the efficacy of the approach.

## 1 Introduction

Optimisation problems arise in nearly every aspect of our lives, and most real world tasks involve the minimisation or maximisation of an objective, subject to certain constraints. Combinatorial optimisation applications form a special class, and include some of the most challenging and well-studied problems. Many members of this class are NP-complete, which makes their exact solution infeasible. The search for efficient heuristics to obtain good solutions in a reasonable amount of time has therefore engaged the attention of researchers for many years.

In their papers in the 1980s, Hopfield and Tank [1, 2] showed that a coupled system of neurons, now well known as the 'Hopfield net', converges to a local minimum of an associated energy function. Their approach involves constructing an artificial network with a specific energy function, so that a desired objective can be minimised. The approach has since been applied to literally thousands of problems in a variety of fields [2-10].

A significant hindrance to the effective application of neural based optimisation has been the hardware complexity, which also affects the cost of simulation. To solve an Af-node graph colouring problem, for example, one requires a Hopfield net with *N x N* neurons and $O(A^4)$ connections.

In this paper, we propose a new approach, to be termed as the compact analogue neural network (CANN), and illustrate its application to optimisation tasks with graph colouring, a classical NP-complete problem. However, the proposed approach can be adopted to solve other combinatorial optimisation applications as well.

*IEE Proceedings* online no. 19990314

*DOI:* 10.1049/ip-cds:19990314

Paper first received 27th May 1998 and in revised form 29th January 1999

Jayadeva and S.C. Dutta Roy are with, and A. Chaudhary (deceased) was with the Department of Electrical Engineering, Indian Institute of Technology, Delhi, Hauz Khas, New Delhi - 110016, India

Graph colouring is formulated in a novel manner as an optimisation problem in only *N* variables. An energy function whose minima correspond to valid colourings of a given graph is derived. We also describe how the approach can be extended to other combinatorial optimisation tasks. A scheme for the hardware realisation of the CANN is then discussed. The design requires only *N* neurons and $O(N^2)$ interconnections to solve the most complex graph colouring problem with *N* nodes. Note that *N* colours would be needed to colour a clique of size *N*. Each neuron is constructed by using a modified phase locked loop (PLL), whose frequency represents the neuron's output state, and equivalently, the colour of the corresponding node. With appropriate interconnections, the network of PLLs converges to a set of frequencies which meet the adjacency constraints imposed by the graph. Experimental results using breadboard versions of the hardware corroborate the validity of the CANN design. A simulation model used for larger examples is then discussed. Computer simulations based on a MATLAB model also bear out the efficacy of the approach.

## 2 New approach to combinatorial optimisation

Preliminary ideas of this Section were presented in [11].

A combinatorial optimisation problem can typically be described in terms of a set of *N* variables $V_t$, $i = 1, 2, ..., N$, where *N* is the problem size. Each of the *N* variables needs to be assigned one label from a set of *K* available ones $\{L_b\ L_2, ..., L_k\}$. For example, the travelling salesperson problem (TSP) involves determining the position at which each of *N* cities is visited on a cycle; the assignment task requires finding which of *K* tasks is assigned to each of *N* processors.

The cost function and constraints differ from one application to another. However, without loss of generality, one can assign an integer from 1 to *K* to the available labels. Therefore, the problem reduces to finding the optimal values of $V_b$ where $V_t$ can assume integer values from 1 through *K*. This requirement can be met by imposing the constraints:

$$sin(irVi) \;—0 \qquad i = 1, 2, ..., IV \qquad (1)$$

$$1 \leq V_i \leq K \qquad (2)$$

A combinatorial optimisation task can now be formulated in terms of a set of real valued variables $V_i$ with the incorporation of eqns. 1 and 2. This relaxation permits one to formulate a continuous function whose minima lie only at a discrete set of points. This function is also the 'energy' function of the corresponding CANN.

