Mean Field Annealing for Pattern Classification using different response functions: A Comparative Approach.

Dr. Hussein Rady

The Shorouk Academy
Higher Institute for Computers and Information Systems
e.mail : dr_hussein_rady@yahoo.com

Abstract

Mean Field Annealing (MFA) merges collective computation and annealing properties of Hopfield Neural Networks (HNN) and Stochastic Simulated Annealing (SSA), respectively, to obtain a general algorithm for solving combinatorial optimization problems. Mean Field Annealing is a deterministic approximation, using mean field theory and stochastic simulated annealing. Since MFA is deterministic in nature, this gives the advantage of faster convergence to the equilibrium temperature, compared to stochastic simulated annealing. The mathematics of MFA is shown to provide a powerful and general tool for deriving optimization algorithms. In this paper, the MFA concepts are studied, the mathematics of MFA are derived, and different response functions are used to implement the MFA algorithm. Experimental results are implemented using different network topologies on a real classification problem known as Graph bipartitioning which was applied on Circuit Bi-partitioning. A comparative approach using the different response functions is applied. Two annealing schedules namely: the Cauchy annealing schedule and the linear annealing schedule are used and compared. The study and results are encouraging and promising.

Keywords: *Mean Field annealing, Hopfield Neural Network, Deterministic Annealing**,** Response Function, Ising Model, Annealing Schedule.*

1.Introduction.

Many optimal classification techniques are limited in their use since the solution is frequently trapped in a local minimum. In other words, the solution cannot free itself from the local minimum and move towards the global minimum. The deterministic gradient method and the stochastic annealing method are two major classes of optimal techniques for resolving combinatorial optimization problems. The Hopfield neural network is the most widely used deterministic method. The Hopfield network is a recurrent network that embodies a profound physical principle, namely, storing information in a dynamically stable configuration. However, energy change in the Hopfield network is a steepest descent process. Therefore, it always converges to the local minimum not the global minimum. Stochastic Simulated Annealing is a stochastic optimization algorithm based on the physical analogy of annealing a system of molecules to its ground state. It has a non-zero probability of transition from one state to another and the ability to move towards a worse state to escape from local traps. The cooling process is simulated using the Monte Carlo simulation method following the Boltzmann transition rule. The MFA algorithm gives a simple approximation to the state in thermal equilibrium. In the MFA, the state values are replaced by their means. In contrast, SSA performs computationally expensive Monte Carlo simulations and then extracts the average as the final result. Therefore, a major computational saving can be obtained through the MFA algorithm. Indeed, many simulation results confirm the MFA reaches equilibrium faster than SSA [22].

A Hopfield network is a fully connected recurrent single layer, unsupervised network. Hopfield and Tank were the first to use a neural network model for solving optimization problems [18,19]. Hopfield neural networks contain highly interconnected nonlinear processing elements ("neurons") with two-state threshold neurons or graded response neurons. Hopfield neural networks (HNN) are a class of densely connected single layer nonlinear networks of perceptrons. It contains highly interconnected nonlinear processing elements [21]. The network's energy function is defined through a learning procedure so that its minima coincide with states from a predefined set. However, because of the network's nonlinearity, a number of undesirable local energy minima emerge from the learning procedure. This has shown to significantly affect the network's performance [20].

Mean field annealing is a deterministic approximation to stochastic simulated annealing which is significantly more computationally efficient (faster) than stochastic simulated annealing. Instead of directly simulating the stochastic transitions in stochastic simulated annealing, the mean (or average) behavior of these transitions is used to characterize a given stochastic system. Because computations using the mean transitions attain equilibrium faster than those using the corresponding stochastic transitions, mean field annealing relaxes to a solution at each temperature much faster than does stochastic simulated annealing. This leads to a significant decrease in computational effort. The idea is to use a deterministic mean-valued approximation for a system of stochastic equations to simplify the analysis that has been adopted at various instances. Generally speaking, such approximations are adequate in high dimensional systems of many interacting units (states) where each state is a function of all or a large number of other states allowing the central limit theorem to be used.

Deterministic annealing is a stochastic simulated annealing based method, which replaces computationally intensive stochastic simulations by straightforward deterministic optimization of the modeled system error energy[6]. A deterministic annealing approach replaces explicit stochastic simulations by expectations [7]. Unlike stochastic simulated annealing, where random moves are made on the given energy surface, deterministic annealing can be viewed as incorporating the “randomness” into the energy function by extracting properties of the macroscopic system from microscopic averages. In this approach, an effective energy function is obtained and is deterministically optimized at each temperature sequentially, starting from a high temperature and going down [5]. The word deterministic refers to the fact that thermal equilibrium is obtained by directly minimizing the free energy, in opposition to the stochastic simulation used by stochastic simulated annealing [2]. Deterministic annealing for optimization of the organized modularity provides good solutions either on a classification point of view or on a visual point of view [3]. Its extensions also rely on clustering, regression and parsimonious modeling [6,8].

The rest of the paper is organized as follows: Section two discusses the Simulated Annealing (stochastic and deterministic). In section three, the physical analogy between Hopfield Networks and Statistical Mechanics is illustrated with its mathematical interpretations which shows the isomorphism between Hopfield Networks and the *Ising Model*. Section four discusses the Mean Field theory, the Mean Field approximation, and the derivation of the Mean Field equations. Experimental results are discussed in section five. Conclusion is outlined in section six.

2. Simulated Annealing.

Simulated Annealing (SA) is a compact and robust technique, which provides excellent solutions to single and multiple objective optimization problems with a substantial reduction in computation time. It is a method to obtain an optimal solution of a single objective optimization problem and to obtain a pareto set of solutions for a multi-objective optimization problem. It is based on an analogy of thermodynamics with the way metals cool and anneal. If a liquid metal is cooled slowly, its atoms form a pure crystal corresponding to the state of minimum energy for the metal. The metal reaches a state with higher energy if it is cooled quickly [13].

Simulated Annealing (SA) is one of the emergent calculation algorithms that solves optimization problems, and is an effective technique for solving combination optimization problems [30, 31, 32, 33]. SA is an algorithm that simulates the physical evolution of a solid from a high temperature state to a thermal equilibrium state. SA searches randomly around the neighborhood of a present searching point. The next searching point can be accepted even when the fitness value of the next point is worse than that of the present. SA algorithms repeat these steps, and the optimization state is finally expected from given initial state. Therefore, it can derive the global solution.

In physics, the method for allowing a system such as many magnets or atoms in an alloy to find a low-energy configuration is based on *annealing* [34]. In physical annealing the system is heated, thereby conferring randomness to each component (magnet). As a result, each variable can temporarily assume a value that is energetically *un*favorable and the full system explores configurations that have high energy. Annealing proceeds by gradually lowering the temperature of the system — ultimately toward zero and thus no randomness — so as to allow the system to relax into a low-energy configuration. Such annealing is effective because even at moderately high temperatures, the system slightly favors regions in the configuration space that are overall lower in energy, and hence are more likely to contain the global minimum. As temperature is lowered, the system has an increasing probability of finding the optimum configuration [35].

In metallurgy [29] and material science, annealing is a heat treatment of material with the goal of altering its properties such as hardness. Metal crystals have small defects, dislocations of ions which weaken the overall structure. By heating the metal, the energy of the ions and, thus, theirdiffusion rate is increased. Then, dislocations can be destroyed and the structure of the crystal is reformed as the material cools down and approaches its equilibrium state. When annealing metal, the initial temperature must not be too low and cooling must be done sufficiently slowly so as to avoid the system getting stuck in a meta-stable, non-crystalline, state representing a local minimum of energy.

Simulated annealing algorithm is a general purpose optimization technique that has been used to solve many combinatorial optimization problems [28]. It has been derived from the concept of metallurgy in which we have to crystallize the liquid to required temperature [29]. In this process liquids will be initially at high temperature and molecules are free to move. As the temperature goes down, there shall be restriction in the movement of the molecules and liquid begins to solidify. If the liquid is cooled slowly enough, then it forms a crystallized structure. This structure will be in minimum energy state. If the liquid is cooled down rapidly then it forms a solid which will not be in minimum energy state. Thus the main idea in simulated annealing is to cool the liquid in a control matter and then to rearrange the molecules if the desired output is not obtained. This rearrangement of molecules will take place based on the objective function which evaluates the energy of molecules in the corresponding iterative algorithm. SA aims to achieve global optimum by slowly converging to a final solution, making downward moves hoping to reach global optimum solution [27].

2.1 Stochastic Simulated Annealing.

The stochastic simulated annealing (SSA) combines the gradient descent technique which is a probabilistic hill-climbing algorithm with a random process to find the global minimum for its energy function E [36]. Stochastic simulated annealing models the degrees of freedom as a collection of atoms, slowly being cooled into their stable states with the temperature T as the controlling parameter. The energy surface defined as E(s) for a particle state s is a Boltzmann distribution function that allows changes in s to increase E, thus providing the network with a mechanism to escape from being trapped in a local minimum [37]. This is made possible since changes to s which decrease E are always accepted, whereas a move which causes an increasing ΔE will be taken with the Boltzmann probability, Pr{uphill move} = $\exp(-\Delta E / T)$.

