Physica A 388 (2009) 1024–1030

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Stochastic optimization of spin-glasses on cellular neural/nonlinear network based processors

M. Ercsey-Ravasz [a,b,*], T. Roska [a], Z. Néda [b]

[a] Pázmány Péter Catholic University, Department of Information Technology, HU-1083 Budapest, Hungary

[b] Babeş-Bolyai University, Department of Physics, RO-400084 Cluj, Romania

ARTICLE INFO

Article history:
Received 17 June 2008
Received in revised form 7 October 2008
Available online 3 December 2008

PACS:
07.05.Mh
05.10.-a
89.20.Ff

Keywords:
Cellular neural/nonlinear networks
Spin–glass systems
Stochastic optimization

ABSTRACT

Nowadays, Cellular Neural/Nonlinear Networks (CNN) are practically implemented in parallel, analog computers, showing a fast developing trend. It is important also for physicists to be aware that such computers are appropriate for implementing in an elegant manner practically important algorithms, which are extremely slow on the classical digital architecture. Here, CNN is used for optimization of spin-glass systems. We prove, that a CNN in which the parameters of all cells can be separately controlled, is the analog correspondent of a two-dimensional Ising type spin-glass system. Using the properties of CNN we show that one single operation on the CNN chip would yield a local minimum of the spin-glass energy function. By using this property a fast optimization method, similar to simulated annealing, can be built. After estimating the simulation time needed for this algorithm on CNN based computers, and comparing it with the time needed on normal digital computers using the classical simulated annealing algorithm, the results look promising: a speed-up of the order 10^{12} is expected already at 50×50 lattice sizes. Hardwares realized nowadays are of 128×128 size. Also, there seem to be no technical difficulties adapting CNN chips for such problems and the needed local control of the parameters could be fully developed in the near future.

1. Introduction

There are several good reasons why new computational paradigms are nowadays needed for (at least) supplementing classical digital computers. In the last decades a fast increasing speed of the newly appearing processors has been observed [1], but nowadays this process is slowing down. The characteristic size of the elements in a CMOS chip is getting close to the atomic size, and also the power dissipation limit has been almost reached. This has saturated the clock frequency in the last years, and instead the number of processors is increasing, many times leading to a cellular, locally high-speed — globally lower-speed architecture [2,3]. The fast development of sensors producing analog signals waiting for real-time processing and the worrying increase in power consumption also indicate that new paradigms are needed.

Many new trends appeared in the last decades: parallel computing, grid computing, quantum computing, DNA computing or membrane computing. All of them show different types of advantages, but also a lot of difficulties and limiting circumstances. Cellular neural/nonlinear network (CNN) computing is more popular in the engineering communities and much less known among physicists. Relative to all other trends, the main advantage is that several practical realizations of CNN computers are already available.

* Corresponding author at: Pázmány Péter Catholic University, Department of Information Technology, HU-1083 Budapest, Hungary.
E-mail addresses: ravasz@digitus.itk.ppke.hu, ravaszmarek@yahoo.com (M. Ercsey-Ravasz), roska@itk.ppke.hu (T. Roska), zneda@phys.ubbcluj.ro (Z. Néda).

0378-4371/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2008.11.037

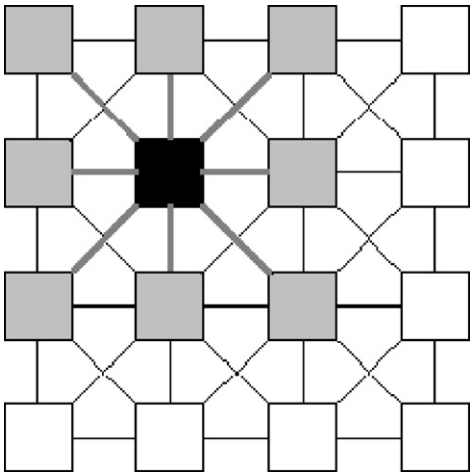

Fig. 1. The lattice structure of CNN.

After the idea of CNN appeared in 1988 [4] a detailed plan for a CNN computer was developed in 1993 [5]. Since then the chip has had a fast developing trend and the latest – already commercialized – version is the Q-Eye chip with lattice size 176 × 144, included in the stand-alone camera computer, Eye-Ris [6]. The physics community can also benefit from CNN based computers. It was shown in several previous studies that this novel computational paradigm is useful in solving partial differential equations [7,8], studying cellular automata models [9,10], doing image processing in real time [11], and performing Monte Carlo simulations on lattice models [12,13]. When testing new computing paradigms, it is important to study those complex problems, for which the algorithms are extremely slow on digital computers. Here we argue that CNN computers are suitable for implementing fast and elegant optimization algorithms. The problem studied here is the energy minimization of Ising-type spin-glass systems.