In contrast, Hopfield net based approaches use $NK$ variables $V_{ij}$ each of which is either 0 or 1. For example, in the TSP, $V_{ij} = 1$ if city $i$ is visited at position $j$. The number of labels $K$ is usually equal to $N$. Therefore, the number of variables, or the number of neurons, is $N^2$. Furthermore, since the energy function of the Hopfield network depends on all variables, all neurons are typically coupled to each other. The number of interconnections is thus $O(N^4)$. In contrast, the number of weights in a CANN would be $O(N^2)$. It is important to point out that, in many applications, the number of available labels is $O(N^2)$ or larger; in such cases, the CANN would lead to greater savings in the numbers of neurons and weights.

The state space of a Hopfield net can be visualised as a hypercube of dimension $NK$, with valid states lying only at a subset of the vertices. The CANN state space can be thought of as a projection of the hypercube onto a lower dimensional space; such a projection yields a lattice of lower dimension. Note that constraints (eqns. 1 and 2) restrict all solutions or minima to lie on an ordered integer lattice.

### 2.1 Graph colouring: the Hopfield net approach

In this Section, we briefly discuss how the Hopfield net is applied to graph colouring. In Section 2.2, we discuss the CANN approach to facilitate a comparison.

Given a set of $N$ nodes and their adjacencies, each node is required to be assigned a colour so that no two adjacent nodes are similarly coloured. Fig. 1*a* shows a graph with its associated adjacency matrix. Fig. 1*b* illustrates the corresponding solution. The Hopfield net based approach requires an array of $N \times N$ neurons. $N$ colours are needed if the nodes form a clique; for a map, the corresponding graph is planar, and can be coloured with at most four colours [12].

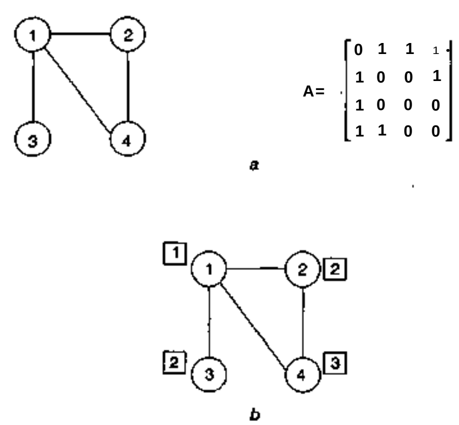

**Fig. 1** *Graph colouring for four-node graph*
*a* Graph with four nodes and its adjacency matrix
*b* Valid colouring of the graph of *a*. The colours are indicated by the numbers in the small squares adjacent to each node

Let $U_{ij}$ and $V_{ij}$ denote the state and output of the neuron in row $i$ and column $j$, respectively, where:

$$V_{ij} = \begin{cases} 1 & \text{if node } i \text{ is assigned colour } j \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Also,

$$V_{ij} = f(u_{ij}) \qquad (4)$$

where $f$ is a squashing function (e.g. the sigmoid or the step), which has values between 0 and 1. Let:

$$d_{ij} = \begin{cases} 1 & \text{if node } i \text{ is adjacent to node } j \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

Then, the graph colouring problem may be formulated as the following task:

$$\text{minimise} \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij}^2 \qquad (6)$$

subject to the constraints

$$\sum_{j=1}^{N} V_{ij} = 1 \qquad i = 1, 2, \ldots, N \qquad (7)$$

$$\sum_{k=1}^{N} d_{ij} V_{ik} V_{jk} = 0 \qquad i, j = 1, 2, \ldots, N \qquad (8)$$

Constraint (eqn. 7) requires each node to be assigned only one colour, while eqn. 8 constrains adjacent nodes to have different colours. Following the work in [13, 14], and by using the theory of sequential unconstrained minimisation techniques (SUMTs) [15, 16], the energy function for graph colouring may be shown to be of the form:

$$E = \frac{A}{2} \sum_{i=1}^{N} \left( \sum_{j=1}^{N} V_{ij} - 1 \right)^2 + \frac{B}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} \left( \sum_{k=1}^{N} V_{ik} V_{jk} \right)^{\alpha} + \frac{C}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij}^2 \qquad (9)$$

where $\alpha = 2$. (Using the theory of SUMTs, it has been shown in [13, 14] that convergence to feasible solutions is **not ensured unless a sequence of functions $E_p$, $p = 1, 2, \ldots$,** is minimised, in which the weights A and B are increased with $p$. This leads to networks in which the weights adapt with time. However, most neural optimisation approaches minimise a time-invariant function.) Many researchers have proposed energy functions similar to eqn. 9 in the literature (e.g. Dahl [17], Moopenn *et al.* [18] and Thakoor *et al.* [19]. Takefuji and Lee [20] used the first two terms of eqn. 9 with $\alpha = 1$). Note that all of these require $O(N^2)$ neurons and $O(N^4)$ interconnections for the most general problem.