The stochastic simulated annealing method for finding an optimal configuration of neuron states given a set of weights is based on the physical annealing metaphor. It involves the following basic steps:

1. Randomize neuron states once in the beginning, and initialize the temperature to a high value.
2. Choose a neuron I randomly from the network.
3. Compute the energy E_A of the present configuration A.
4. Flip the state of neuron I to generate a new configuration B.
5. Compute the energy E_B of configuration B.
6. If $E_B < E_A$ then accept the state change for I. otherwise accept the state change for neuron I with a probability $e^{-\Delta E/T}$ where $e^{-\Delta E/T} = E_B - E_A$.
7. Continue selecting and testing neurons randomly, and set their states several times in this way until a thermal equilibrium is reached.
8. Finally, lower the temperature and repeat the procedure.

Decreasing the temperature continues until it reaches a very small value.

2.2 Mean Field Annealing.

Mean Field Annealing combines the collective computation property of Hopfield neural network model with the annealing notion of stochastic simulated annealing in order to form a better algorithm. In MFA, discrete variables called *spins* (neurons or nodes) are used for encoding the combinatorial optimization problems. In MFA each node or neuron responds to the average or mean of forces (fields) due to the nodes connected to it [11]. An *energy* function written in terms of spins is used for representing the cost function of the problem. Then, using the expected values of these discrete variables, a gradient descent type relaxation scheme is used to find a configuration of the spins which minimizes the associated energy function. MFA is also a general strategy like stochastic simulated annealing, and can be applied to different problems with suitable formulations.

Mean field annealing (MFA) merges collective computation and annealing properties of Hopfield neural networks[18], and stochastic simulated annealing, respectively, to obtain a general algorithm for solving combinatorial optimization problems. MFA can be used for solving a combinatorial optimization problem by choosing a representation scheme in which the final states of the spins (neurons) can be decoded as a solution to the target problem. Then, an energy function is constructed whose global minimum value corresponds to an optimum solution of the problem to be solved. MFA is expected to compute the optimum solution to the target problem, starting from a randomly chosen initial state, by minimizing this energy function. Steps of applying mean field annealing technique to a problem can be summarized as follows:

1. Choose a representation scheme which encodes the configuration space of the target optimization problem using spins. In order to get a good performance, number of possible configurations in the problem domain and the spin domain must be equal, i.e., there must be a one-to-one mapping between the configurations of spins and the problem.
2. Formulate the cost function of the problem in terms of spins, i.e., derive the energy function of the system. Global minimum of the energy function should correspond to the global minimum of the cost function.
3. Derive the mean field theory equations using the energy function i.e., derive equations for updating averages (expected values) of spins.
4. Minimize the complexity of update operations in order to get an efficient algorithm.
5. Select the energy function and the cooling schedule parameters.

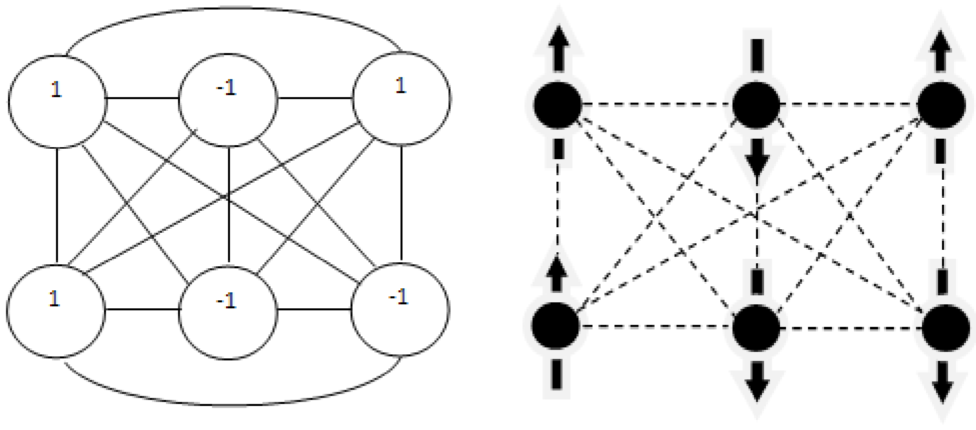
Mean field annealing is identical to stochastic simulated annealing except for one aspect: instead of choosing moves randomly, it intelligently selects what appears to be the best candidate. Mean field annealing is a specific type of a graduated nonconvexity algorithm[9]. The advantages of mean field annealing is that, it is one of two orders of magnitude faster than stochastic simulated annealing

3. The physical analogy between Hopfield Networks and Statistical Mechanics

The mean field annealing algorithm is a formulation of combinatorial optimization in terms of artificial neural networks (ANN). Artificial neural networks are mostly used in applications such as pattern recognition and classification but may also be used for optimization.

Historically, the connection between artificial neural networks and statistical mechanics has its roots in a 1982 paper by Hopfield. With the introduction of an energy function, the isomorphism between recurrent neural networks and the *Ising model* as shown in figure(1), a fundamental in statistical mechanics, a whole new set of analysis tool became available. As a consequence, a large number of physicists are involved in the development of artificial neural networks.

Physicists worked on *Ising models* for many years and obtained brilliant achievements in the theory of continuous phase transition, which occur when a small change in a parameter such as temperature or pressure causes a large-scale, qualitative change in the state of a system. Phase transitions are common in physics and familiar in everyday life [17]. The Ising model is defined on a discrete collection of variables called spins, which can take the value of 1 or -1. The spins interact, in pairs, with energy that has one value when the two spins are the same, and a second value when the two spins are different. In Ising model, the spin at every site is either up (+1) or down (-1)[16].



Figure(1) : The isomorphism between neural networks and Ising spin-systems. (Left) A fully connected neural network with nodes taking values {-1, 1}. The solid lines represent the coupling between the neurons. (Right) A spin-system with particles (solid dots) having either spin up or down (arrows) and the force interaction between the spins (dashed lines). Both systems are described by the energy function(1) and the update rule(2).

The conventional Hopfield neural network model is the most commonly used model for auto-association and optimization. Hopfield networks are auto-associators in which node values are iteratively updated on local computation principle: the new state of each node depends only on its net weighted input at a given time. This network is fully connected network and the weight matrix determination is one of the important tasks when used for certain applications[19]. In Hopfield neural network model, processing devices are called neurons. Each neuron has two states firing 1 and not firing 0. Each neuron is connected to other neuron by a weight factor W_{ij}, such that $W_{ij} = W_{ji}$ [10,15].

The Hopfield energy function

$$f = -\frac{1}{2}\sum_{i \neq j=1}^{N} w_{ij} s_i s_j \qquad \rightarrow (1)$$

and the rule for updating the state variables

$$s_i(n+1) = sgn\left(\sum_{j \neq i} w_{ij} s_j(n)\right) \qquad \rightarrow (2)$$

provides a simple way of exemplifying the connection between neural networks and the spin-systems of statistical mechanics. In the neural network interpretation the variables s_i represent the state of an individual node and w_{ij} the connection strength between node *i* and node *j*. The updating rule prescribes that a node flips its state according to an *activation potential*,

$$u_i = \sum_{j \neq i} w_{ij} s_j \qquad \rightarrow (3)$$

which is a weighted sum of all the states of the other nodes. In the statistical mechanics interpretation, the variable s_i represent the *spin* of a particle and w_{ij} the force interaction between the spin of particle *i* and *j*.

Hopfield defined a single-layer network consisting of interconnected individual perceptrons and modified perceptrons (with sigmoid nonlinearities)[20]. The Hopfield net has a large number of stable states, corresponding to local minima of the energy function (1). Given a starting state, the update rule (2) will drive the network to the nearest local minimum. By associating the states with a *pattern* the network can be used as an associative pattern memory, where a distorted or partial pattern is used as a starting state from which the network can relax to a previously stored valid pattern. The patterns are stored in the network by clever coding of the connection weight w_{ij} such that a pattern is stored in w_{ij} through a *learning rule.*

Stochastic nodes

To reach the global minimum of the energy function (1) from any given starting state, the updating rule (2) is not sufficient, and techniques such as simulated annealing, allowing the process to escape local minima, must be employed.

In a physical spin-system in an environment with a non-zero temperature the spins will fluctuate due to *thermal noise.* To emulate the thermal noise behavior, the deterministic updating rule (2) is replaced by a stochastic rule

$$s_i = \begin{cases} +1 \; with\; prbability\; g(u_i) \\ -1 \; with\; probability\; 1 - g(u_i) \end{cases} \qquad \rightarrow (4)$$

where

$$g(u_i) = \frac{1}{1 + e^{-2w_i/c}} \qquad \rightarrow (5)$$

Since 1 – g(u_i) = g($-u_i$) we can write the probability of finding s_i in either of its states at any given value of the control parameter *c* as

$$P(s_i = \pm 1) = \frac{1}{1 + e^{+2w_i/c}} \qquad \rightarrow (6)$$

This makes it reasonable to talk about the *thermal average* of a variable, interpreted as the expectation value

$$E[s_i] = (+1)P(s_i = 1) + P(s_i = -1)$$

$$= \frac{1}{1 + e^{-\frac{2u_i}{c}}} - \frac{1}{1 + e^{\frac{2u_i}{c}}}$$

$$= \frac{e^{u_i/c} - e^{u_i/c}}{e^{u_i/c} + e^{u_i/c}} \equiv \tanh(u_i/c) \qquad \rightarrow (7)$$

It is easy to see that in the limit of $c \rightarrow 0$, which means that there is no thermal noise present, the stochastic updating rule (4) turns into the original deterministic updating rule(2).

The process of searching for the global minimum open up the possibility of using Hopfield networks to solve optimization problems coded inw_{ij}. The initial approach was only partly successful and frequently led to infeasible or low quality solutions but the idea was important in spurring development of similar more efficient optimization methods.

As a remedy for the slow convergence, the Mean Field learning rule, which replaces the random sampling of states with an approximate calculation, was introduced. The basic idea, which also explains the name of the algorithm, is that instead of explicit (a very large number of) states, the expectation value of the node under study is determined by the *mean field* generated by the surrounding nodes.

4. Mean field theory

The main idea of the Mean Field theory is to focus on one particle and assume that the most important contribution to the interactions of such particle with its neighboring particles is determined by the mean field due to the neighboring particles. For a physical many-particle system in thermal equilibrium, the state probability distribution is given by the Boltzmann distribution:

$$P(s) = \frac{e^{-f(s)/c}}{Z} \qquad \rightarrow (8)$$

Where Z denotes the partition function

$$Z = \sum_S e^{-f(s)/c} \qquad \rightarrow (9)$$

and S is the set of all possible states s.

The way to improve the convergence speed of the Boltzmann machine is to work with the mean values, or rather expectation values, of the state variables. The reasoning behind this is quite simple. We already know that the Boltzmann distribution (8) holds all information on a system in thermal equilibrium and from it we can calculate the thermal average of any quantity *A* that depends on **S.** In particular, the expectation value of a state, $E[\mathbf{s}] = \{E[s_i]\}$, for a particular value of *c* is given by

$$E[s_i] = \frac{1}{Z}\sum_{s\in S} s_i e^{-f(s)/c} \qquad \rightarrow (10)$$

where Z is the partition function (9).

It is in fact possible to obtain $E[\mathbf{s}]$, and many other quantities, directly from Z without the need for the extra summation in (10). The procedure is best shown by an example: start from a Hopfield energy function

$$f = -\frac{1}{2}\sum_{ij} w_{ij}s_i s_j - \sum_i b_i s_i \qquad \rightarrow (11)$$

where the bias terms b_i are introduced for the sake of this example, they can be set to zero later.

Differentiating the partition function Z for the energy given in (11) with respect to b_i

$$\frac{\partial Z}{\partial b_i} = \sum_{s\in S} s_i e^{-f/c} \qquad \rightarrow (12)$$

it can immediately be seen that the result is indeed $E[s_i]$ save for a factor c/Z, which means that we can establish the following relation

$$E[s_i] = \frac{c}{Z}\frac{\partial Z}{\partial b_i} = c\frac{\partial}{\partial b_i} logZ. \qquad \rightarrow (13)$$

Furthermore, under the condition that there is a single optimal state $\boldsymbol{s}_{opt}$

$$\lim_{c\to 0} E[\boldsymbol{s}] = \lim_{c\to 0}\frac{1}{Z}\sum \quad \boldsymbol{s}e^{-f(s)/c} = \boldsymbol{s}_{opt} \qquad \rightarrow (14)$$

This suggests that instead of minimizing f(**s**) directly, we could try to evaluate the mean value *E*[**s**] at a sufficiently high temperature, and then track it as the control parameter *c* is lowered towards zero, and the partition function contains enough information to yield *E*[**s**].

4.1The mean field approximation

In statistical physics the mean-field approximation is one of the most common and easy-to-use frameworks [25]. It is also a powerful method for finding minimum points of cost or energy functions. The method has similarities to Boltzmann machines, as both methods are based on stochastic simulated annealing in order to avoid local minimum. Mean-field approximation is a deterministic method that uses the results of spin-glass theory[24]. The term ‘mean-field approximation’ can have various meanings in statistical mechanics and it actually refers to a whole set of different approximations all with different levels of accuracy. The main idea is to ignore the fact that the constituents of the solid under investigation are interacting, and to treat the interaction as an ‘average effect’. The difference between approximations lies in the way we perform averaging [26].

The following method to estimate *E*[**s**] by means of the mean field (MF) equations is a standard method of statistical mechanics. I will however skip on some of the details in the derivation of the MF-equations in order to keep the complexity down while still providing reasonable arguments backing the results.

Before deriving the MF-equations, we need some prerequisites. Consider the set S = {s | s = {s_i}, s_i in *P*, i = 1,2,…,L} where *P* is a problem specific set of feasible values. Furthermore, a mapping $f: S \rightarrow R$ is assumed to exist for every problem, associating each state with a real valued cost, with the lowest cost corresponding to the optimal solution. Given a function $g(s_j)$ used in a summation over the values of s_j we want to express it as an integral in a pair of continuous variables (u_j, v_j). Generally, we can write

$$g(s_j) = \int_{\mathbb{R}} g(v_j)\delta(s_j - v_j)dv_j \qquad \rightarrow (15)$$

where $\delta(v_j)$ is the Dirac delta function, defined as

$$\delta(v_j) = \frac{1}{i2\pi}\int_{\mathbb{C}} e^{u_j v_j} du_j \qquad \rightarrow (16)$$

which in combination with (15) gives

$$g(s_j) = \frac{1}{i2\pi}\int_{\mathbb{R}}\int_{\mathbb{C}} g(v_j)e^{u_j(s_j - v_j)}du_j dv_j. \qquad \rightarrow (17)$$

Using (17) we can now state an expression for

$$\Sigma_{s_j=\pm1} g(s_j) = \frac{1}{i2\pi}\Sigma_{s_j=\pm1}\int_{\mathbb{R}}\int_{\mathbb{C}} g(v_j)e^{u_j(s_j - v_j)}du_j dv_j \qquad \rightarrow (18)$$

$$= \frac{1}{i2\pi} \int_{\mathbb{R}} \int_{\mathbb{C}} g(v_j) e^{-u_j v_j} \sum_{s_j = \pm 1} e^{u_j s_j} du_j dv_j \qquad \rightarrow (19)$$

$$= \frac{1}{i\pi} \int_{\mathbb{R}} \int_{\mathbb{C}} g(v_j) e^{-u_j v_j} e^{\log \cosh u_j} du_j dv_j \qquad \rightarrow (20)$$

where the last step is accomplished by noting the identity

$$e^{u_j} + e^{-u_j} \equiv 2 \cosh u_j = 2 e^{\log \cosh u_j}. \qquad \rightarrow (21)$$

To perform the summation over all states **s** we note that

$$\sum_{s \in S} g(\mathbf{s}) = \sum_{s_1 = \pm 1} \sum_{s_2 = \pm 1} \cdots \sum_{s_L = \pm 1} g(\mathbf{s}) \qquad \rightarrow (22)$$

and consequently

$$\sum_{s \in S} g(\mathbf{s}) \alpha \prod_{j=1}^{L} \int_{\mathbb{R}} \int_{\mathbb{C}} g(\mathbf{v}) e^{-\sum_{j=1}^{L} (u_j v_j - \log \cosh u_j)} du_j dv_j \qquad \rightarrow (23)$$

where a constant factor has been left out.
Given the expression (23) to estimate the sum over all states of a general expression in **s** we can now continue with the derivation of the MF-equations. Using (23) we can rewrite the partition function (9) in terms of the new, continuous, variables **u** and **v** as

$$Z \ \alpha \prod_{j=1}^{L} \int_{\mathbb{R}} \int_{\mathbb{C}} e^{-\frac{1}{c} f_e(\mathbf{v}, \mathbf{u}, c)} du_j dv_i \qquad \rightarrow (24)$$

where the *effective energy* (cost) f_e is:

$$f_e(\mathbf{v}, \mathbf{u}, c) = f(\mathbf{v}) + c \sum_{j=1}^{L} (u_j v_j - \log \cosh u_j) \qquad \rightarrow (25)$$

From statistical mechanics it is well known that the double integral of (24) is dominated by its *saddle points* and *Z* can thus be approximated by the integrand itself,

$$Z \propto e^{-\frac{1}{c} f_e(\mathbf{v}, \mathbf{u}, c)} \qquad \rightarrow (26)$$

Evaluated for the saddle points given by the simulations stationarity in

$$\frac{\partial f_e}{\partial v_i} (\mathbf{u}, \mathbf{v}, c) = 0 \qquad \rightarrow (27)$$

and

$$\frac{\partial f_e}{\partial u_i} (\mathbf{u}, \mathbf{v}, c) = 0. \qquad \rightarrow (28)$$