### 2.2 Graph colouring with a CANN

Following the discussion at the beginning of this Section, consider an array of $N$ neurons, whose outputs are real numbers $V_i$, $i = 1, 2, \ldots, N$, where $V_i$ denotes the colour of node $i$. For a valid colouring, $V_i$ must be an integer in the range 1 to $N$. The constraints for the graph colouring task may be written as:

$$V_i \neq V_j \qquad \text{if } d_{ij} = 1 \qquad (10)$$

$$\sin(\pi V_i) = 0 \qquad i = 1, 2, \ldots, N \qquad (11)$$

$$1 \leq V_i \leq N \qquad (12)$$

A feasible solution to eqns. 10-12 may be found by minimising

$$E_p = -\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}(V_i - V_j)^2 + \sum_{i=1}^{N} B_{pi}\sin^2(\pi V_i) + \sum_{i=1}^{N} C_{pi}\left[\min(0, V_i - 1)\right]^2 + \sum_{i=1}^{N} D_{pi}\left\{\min\left[0, (N - V_i)\right]\right\}^2 \qquad (13)$$

where 5^;, $C_{pi}$, and D^; are scalars. Note that the first term of eqn. 13 maximises the difference between the colours of adjacent nodes.

By expressing the constraints, particularly eqn. 10, in different ways, several alternatives to eqn. 13 can be formulated. One can employ different objective functions for minimising the number of colours, such as:

$$\sum_{i=1}^{N} V_i^2 \qquad (14)$$

which would tend to minimise the values of colour labels assigned.

At this point, we note that, in graph colouring, as in any other combinatorial optimisation problem, all we really need is to have a consistent labelling; the actual values of the labels are not important, because one can sort the labels and assign any set of integers to them. In graph colouring, we require that the number of distinct labels be minimised, and we therefore choose the objective function:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(1 - d_{ij})(V_i - V_j)^2 \qquad (15)$$

which is a minimum if all nonadjacent nodes have the same colour.

From eqns. 10-15 and following [13, 14], we note that the graph colouring problem can be solved by minimising an energy function of the form:

$$E_p = \sum_{i=1}^{N}\sum_{j=1}^{N}(1 - d_{ij})(V_i - V_j)^2 + \sum_{i=1}^{N} B_{pi}\sin^2(\pi V_i) + \sum_{i=1}^{N} C_{pi}\left[\min(0, V_i - 1)\right]^2 + \sum_{i=1}^{N} D_{pi}\left\{\min\left[0, (N - V_i)\right]\right\}^2 - \sum_{i=1}^{N} F_{pi}\sum_{j=1}^{N} d_{ij}(V_i - V_j)^2 \qquad (16)$$

Energy functions such as eqn. 16 represent the 'energy function' of a corresponding neural network. We discuss the corresponding network in Section 3.

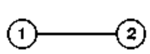


**Fig.** 2 *Graph with two nocks adjacent to each other*

Consider the example of Fig. 2, which shows a graph with two nodes connected to each other. Two colours are needed to colour this graph. Let $V_x$ and $V_2$ denote the colours of the two nodes. Fig. 3 shows the energy function (eqn. 16) for the corresponding neural network, for a chosen set of values of $B_{pb}$ $C_{pi}$, $D_{pi}$, and $F_{pi}$. Note that there are several local minima, but only one global minimum. Fig. 4 shows a two-dimensional projection of the surface. Note that the local minima lie on a regular lattice. For illustration, V\ and $V_2$ have been permitted to lie between 1 and 10.