The saddle point requirements lead to

$$u_i = -\frac{1}{c} \frac{\partial f(\mathbf{v})}{\partial v_i} \qquad \rightarrow (29)$$

and

$$v_i = \tanh u_i \qquad \rightarrow (30)$$

which are subsequently combined to form the MF-equations

$$v_i = \tanh \left(-\frac{1}{c} \frac{\partial f(\mathbf{v})}{\partial v_i} \right) \qquad \rightarrow (31)$$

Applying the MF-equations to the now familiar Hopfield energy function (3) we find that (31) states the relation

$$v_i = \tanh(\frac{1}{c}\sum_{j\neq i} w_{ij}v_j) \qquad \rightarrow (32)$$

which, when compared to (9) from the discussion on stochastic nodes leads to the interpretation of v_i as the thermal average $E[s_i]$ of s_i for c > 0. Again, in the limit of c = 0 the expression (3) is recovered.
The only, but important, difference between (9) and (32) is that the activation potential u_i is computed from the thermal average $E[s_i]$ of s_i rather than s_i itself. By explicitly stating u_i and moving the summation inside the expectation operator

$$u_i = \frac{1}{c}\sum_{j\neq 1} w_{ij}E[s_i] = \frac{1}{c}E\left[\sum_{j\neq i} w_{ij}s_i\right] \qquad \rightarrow (33)$$

We can interpret the activation potential as the mean of the contributions from the field of surrounding states (spins).
As an alternative to interpretation by visual inspection, returning to the Hopfield energy in equation (11) expressed in terms of **v**

$$f(\mathbf{v}) = -\frac{1}{2}\sum_{ij} w_{ij}v_iv_j - \sum_i b_iv_i \qquad \rightarrow (34)$$

and the corresponding u_i

$$u_i = -\frac{1}{c}\frac{\partial f(\mathbf{v})}{\partial v_i} = \frac{1}{c}\left(\sum_j w_{ij}v_j + b_i\right). \qquad \rightarrow (35)$$

From the approximation expression (26) to compute *Z* we find

$$\log Z = C - \frac{1}{C} f_e(\mathbf{v},\mathbf{u},c)$$
$$= C - \frac{1}{2c}\sum_{ij} w_{ij}v_iv_j + \sum_i \log\cosh\left(\frac{1}{c}\left(\sum_j w_{ij}v_j + b_i\right)\right) \qquad \rightarrow (36)$$

where *C* is a constant and by using the relation between $E[s_i]$ and *Z* established in (13) we find

$$E[s_i] = c\frac{\partial}{\partial b_i}\log Z = \tanh\left(\frac{1}{c}\left(\sum_j w_{ij}v_j + b_i\right)\right) = \tanh u_i = v_i \qquad \rightarrow (37)$$

As expected.
In a similar way, if we replace equation (4) by the following equation:

$$s_i = \begin{cases} 1 \;\; with\ prbability\ g(u_i) \\ 0 \;\; with\ probability\ 1 - g(u_i) \end{cases} \qquad \rightarrow (38)$$

We reach to the following equation:

$$E[s_i] = \frac{1}{1+e^{-u_i}} \text{ , which is a sigmoid function.} \qquad \rightarrow (39)$$

5. Results.

Suppose we have a large number of variables (neurons or nodes) s_i , i = 1,2,…,N, where each variable can take one of two discrete values, for simplicity chosen to be ± 1 . The optimization problem is: find the values of s_i so as to minimize the cost or energy

$$E = \frac{-1}{2} \sum_{i,j=1}^{N} w_{ij} \, si \, sj \qquad \rightarrow \quad (40)$$

where w_{ij} is the weights (edges) that are symmetric, and can be positive or negative. We require the self-feedback terms to vanish (i.e., $w_{ii} = 0$) , because the non-zero w_{ii} merely add an unimportant constant to E, independent of s_i . This optimization problem can be visualized in terms of a network of nodes, where bidirectional links or interconnections correspond to the weights $w_{ij} = w_{ji}$. This network suggests a physical analogy which in turn will guide our choice of solution method. Imagine that the network represents N physical magnets, each of which can have its north pole pointing up ($s_i = +1$) or pointing down ($s_i = -1$). The w_{ij} are functions of the physical separations between the magnets. Each pair of magnets has an associated interaction energy which depends upon their state, separation, and other physical properties: $E_{ij} = -1/2 \, w_{ij} \, s_i \, s_j$. The energy of the full system is the sum of all these interaction energies as given in Equation (40).

The optimization task is to find the configuration of states of the magnets (nodes or units) with the most stable configuration; the one with lowest energy. This general optimization problem appears in a wide range of applications, in many of which the weights do not have a physical interpretation. In this paper, the implementation of our work was applied on a real classification problem known as **graph bipartitioning** (**bipartite graph**) that is defined as follows:
Definition: A *bipartite graph* is a graph G whose vertex V can be partitioned into two non empty sets V_1 and V_2 . The sets V_1 and V_2 are often called the color classes of G.

The **graph bipartitioning** problem can be applied, for example, on **Circuit Bi-partioning** i.e., when one has to split a circuit design over two circuit boards, and the objective is to place the components on either boards (partitions) in such a way that the number of connections between the circuit boards (cut size) is minimized, with the added constraint that the number of components on both boards is the same (balance constraint). This will arise in a VLSI circuit partitioning problem. Partitioning a VLSI circuit graph into disjoint graphs of minimum cut size is the objective.

The partitioning task is ubiquitous to many subfields of VLSI CAD. Partitioning heuristics are used to address the increasing complexity of VLSI design; systems with several million transistors are now common, presenting instance complexities that are unmanageable for existing logic level and physical level design tools. Partitioning divides a system into smaller, more manageable components; the number of signals which pass between the components corresponds to the interactions between the design sub-problems. In a top-down hierarchical design methodologies, early-made decisions in the system synthesis process will constrain succeeding decisions. Thus, the feasibility, not to mention the quality, of automatic placement, global routing, and detailed routing will somewhat depend on the quality of partitioning problem.

The circuit bi-partitioning optimization is focused on finding an acceptable solution based on the delay, power, and cut-set cost. The cut-set cost is the number of inter-partition connects, which if not selected carefully, will immensely degrade the overall solution quality.

Cut-set cost function. The cut-set cost is incremented each time a net has at least a connection in another partition. The pertinent net is checked for each node in the circuit; once found, the net is checked if it has connections in other partitions. A net that has connections in more than one partition means that these connections will have to be cut, i.e., increases the undesirable cut-set.

Imbalance Constraint. The partitions imbalance is the difference between the numbers of nodes of the partitions. The imbalance constraint is verified to be within a predetermined value, namely the imbalance tolerance.

If we identify the circuit with a graph where the components are the nodes and the connections between components are the edges of the graph, the state variables s_i have the following interpretation:

$$s_i = \begin{cases} +1\,, the\ i^{th} component\ belongs\ to\ the\ first\ board \\ -1\,, the\ i^{th} component\ belongs\ to\ the\ second\ board \end{cases}$$

We can thus formulate the following objective function:

$$f(s) = -\frac{1}{2}\sum_{i,j} w_{ij}\, s_i s_j + \beta\frac{1}{2}\left(\sum_i s_i\right)^2 \qquad \rightarrow (41)$$

Where the first term is minimal if the nodes of all connected pairs (s_i, s_j) are in the same partition and the second term is minimal if there is an equal number of nodes in each partition.

The parameter β is used to trade off the importance of a minimal cut size against the constraint that the number of nodes in each partition should be the same. A value of $\beta = 0$ would immediately lead to a solution where all nodes are gathered in the same partition. If the chosen value of β is too large the cut size would be far from optimal. There will in general always be a small imbalance in the final solution, but in most problems of this kind, a small imbalance is not a problem. If the β value equals 1, this will give equal weight to an additional cut and an added imbalance.

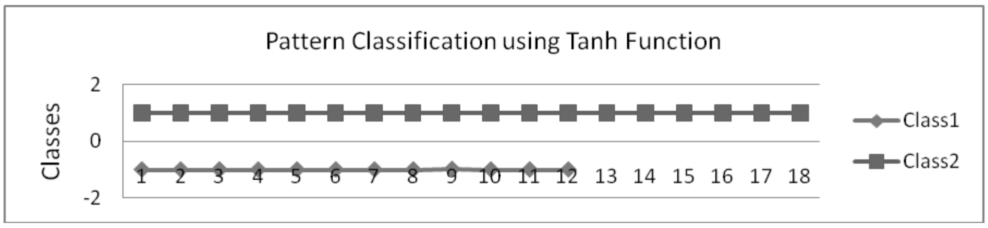
In MFA, each node (magnet) can take a continuous value $-1 \leq s_i \leq +1$, which equals the expected value of a binary node in the system at temperature T. In other words, the analog value s_i replaces the expectation of the discrete variable, $E[s_i]$. The behavior of the force exerted by the nodes connected to s_i behaves as follows: the larger this force, the closer the analog s_i getsto +1; the more negative this force, the closer s_i gets to -1. The temperature T also affects s_i. If T is large, there is a great deal of randomness and even a large force will not insure $s_i \approx +1$. At low temperature, there is little or no randomness and even a small positive force insures that $s_i = +1$. Thus at the end of an anneal, each node has value $s_i = +1$ or $s_i = -1$ as shown in our experimental results.

One of the most important features in MFA is the choice of the annealing schedule. The annealing is an operation in metal processing. Metal is heated up very strongly and then cooled slowly to get a very pure crystal structure with a minimum of energy. In this paper, we discuss two important annealing schedules as follows:

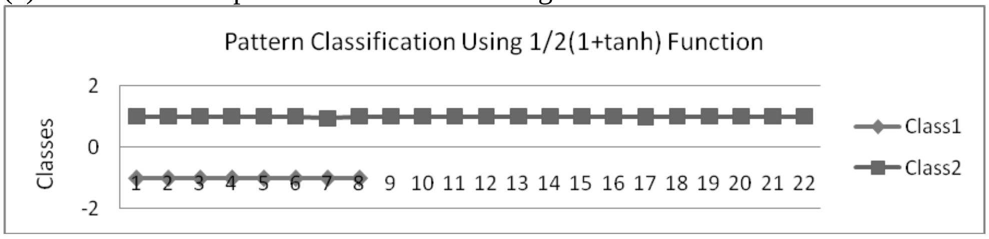
1- For the first schedule $T(k) = \alpha\, T(k-1)$, $0 < \alpha < 1$ → (S1)

This is a linear annealing function which is the simplest form of the polynomial annealing functions interpolated between the points determined by the start temperature, which is the beginning of the annealing process, the ending temperature, which denotes the ending process, the maximum temperature, and the minimum temperature. Experiences has shown that α should be between 0.8 and 0.99. Of course the higher the value of α, the longer it will take to decrement the temperature to the stopping criterion.

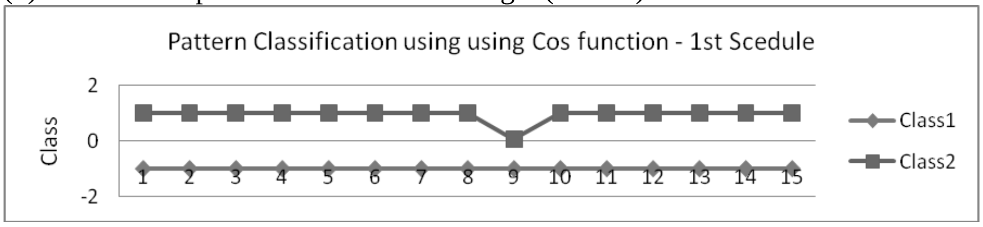
Figure(2) represents the classes obtained after training the network on the same adjacency matrix used for all response functions. The patterns are classified into two different classes. The number of patterns in each class differs from response function to another. Although all states lies in the interval [-1,1], i.e., $-1 \leq s_i \leq 1$, at the end of the annealing process we find that $s_i \rightarrow -1 \ or\ s_i \rightarrow 1$. This means that if the force exerted by the nodes connected to s_i is large then the analog s_i tends to +1. If the force exerted by the nodes connected to s_i is small (more negative) then the analog s_i tends to -1. At the beginning of the annealing, the temperature T is large, and there is a great deal of randomness. At the end of the annealing, the temperature T is low, and in this case there is little or no randomness.



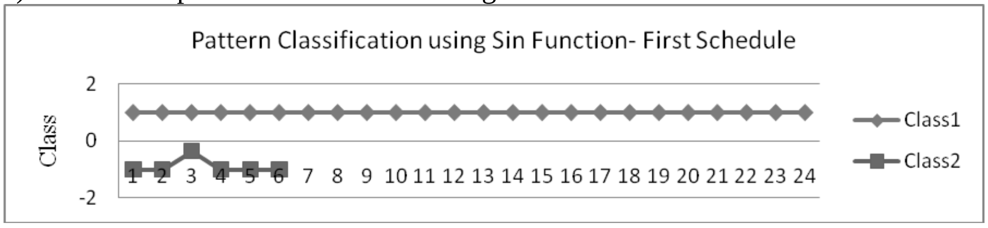
(a): The number of patterns in each class using Tanh Function.



(b) : The No. of patterns in each class using ½(1+tanh) Function.



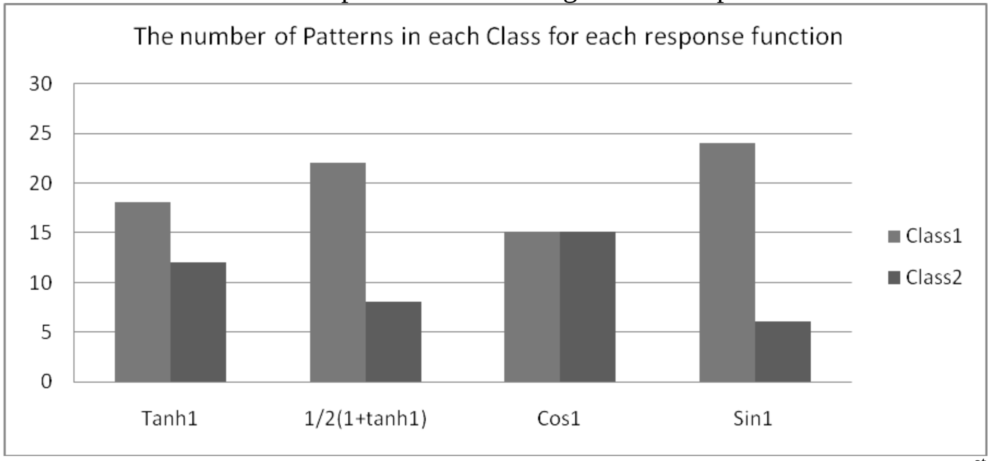
c): The No. of patterns in each class using Cos Function.



(d): The No. of patterns in each class using Sin Function.

Figure(2): Pattern Classification for using different response functions – the 1st Schedule.

In Figure(3), the number of patterns in each class using cosine response function are equal. This means that there is partitioning balance for each board when we apply the Circuit bi-partitioning problem. Each board contains an equal number of wires. The difference using Tanh function is less than the difference using ½(1+tanh) function which is less than the difference using Sin response function. In case of the Circuit bipartitioning problem, we find that the difference between the number of wires on the two boards, i.e., the partitioning imbalance using the Tanh response function is less than the partitioning imbalance using ½(1+tanh) which is less than its counterpart in case of using the sine response function.



Figure(3): the number of patterns in each class for the response functions- 1st Schedule.

Table(1) : the end value of each state at different functions using the 1st schedule.

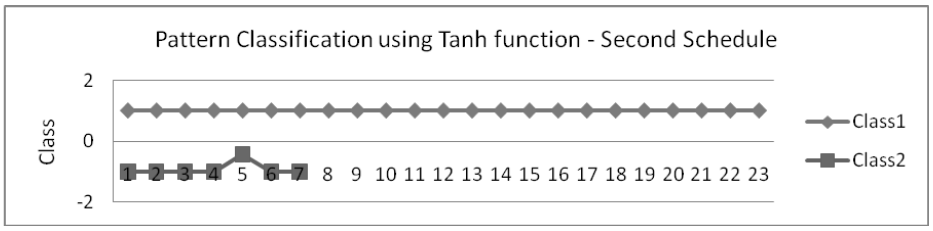
State	Tanh	1/2(1+Tanh)	Cos	Sin
S1	-1	-1	-1	-0.999185725679839
S2	1	1	-1	0.999984804160683
S3	-1	-1	-1	1
S4	1	1	-0.999999999999	1
S5	1	1	1	1
S6	1	1	0.9322041269898	1
S7	-0.9999999999	1	1	1
S8	-1	1	1	1
S9	1	0.93757078694	1	-0.999977806845763
S10	1	-1	-1	1
S11	1	1	1	1
S12	1	0.99999999363	-1	1
S13	1	-1	1	1
S14	-1	0.99999999999	1	1
S15	-1	1	-1	-0.366252200291844
S16	-0.9999999999	1	-1	1
S17	1	1	-1	1
S18	1	1	1	1

S19	-0.9828661461	1	0.0507615962715	1
S20	1	1	-1	1
S21	-0.9999999999	1	1	1
S22	1	-1	-1	1
S23	-1	-1	1	1
S24	1	0.98218915225	-1	1
S25	1	0.99999999995	-0.999999999999	-0.999657865027457
S26	1	-1	-1	-0.999788473314367
S27	-1	-1	1	1
S28	1	1	1	1
S29	1	1	-1	1
S30	1	1	1	-0.999998472487542

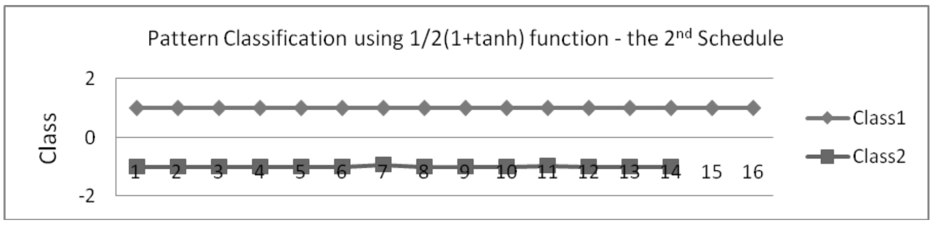
Table(1) indicates that for the first schedule, and at the beginning of the annealing, we find that the states of each node lies between -1 and +1 i.e., $-1 \leq s_i \leq +1$. Increasing the number of iterations implies that the state of each node tends to +1 or -1. A faster convergence to the equilibrium temperatures (i.e., at low temperature) occurs when the state reaches ± 1. In this case a minimum energy has been obtained. At the end of the annealing, we find that the state of each node converges either to +1 or -1. Applying all response functions, we find that there are two classes of patterns. The number of patterns in each class differs from the number of patterns in its counterpart using other response function.