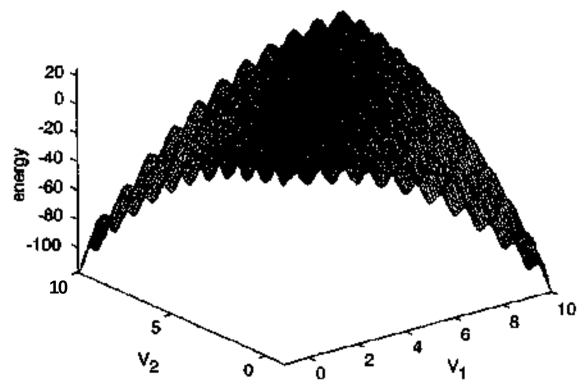


Fig. 3 *Energy surface (eqn. 16)*

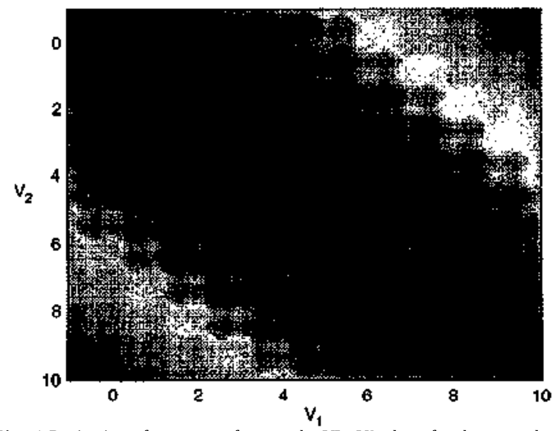


**Fig.** 4 *Projection of energy surface on the VI—V2 plane for the example of Fig. 2*
$B_{pi}$ = 10, $C_{pi}$ = $D_{pl}$ = ^,.= 1,'1=1, 2, ..., N

The minima of eqn. 16 correspond to solutions of the graph colouring problem. These can be determined by using any nonlinear optimisation technique. Applicable techniques include gradient descent, simulated annealing [21] and chaotic annealing [22], among others. Regardless of the procedure employed, our formulation leads to substantial savings in computing time since the number of variables is reduced by *O(N)* in comparison with Hopfield net based approaches, such as eqn. 9.

## 3 New architecture for combinatorial optimisation: the compact analogue neural network

Our focus in this paper is not on the minimisation of the energy functions derived in Section 2.1. Instead, it is demonstrated that the CANN can be realised in an elegant way in hardware. Since the realisation will slightly differ from one application to another, we focus on graph colouring even though the architecture can be adopted to solve other tasks.

As pointed out in Section 2, the actual label values in a combinatorial optimisation application are inconsequential. We relax the requirement for $V_f$ to be an integer, and let it assume any value within a given range. Fig. 5 shows the schematic of a neuron in a CANN. It consists of a modified phase locked loop (PLL), comprising a loop filter, a voltage controlled oscillator (VCO), and several phase detectors (PDs). The frequency of the *ith* VCO, $f_h$ represents the output of the rth neuron (i.e. $V_t$; where $f_{min} \le f_t \le f_{max}$)'

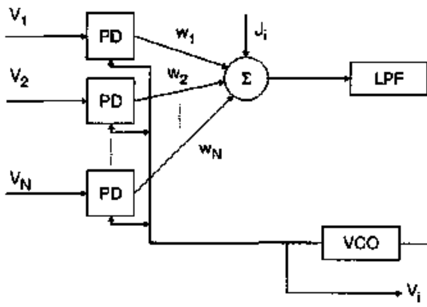


Fig. 5 *Model Oj(a neuron in a CANN*
*V],* K₂, ..., *Vy* denote inputs to neuron *i* from other neurons; *w*i ... tv^ are the corresponding weights. *Vj* is the output of neuron /

In general, the *jth* PD receives two inputs: the signal $f_t$ from neuron / itself, and signal *fj* from neuron/ The phase error is weighted by a weight *wy,* and the weighted errors from all the PDs are summed to form the net input to the loop filter. An external input /,· can also be input to the adder. The loop filter is usually chosen to be a first-order low pass filter, whose output is generally rectified so that the VCO does not receive negative inputs. However, this is not necessary if the VCO free running frequency is kept sufficiently high. The weights associated with the PDs depend on the specific application being considered. They may also be functions of the neuron outputs for certain combinatorial optimisation problems.