2- For the second schedule T(k) = T(0) / k. → (S2)

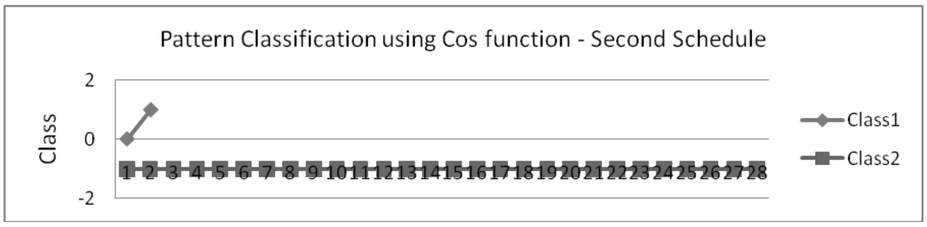
This schedule is known as the Cauchy annealing. A faster schedule is the Cauchy schedule in which the equation converges to the global minimum when moves are drawn from the Cauchy distribution. MFA allows each node to take on analog values during search; at the end of the search the values are forced to be $s_i = \pm 1$, as required by the optimization problem. Figure(5) shows the first eleven iterations against the different four response functions stated above. The eleventh iteration is when the first node converged to +1 using the tanh and $\frac{1}{2}(1 + tanh)$ response functions . The other two functions sine and cosine lies between ± 1. The convergence using this schedule as shown in figure(5) is faster than the convergence using the previous schedule which occurs at iteration number 18th as shown in figure(2).



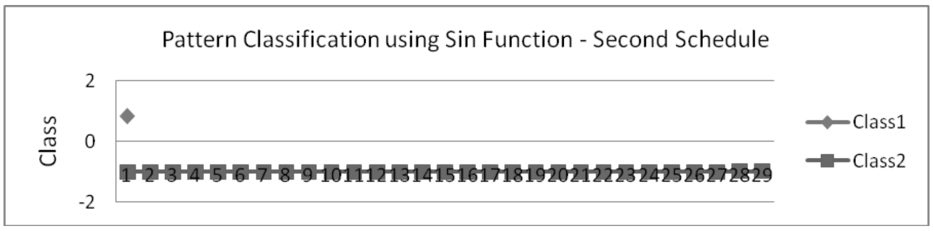
(a): The number of patterns in each class using Tanh response Function.



(b) : The No. of patterns in each class using ½(1+tanh) response Function.

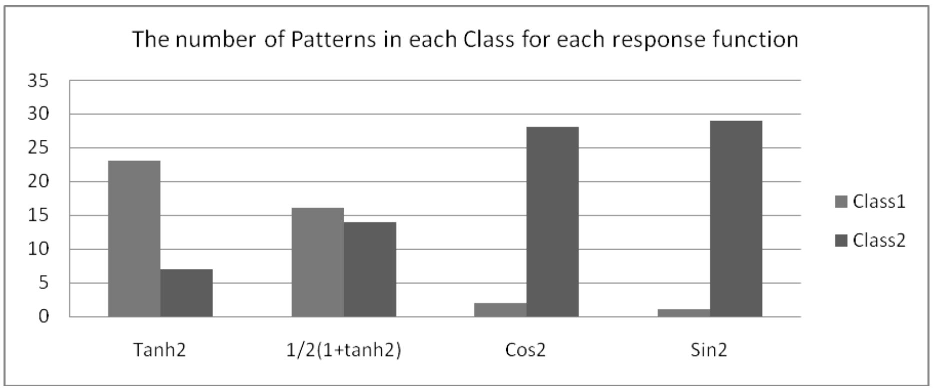


(c): The No. of patterns in each class using Cos response Function.



(d): The No. of patterns in each class using Sin response Function.
Figure(4): Pattern Classification using the response function – the 2nd Schedule.

Figure(5) shows the number of patterns in each class. The difference between the number of patterns in each class when using the $\frac{1}{2}\ (1 + tanh)$ response function is less than the difference when using the tanh response function, which is less than its counterpart using the cos response function. The effect of using the sin response function makes the difference is the largest.

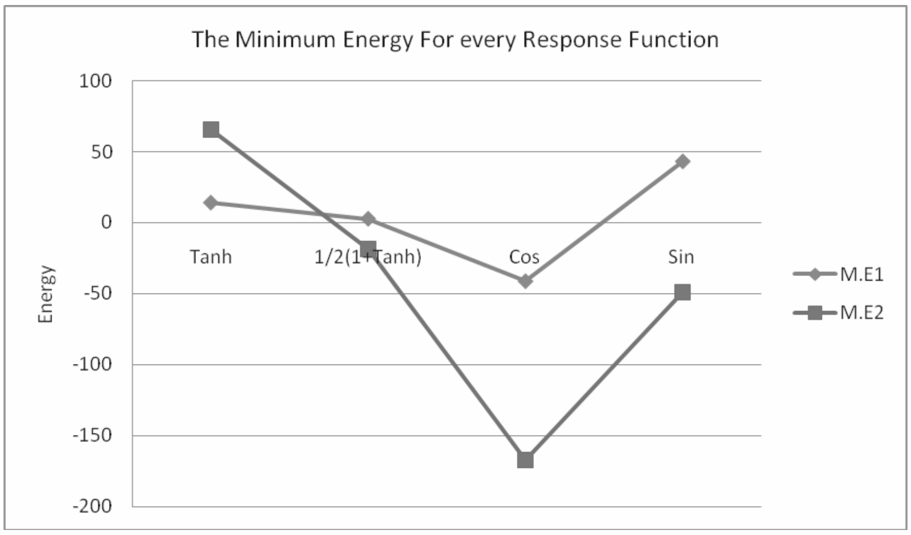


Figure(5): the number of patterns in each class for the response functions- 2nd Schedule.

On the other hand, in case of the Circuit Bipartitioning problem we find that the partitioning imbalance using $\frac{1}{2}\,(1 + tanh)$ response function is less than the partitioning imbalance using the tanh response function which is less than the difference between the number of wires for the two boards (classes) using cosine response function. Using the sine response function, we find that the partitioning imbalance is the largest among all the response functions. In this schedule, there is no partitioning balance for any response function.

Comparing Figure(3) and Figure(5), we find that the annealing schedule affecting the number of patterns in each class. For the circuit bipartitionig problem, we find that the two boards contain different number of wires for the same response function for the Cauchy annealing schedule and linear annealing schedule.

Figure(6) shows that the cosine response function using the second schedule gives the lowest energy function. In schedule1, we find that the minimum energy using the cosine response function is less than the minimum energy using the $\frac{1}{2}\,(1 + tanh)$ response function which is less than the minimum energy of using tanh response function. The minimum energy using sine response function is the largest one among all the others in this schedule. On the other hand, in schedule 2 we find that: the minimum energy when using the cosine response function is less than the minimum energy when using the sine response function which is less than the minimum energy when using $\frac{1}{2}\,(1 + tanh)$ response function. The minimum energy when using tanh response function is the largest one among all the others in this schedule.



Figure(6): the Minimum Energy for the two Schedules for each response function.

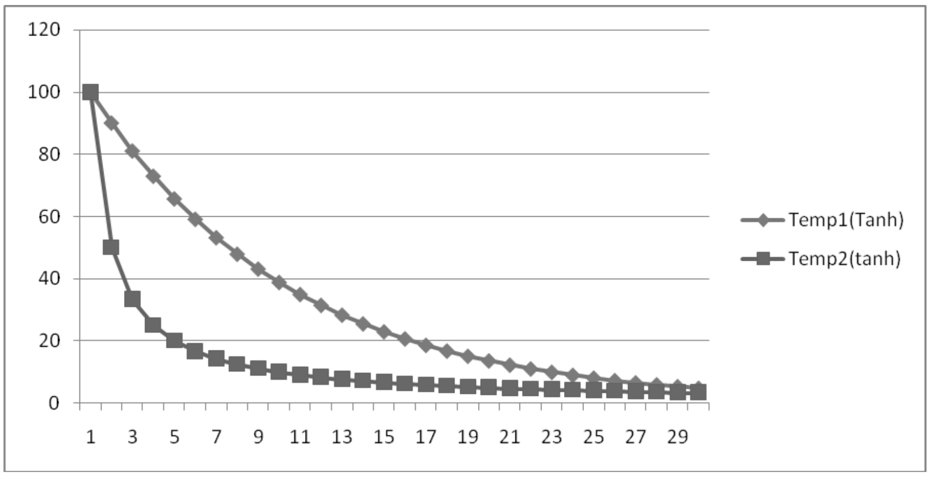
Table(2) represents the end values of the states when the second schedule is used. The values of the states converged to +1 or converged to -1 similar to the values of states in table(1). The patterns contained in each class differs from class to class using the same response function or using different response functions. Also the annealing schedule affects the number of patterns contained in each class.