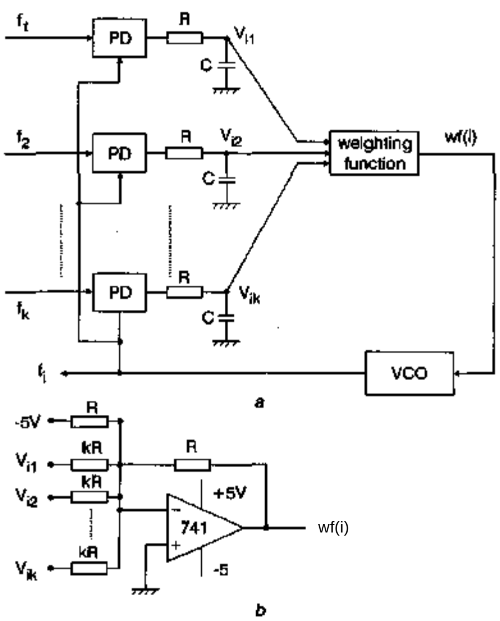


Fig. 6 *Circuit for neuron in a CANN*
*a* Circuit for the ith neuron in a CANN for the graph colouring problem. Node / is adjacent to *k* other nodes
*h* Detail of weighting function circuit

The CANN for graph colouring consists of *N* PLLs; if nodes / and *j* are adjacent, each of the corresponding PLLs has a phase detector with^ and/J as inputs; the associated weights are negative. External inputs /, are absent. If two nodes are not adjacent, the corresponding phase detectors are absent; we return to this aspect in the sequel. Fig. 6 shows a schematic illustrating the circuit for the rth node. The weighting function assigns a weight equal to (-1/&) to each connection. In a normal PLL, feedback is designed to make the PLL frequency lock to the input. In the CANN for graph colouring, the sign of the feedback is reversed, so that the frequencies diverge, or the difference is maximised. Observe the analogy with the first term of eqn. 13. If the *N* nodes form a clique, the frequencies should ideally be given

*ty fmin + Wnax -f<sub></sub>min)HN\,* "' = 1, 2, ..., *N.*

The circuit is very simple to construct and experiment with. The major advantage is that the hardware complexity is substantially reduced, since only *N* neurons with $O(N^2)$ interconnections are required, making it possible to realise larger systems in a given chip area or with a limited hardware resource.

Figs. 7 and 8 depict the results obtained in the laboratory for some simple three- and four-node examples. The VCO of each PLL was configured to operate between lOHz ($f_{min}$) and 1 kHz ($f_{max}$). In each case, the circuit was switched on from rest and allowed to settle into an equilibrium state. Optimal solutions were obtained for all three node examples; however, the CANN implementations converged to the optimal solution for only a few four-node examples. This issue is discussed in Section 4.

| Problem | Solution |
|---|---|
| 1 2 / 3 | country 1 = 1000 Hz country 2= 1000 Hz country 3= 1000 Hz |
| 1 2 / 3 | country 1 = 1000 Hz country 2= 10 Hz country 3= 1000 Hz |
| 1 2 / 3 | country 1 = 1000 Hz country 2= 10 Hz country 3= 495 Hz |
| 1 2 3 | country 1= 1000 Hz country 2= 10 Hz country 3= 1000 Hz |

Fig. 7 *Solutions obtained for three-node problems by using a CANN*
The circuits were made on a breadboard with discrete components and commercially available PLL ICs

| Problem | Solution |
|---|---|
| 1 2 / 3 4 | country 1 = 1000 Hz country 2 = 1000 Hz country 3 = 1000 Hz country 4 = 1000 Hz |
| 1 2 / 3 4 | country 1 = 1000 Hz country 2 = 10 Hz country 3 = 1000 Hz country 4 = 1000 Hz |
| 1 2 / 3 4 | country 1 = 1000 Hz country 2 = 600 Hz country 3 = 400 Hz country 4 = 10 Hz |
| 1 2 3 4 | country 1 = 1000 Hz country 2 = 10 Hz country 3 = 1000 Hz country 4 = 10 Hz |
| 3 / 1 2 / 4 | country 1 = 10 Hz country 2 = 600 Hz country 3 = 300 Hz country 4 = 1000 Hz |

Fiq.8 *Solutions obtained for a four-node (country) graph (map) colouring problem by using a CANN*
The circuits were realised on a breadboard

## 4 Modelling the hardware

While a linear PLL model is typically used in the literature for analysis, the VCO characteristic is a squashing function (i.e. it saturates for very low and very high inputs). This naturally limits the practical values of $V_b$ making constraints such as eqn. 12 unnecessary. Fig. 9 shows the experimentally obtained characteristic for the VCO on a PLL chip. The curve was approximated by the function:

$$f(x) = 25 + \frac{1350}{1 + \exp[1.3 \cdot (3.4 - x)]} - \frac{360}{1 + \exp[0.7 \cdot (3.55 - x)]} \qquad (17)$$

where $f(x)$ is the output frequency of the VCO for an input x. The tristate phase detector in the PLL is modelled as a finite state machine. Simulations were conducted in MATLAB and by programming in the 'C language. For 'C code, the differential equations were simulated by using a forward Euler approximation with a time step of 10 microseconds.

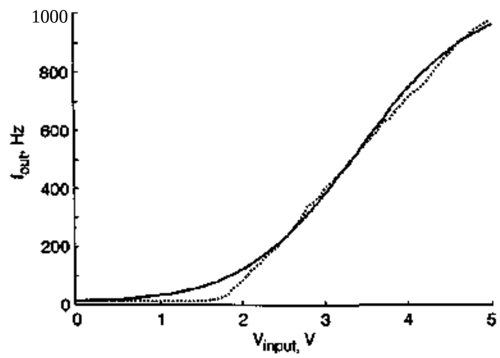


**Fig.** 9 *Experimental and modelled characteristics of the VCO on the PLL chip used in the experiments*
———— simulated
.......... measured

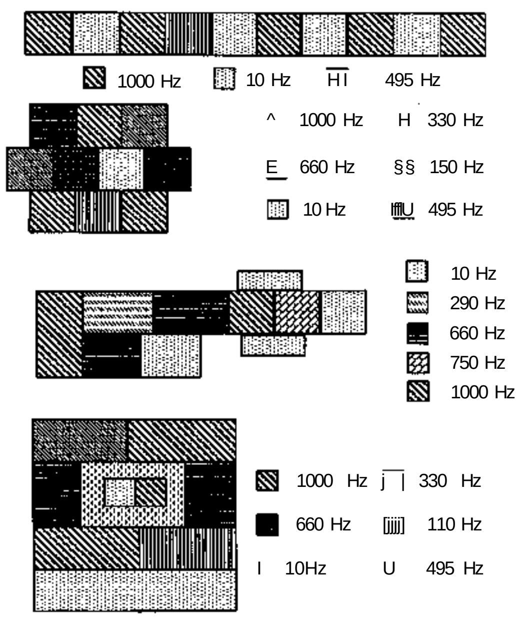


**Fig.** 10 *Simulation results using a CANN for some ten-node graph colouring problems*

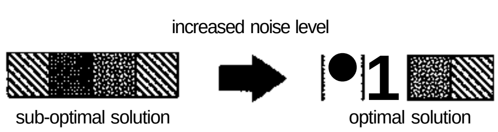


Fig. 11 *Increased noise levels help push the system out of local minima*

Fig. 10 shows solutions for some ten node examples. In the simulations, a small amount of uniformly distributed noise with amplitudes in the microvolt range was added. Noise is inherent in the actual hardware and plays an important role in convergence. In the simulated examples, suboptimal solutions were found in some cases. When the noise level is increased to a larger value, we observe that the system escapes from local minima and converges to the global optimum. Fig. 11 illustrates a simple example for which this occurred.