Table(2) : the end value of each state at different functions using the 2nd schedule.

State	Tanh	1/2(1+tanh)	Cos	Sin
S1	1	-0.99380788981618	-1	-1
S2	1	-0.99961186188042	-1	0.83365460701215
S3	1	0.99991930493661	-1	-1
S4	1	-0.99999999999961	-1	-1
S5	0.999999999999998	-0.9999999959990	-0.9999999995747	-1
S6	0.99999999996152	-0.99994143646702	-1	-1
S7	1	0.99999970602817	-1	-1
S8	1	1	-1	-1
S9	0.999999999962372	-0.99999999999998	-1	-1
S10	1	-0.93748842328295	-1	-1
S11	1	0.99999999999967	-1	-1
S12	1	1	-1	-1
S13	1	0.9999999999999	0	-1
S14	-0.99999999999999	1	-1	-1
S15	0.999999999999961	-0.9999999999999	-1	-1
S16	-0.99999999999998	1	-1	-1
S17	-0.99822084371429	-0.99921787980353	-1	-1

S18	1	0.99999999999999	-1	-1
S19	1	-0.99999999980890	-1	-1
S20	-1	1	-1	-1
S21	-0.42801569048668	-0.97802611846943	-1	-1
S22	1	0.99999999999998	1	-1
S23	1	0.99999999999999	-1	-1
S24	1	-0.99999999999999	-1	-1
S25	-0.99999999999996	0.99999999999999	-1	-1
S26	-0.99999999999999	1	-1	-1
S27	0.999999999869405	-0.99999999999999	-1	-1
S28	1	1	-1	-1
S29	0.999999999999924	0.99999999999999	-1	-0.97600699776045
S30	1	-0.99999999890035	-1	-0.98143955255299

Figure(7) shows a comparison between the behavior of the temperature in each schedule. The Cauchy annealing schedule decreases the temperature more rapidly than the linear annealing schedule. On the other hand the Cauchy annealing schedule converges better than the linear annealing schedule. The randomness occurs rapidly in the first ten iterations when using the second schedule rather than using the first schedule.

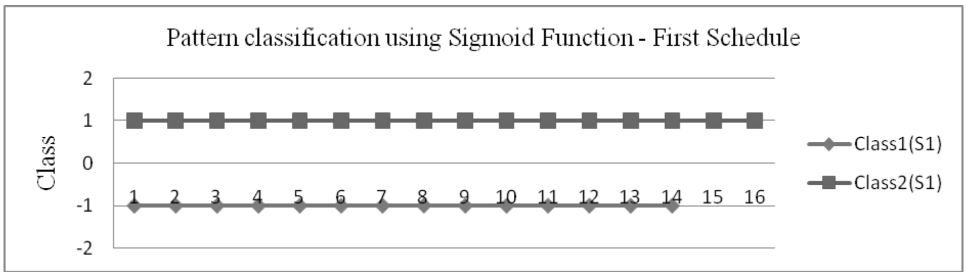


Figure(7) : the behavior of the temperature for the two schedules.

Table(3) shows the minimum energy which represents the cost function using the different response functions. From the table, we find that the minimum energy results from using the cosine function, while the largest one results from using the response sine function.

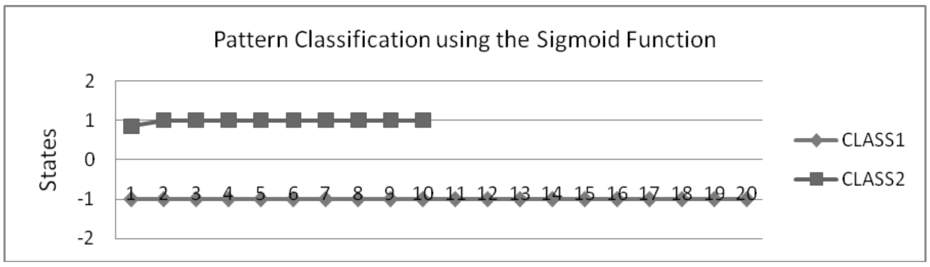
Table (3) the minimum energy for each schedule using different functions.

Function	Tanh	1/2(1 + Tanh)	Cos	Sin
M.E (S1)	13.925767	99.4697155	-40.796316	216.54952
M.E(S2)	65.382885	-16.53134	197.30314	346.353



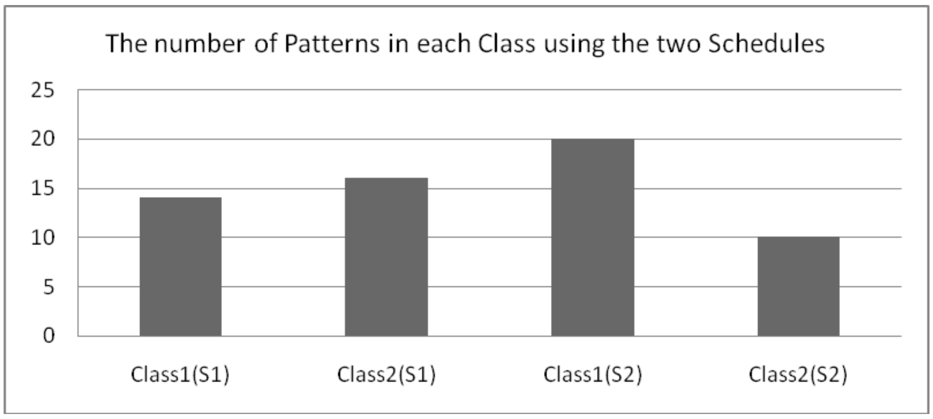
Figure(8) Classification of patterns using the Sigmoid Function – 1st Schedule.

Figure(8) shows that patterns are classified in two classes. The first class contains number of patterns greater than the number of patterns in the second class when using the first schedule.



Figure(9) Pattern classification using the response function (Sigmoid) – 2nd Schedule

Figure(9) shows that there exists two partitions of classes. The first class contains number of patterns less than the number of patterns in the second class when using the second schedule.



Figure(10) Pattern Classification using Sigmoid Function for the Two Schedules.

Figure(10) shows that the number of patterns in the first class when using schedule 2 is greater than the number of patterns in the second class when using the schedule 1, which is greater than the number of patterns in the first class when using the first schedule. While the second class contains the lowest number of patterns when using the second schedule . For the case of circuit bipartitioning problem, we find that the partitioning imbalance when using the first schedule is less than its counterpart when using the second schedule for applying the sigmoid response function.

Table(4) shows the minimum energy for the two schedules using the sigmoid function. In the Cauchy annealing, the minimum energy is less than the minimum energy of the linear annealing function.

Table(4): the minimum energy for the two schedules using the sigmoid function.

The Schedule	T(k) = T(0) / k	T(k) = α * T(k – 1)
M.E	-132.201801024419	-16.5283015519348

Table(5) shows the mean value of the number of patterns in each class using all the response functions in the two schedules. We find that the mean value of the number of patterns in the first class using all the response functions in the first schedule is greater than the number of patterns in the first class when using the second schedule, while the number of patterns in the second class when using the first schedule is less than the number of patterns

Table(5) : Mean and standard deviation for the number of patterns in each class

Class	Tanh	½(1+tanh)	cos	sin	sigmoid	Mean	STD
Class1(S1)	18	22	15	24	14	18.6	4.3
Class2(S1)	12	8	15	6	16	11.4	4.3
Class1(S2)	23	16	2	1	20	12.4	10.3
Class2(S2)	7	14	28	29	10	17.6	10.3

in the second class when using the second schedule. The standard deviation when using the first schedule is less than its counterpart for the second schedule. In case of using the circuit bipartitioning problem, we find that the mean value of partitioning imbalance = 18.6 – 11.4 = 7.5 when using the first schedule is greater than the partitioning imbalance when using the second schedule, which = 17.6 – 12.4 = 5.2 for all the response functions.

For Circuit Bi-partitioning, the partitions imbalance which is the difference between the numbers of nodes of the partitions using the two schedules is shown in table(6).

Table(6): partition imbalance for circuit bi-partitioning into 2 classes for the 2 schedules

Schedule	Tanh	½(1+tanh)	cos	sin	sigmoid
S1	6	14	0	18	2
S2	16	2	26	28	10

Finally, the performance evaluation of stochastic simulated annealing and mean field annealing is discussed as follows:

- Stochastic simulated annealing is slow, in part because of the discrete nature of the search through the space of all configurations. MFA is an alternate, faster method that allows each node to take on analog values during search; at the end of the search the values are forced to be $\pm$ 1, as required by the optimization problem.
- In SSA, searching the configurations space is of a discrete nature, while it is of a continuous nature in MFA.
- In SSA, the moves directions are chosen at random, while MFA is just like SSA except that the (apparently) best move is always chosen.

- MFA is deterministic in nature and this gives the advantages of faster convergence to the equilibrium temperature, compared to SSA.
- The MFA decrease in computational effort than SSA.
- MFA simplify the analysis that has been adopted at various instances than SSA.
- MFA replaces explicit stochastic simulations by Expectations.

6. Discussion and Conclusions

Mean Field Annealing combines the collective computation property of Hopfield Neural Network model with the annealing notion of simulated annealing in order to form a better algorithm. In Mean Field Annealing, discrete variables called spins (neurons) are used for encoding the combinatorial optimization problems. It consists of setting an annealing schedule and then at each temperature finding an equilibrium analog value for every spin. This analog value is merely the expected value of the discrete state of a neuron in a system at temperature T.

In this paper, two annealing schedules are discussed: the linear annealing schedule and the Cauchy annealing schedule and five response functions are used namely: the tanh function, $\frac{1}{2}(1 + tanh)$ function, the cosine function, the sine function, and finally the sigmoid function. In this paper, we state the following conclusions:

1. The mathematical derivation of the Mean Field and Mean Field approximation is derived.
2. A real classification problem known as graph bipartitioning is applied on the Circuit Bipartitioning problem.
3. The patterns are classified into two classes for graph bipartitioning problem, and into two boards for the circuit bipartitioning problem.
4. For the circuit bipartitioning problem, the partitioning imbalance is calculated and compared using the two schedules for all the response functions and shows that: the partitioning imbalance using the linear annealing schedule is less than the partitioning imbalance using the Cauchy annealing schedule.
5. Partitioning balance occurs using the cosine response function for the first schedule.
6. The mean value and standard deviation of the number of patterns in each class are calculated and compared showing that the mean value of the number of patterns (wires) in the 1st class (first board) is greater than the mean value of that in the 2nd class (second board) using the 1st schedule for all the response functions. On the other hand, the Mean value of the

number of patterns in the 2nd class (2nd board) is greater than the mean value of that in the 1st class (1st board) using the 2nd schedule for all the response functions.

7. The standard deviation when using the first schedule is less than the standard deviation when using the second schedule.
8. The minimum energy is calculated and compared for all the response functions using the two schedules.
9. Cauchy annealing converges better than linear annealing.
10. The annealing process using the linear annealing is better than its counterpart of the Cauchy annealing.
11. The Minimum Energy (the cost function) using the sigmoid response function is the lowest among all the other response functions when using the second schedule.

References

1. Choi J., Qiu J., Pierce M., and Fox G.:" Generative Topographic Mapping by Deteministic Annealing". Procedia Computer Science, (2010) 1-10.
2. Angelini L, Nardulli G., Nitti L., Pellicoro M., Perrino D., and Stramaglia S.: " Deteministic annealing as a jet clustering algorithm in hadronic collisions". Physics Letters B 601 (2004) 56-63.
3. Rossi F., and Villa N.: " Topologically Ordered Graph Clustering via Deterministic Annealing". ESANN'2009 proceedings, European Symposium on Artificial Neural Networks – Advances in Computaional Intelligence and Learning, 2009.
4. Sharma P., Salapaka S., and Beck C.: " A Deterministic Annealing Approach to Combinatorial Library Design for Drug Discovery". American Control Conference, USA, 2005.
5. Chang M., Lin C., and Weng R.: " Adaptive Deterministic Annealing for two Applications: Competing SVR of Switching Dynamics and Travelling Salesman Problems". Proceedings of the 9th International Conference on Neural Information Processing ((ICONIP'02), vol. 2, 2002.
6. Czabanski R., " Neuro-Fuzzy Modeling Based on a Deterministic Annealing Approach". International Journal of Applied Mathematics & Computer Science, Vol. 15, No. 4, 561-576, 2005.
7. Rao A., Miller D., Rose K., and Gersho A.: "A Deterministic Annealing Approach for Parsimonious Design of Piecewise Regression Models". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 21, No.2, 1999.
8. Rose K.:"Deteministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems". Proceedings of the IEEE, Vol, 86, No. 11, 1998.
9. Bilbro G., Snyder W., Garnier S., and Gault J.: "Mean Field Annealing: A Formalism for Constructing GNC-Like Algorithms". IEEE Tansactions on Neural Networks, Vol. 3, No. 1, 1992.
10. Popoviciu N., and Boncut M.: " On the Hopfield algorithm. Foundations and examples". General Maathematics Vol. 13, No. 2, 2005.
11. Duda R., Hart P.,and Stork D.: "Pattern Classification" Second Edition, 2001.
12. Eldos T.: "Simulated Annealing with Deterministic Decisions". Journal of Computer Science 5 (12) :977-982, 2009.
13. Suman B., and Kumar P.: "A survey of simulated annealing as a tool for single and multiobjective optimization". Journal of the Operational Research Society (2006) 57, 1143-1160.
14. Salakhutdinov R., and Hinton G.: " Deep Boltamann Machines". Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), 2009.

15. Mishra D., Shukla A., and Kalra P.: " OR-Neuron Based Hopfield Neural Network for Solving Economic Load Dispatch Problem". Neural Information Processing – Letters and Reviews, Vol. 10, No 11, 2006.
16. Jimenez A., Tiampo K., and Posadas A.: " An Ising model for earthquake dynamics". Nonlinear Processes in Geophysics, 2007.
17. Feng Y.: "The Boundary Conditions Geometry in Lattice-Ising Model". Electronic Journal of Theoretical Physics 7 (2005).
18. He H., and Sykora O.: " A Hopfield Neural Network Model for the Outerplannar Drawing Problem". International Journal of Computer Science, 32:4, IJCS-32_4_17, 2006.
19. Mishra D., and Kalra P.: " Modified Hopfield Neural Approach for Solving Nonlinear Algebraic Equations". Engineering Letters 14:1, EL_14_1_23, 2007.
20. Pavlovic V., and Friedman D.: "Enhancement of Hopfield Neural Networks using Stochastic Noise Processes". IEEE workshop on Neural Networks for Signal Processing, 2001.
21. Zhang W., and Tang Z.: " A New Algorithm Using Hopfield Neural Network with CHN for N-Queens Problem". International Journal of Computer Science and Network Security, Vol.9 No.4, 2009.
22. Sheng J., and Liu H.: "Stereo vision using a microcanonical mean field annealing neural network". Computer Network System. 8, 1997.
23. Lee K.: "A Neural Network Model Based on Graph Matching and Annealing : Application to Hand-Written Digits Recognition". International Journal of Mathematics and Computers in Simulation. Issue 4, Volume 1, 2007.
24. Strausz G.: "Mean-field approximation with neural network". IEEE International Conference on Intelligent Engineering Systems, 1997.
25. Agra R., Wijland F., and Trizac C.: "On the free energy within the mean-field approximation". European Journal of Physics, 27 (2006) 407-412.
26. Vedral V.: "Mean-field approximations and multipartite thermal correlations". New Journal of physics 6 (2004).
27. Nair T and Sooda K. "Comparison of Genetic Algorithm and Simulated Annealing Technique for Optimal Path Selection in Network Routing". NCVN-09 octobar 2009 KCG College of Technology.
28. Elhadded Y and Sallabi O. "A New Hybrid Genetic and Simulated Annealing Algorithm to Solve the Travelling Salesman Problem". Proceedings of the World Congress on Engineering 2010 Vol1.
29. Manconi A., Tizzani P., Zeni G., Pepe S and Solaro G. "Simulated Annealing and Genetic Algorithm Optimization using COMSOL Multiphysics: Applications to the Analysis of Ground Deformation in Active Volcanic Areas". Proceedings of the COMSOL Conference 2009 Milan.

30. Alan R., McKendall J, Shang J., and Kuppususamy S.: “Simulated annealing heuristics for the dynamic facility layout problem”. Computers & Operations Research 33 (2006).
31. Bisht S.: “Hybrid Genetic-simulated Annealing for Optimal Weapon Allocation in Multilayer Defence Scenario”. Defence Science Journal, Vol 54, No. 3, 2004.
32. Ho S., Yang S., Wong H., and Ni G.: “ A Simulated Annealing Algorithm for Multiobjective Optimizations of Electromagnetic Devices”. IEEE Transactions on Magnetics, Vol. 39, No. 3, 2003.
33. Simopoulos D., Kavatza S., Vournas C. : “ Unit Commitment by an Enhanced Simulated Annealing Algorithm”. Paper no. TPWRS- 00234-2005. PSCE, 2006.
34. Haidine A., and Lehnert R. : “Multi-Case Multi-Objective Simulated Annealing (MC-MOSA): New Approach to Adapt Simulated Annealing to Multi-objective Optimization”. International Journal of Information Technology 4:3 2008.
35. Duda R., Hart P., and Stork D. : “Simulated Annealing” PATTERN CLASSIFICATION, Second Edition, 2001.
36. Chen D., Lee C., Park C., and Mendes P.: “Parallelizing simulated annealing algorithms based on high-performance computer”. Journal of Global Optimization, 20007.
37. Shi H., and Li W.”Evolving Artificial Neural Networks Using Simulated Annealing-based Hybrid Genetic Algorithms”. JOURNAL OF SOFTWARE, VOL.5, NO.4, 2010.