The reason for suboptimal solutions being found is that, while the objective function eqn. 13 maximises the colour difference between adjacent nodes (pairs for which $dy = 1$), or equivalently, the difference in frequencies between coupled neurons, there is no term which ensures that nodes which are not adjacent try to use the same colour. As discussed in Section 2 this may be achieved by replacing eqn. 13 with eqn. 16.

The modified energy function (eqn. 16) also implies that a positive coupling is introduced between PLLs corresponding to non-adjacent nodes (for whom $dy = 0$). Fig. 12 shows a set of examples where a small positive coupling was introduced. Note that the solutions obtained are optimal. It is known that PLLs which are mutually coupled with positive weights show chaotic behaviour around lock [23]. However, this aspect requires further investigation in the context of the CANN and its discussion has therefore been deferred.

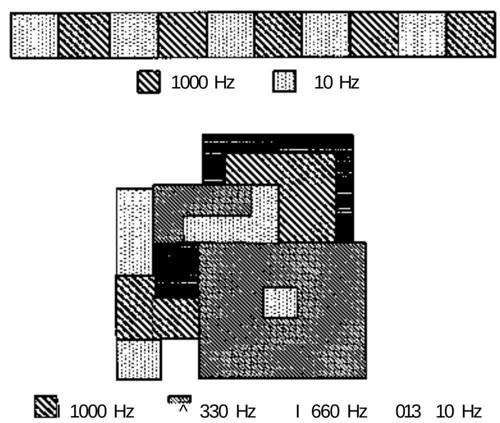


**Fig.** 12 *Simulation results with a CANN for some ten-node graph colouring problems*
A small positive coupling has been introduced between PLL corresponding to non-adjacent nodes

**Table 1: Convergence times for some four- and ten-node examples**

| Number of nodes | Example | Convergence time | |
|---|---|---|---|
| | | $RC=10^{-5}$ | $RC=10^{-3}$ |
| 4 | 1 | 13.56ms | 25.81 ms |
| | 2 | 14.6ms | 33.85ms |
| | 3 | 13.55ms | 28.87 ms |
| | 4 | 6.26 ms | 17ms |
| | 5 | 14.5ms | 38.1ms |
| 10 | 1 | 13.2 ms | 25.6ms |
| | 2 | 3.19ms | 17.8ms |
| | 3 | 12.9ms | 23.1ms |
| | 4 | 12.9ms | 30.8 ms |

Table 1 shows the convergence time for a set of four- and ten-node colouring problems. The table shows how convergence time varies with the time constant of the RC circuit (the low pass filter). Observe that the convergence time is almost independent of the problem size, but depends primarily on the complexity of a specific instance.

A change of two orders of magnitude in the RC time constant roughly doubles the convergence time (i.e. convergence time varies logarithmically with the RC time constant).

## 5 Conclusions

A novel approach for solving combinatorial optimisation tasks, termed as the compact analogue neural network or CANN, has been proposed in this paper and its application demonstrated for the graph colouring problem. An energy function has been derived for the application. A CANN with only $N$ neurons and $O(N^2)$ interconnections is required for solving the most complex graph colouring problem with $N$ nodes, while other reported approaches require $N^2$ neurons and O(A^) interconnections. A scheme for realising the CANN in hardware form has been discussed, which uses an array of modified PLLs whose frequencies represent the node colours. The CANN is highly amenable to VLSI implementation. A detailed analysis of the CANN, its convergence properties and related aspects will be presented elsewhere. It is worth mentioning that encoding in terms of duty cycle or phase can also be used with the same architecture.

## 6 References


1 HOPFIELD, J.J.: 'Neural networks and physical systems with emergent collective computational capabilities', *Proc. Nat. Acad. Sci. USA,* 1982, 79, pp. 2554-2558

2 HOPFIELD, J.J., and TANK, D.W.: 'Neural computation of decision optimization problems', *Biol. Cybern.,* 1985, 52, pp. 141-152

3 BILBRO, G., WHITE, M., and SNYDER, W.: 'Image segmentation with neurocomputers' *in* ECKMILLER, R., and MALSBURG, C. v.d (Eds.): 'Neural computers'. NATO ASI Series, 1988, F41, pp. 71-79

4 SANTINI, S., DEL BIMBO, A., and JAIN, R.: 'A Hopfield-like structure to extract road boundaries from a road image'. Proceedings of ICANN-91, 1991, (North-Holland), pp. 1681-1684

5 TIE-SEN, K., KANG-ZE, Y., and DIAN-NONG, L.: 'A method of spectral estimation using neural networks'. Proceedings of ICANN-91, 1991, (North-Holland), pp. 1633-1636

6 PROVENCE, J.: 'Neural network implementation for maximum-likelihood sequence estimation of binary signals in Gaussian noise'. Proceedings of the first international conference on *Neural networks,* 1987, Vol. Ill, pp. 703-714

7 HISANAGA, Y., and YAMASHITA, M.: 'Set partitioning of real numbers by Hopfield neural networks', *Syst. Comput. Jpn.,* 1991, 22, pp. 88-95

8 HOPFIELD, J.J., and TANK, D.W.: 'Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit', *IEEE Trans. Circuits Syst,* 1986, 33, pp. 533-541

9 KAMGAR-PARISI, B., and KAMGAR-PARISI, B.: 'An efficient model of neural networks for optimization'. Proceedings of the first international conference on *Neural networks,* 1987, Vol. Ill, pp. 785-790

10 SZU, H.: 'Fast TSP algorithm based on binary neuron output and analog neuron input using zero-diagonal interconnect matrix and necessary and sufficient constraints of the permutation matrix'. Proceedings of the first international conference on *Neural networks,* 1987, Vol. Ill, pp. 259-266

11 JAYADEVA: 'A compact neural network for graph colouring'. Proceedings of the 25th General Assembly of the International Union of Radio Science, 1996

12 APPEL, K., and HAKEN, W.: 'The solution of the four-colour-map problem', *Sci. Am.,* 1977, pp. 108-121

13 JAYADEVA, and BHAUMIK, B.: 'Optimization with neural networks: a recipe for improving convergence and solution quality', *Biol. Cybern.,* 1992, 67, pp. 445-449

14 JAYADEVA: 'Neural-based channel routing of cyclic and non-cyclic channels: lessons from nonlinear optimization', *Neural Network World,* 1997, 3, pp. 241-267

15 FIACCO, A.V., and McCORMICK, G.P.: 'Nonlinear programming techniques: sequential unconstrained minimization techniques' (John Wiley, 1968)

16 McCORMICK, G.P.: 'Nonlinear programming' (John Wiley, 1983)

17 DAHL, E.D.: 'Neural network algorithm for an NP-complete problem: map and graph colouring'. Proceedings of the first international conference on *Neural networks,* 1987, Vol. Ill, pp. 113-120

18 MOOPENN, A., THAKOOR, A.P., DUONG, T., and KHANNA, S.K.: 'A neurocomputer based on an analog-digital hybrid architecture'. Proceedings of the first international conference on *Neural networks,* 1987, Vol. Ill, pp. 479-486

19 THAKOOR, A.P., MOOPENN, A., LAMBE, J., and KHANNA, S.K.: 'Electronic hardware implementations of neural networks', *Appl. Opt.,* 1987, 26, pp. 5085-5092

20 TAKEFUJI, Y., and LEE, K.C.: 'Artificial neural networks for four-colouring map problems and K-colourability problems', *IEEE Trans. Circuits Syst,* 1991, 38, pp. 326-333

21 KIRKPATRICK, S., GELATT, CD., and VECCHI, M.P: 'Optimization by simulated annealing', *Science,* 1983, 220, pp. 671-680

22 JAYADEVA, : 'Sequential chaotic annealing and its application to multilayer channel routing'. Proceedings of VLSI 99: the 12th international conference on *VLSI design,* 1999, pp. 570-573

23 ENDO, T., and CHUA, L.O.: 'Chaos from two-coupled phase locked loops', *IEEE Trans. Circuits Syst,* 1990, 37, pp. 1183-1187