In the next section we will describe briefly the structure and dynamics of cellular neural/nonlinear networks together with the most developed applications of CNN computing. In the third section we prove that a CNN computer on which the parameters of each cell can be separately controlled, can be viewed as the analog correspondent of a locally coupled two-dimensional spin-glass system. Optimization (energy minimization) is done by the circuit system itself, one single operation on this analog computer leading to a local minimum of the spin-glass energy function. This finding can be exploited in an elegant manner. After introducing a suitable noise, a very fast stochastic optimization algorithm can be developed.

The local control of some parameters of the cells is already partially realized on some hardwares (for details see Section 2) and there is no technical obstacle to be fully realized in the near future. Beside the fundamental interest for optimization algorithms, the importance of this study consists also in motivating the development of hardwares in such direction.

2. Cellular neural/nonlinear networks and CNN computers

The CNN Universal Machine (CNN-UM) [5] is one special case of cellular wave computers [2] in which computation is achieved using the spatial-temporal dynamics of a cellular neural/nonlinear network [4]. In laymen terms one could define it as an analog computer with many locally interconnected calculation units that are capable of solving coupled first-order differential equations in completely parallel manner. The CNN is composed of $L \times L$ cells placed on a square lattice and interconnected through their neighbors [4]. Usually the 4 nearest and the 4 next-nearest neighbors (Moore neighborhood) are considered (Fig. 1). Each cell is characterized by a state value: $x_{i,j}(t)$ representing a voltage in the circuit of the cell. The cell has also an input value (voltage) $u_{i,j}$, which is constant in time and can be defined at the beginning of an operation. The third characteristic quantity of the cell is the output value $y_{i,j}(t)$. This is equivalent with the $x_{i,j}$ state value in a given range. More specifically it is a piece-wise linear function, bounded between -1 (called as white) and 1 (black): $y = f(x) \equiv \frac{1}{2}(|x+1| - |x-1|)$.

The wiring between neighboring cells assures that the state value of each cell can be influenced by the input and output values of its neighbors. The equation governing the dynamics of the CNN cells results from the time-evolution of the equivalent circuits. Supposing the 8 Moore neighbor interactions it has the following form [4]:

$$\frac{\mathrm{d}x_{i,j}(t)}{\mathrm{d}t} = -x_{i,j}(t) + \sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1} A_{i,j;k,l}y_{k,l}(t) + \sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1} B_{i,j;k,l}u_{k,l} + z_{i,j} \tag{1}$$

where i, j denotes the coordinates of the cell and the summation indices k, l are for its neighbors. Self-interaction ($k = i, l = j$) is also possible. The set of parameters $\{A, B, z\}$ is called a template and controls the whole system. An operation is performed by giving the initial states of the cells, the input image (the input values of all cells) and by defining a template.

The states of all cells will vary in parallel and the result of the operation will be the final steady state of the CNN. If the state values ($x_{i,j}$) of all cells remain bounded in the $[-1, 1]$ region (i.e. $y_{i,j} = x_{i,j}$ holds for each cell at any time t), than each operation is equivalent with solving a differential equation defined by the template itself [14,7,8]. When $x_{i,j}$ does not remain bounded between the -1 and $+1$ values, than the piece-wise linear function described at the definition of the output value $y_{i,j}$, takes an important role. The final steady state will not be simply the solution of the differential equation, and this case can be used for defining other useful operations as well [14].

The CNN-UM [5] is a programmable analogic (analog & logic) cellular wave computer. Beside the analog circuits leading to the evolution equations described by Eq. (1), each cell contains also a logic unit, local analog and logic memories and a local communication and control unit. The logic unit and logic memories are included to complement the analog computation. In this manner basic logic operations can be performed without defining complicated templates for it. In the local logic memories one can save a binary value (1 and 0 respectively), and in the local analog memories it is possible to save real values between -1 and 1. Since the CNN array is mainly used for image processing and acquisition, the binary values are often referred as black and white, and the real values bounded between -1 and 1 are mapped in a gray-scale scheme. Beside these local units, the CNN-UM has also a global analog programming unit which controls the whole system, making it a programmable computer. It can be easily connected to PC type computers and programmed with special languages, for example the Analogic Macro Code (AMC).

The physical implementations of these computers are numerous and widely different: mixed-mode CMOS, emulated digital CMOS, FPGA, and also optical. For practical purposes the most promising applications are for image processing, robotics or sensory computing purposes [11], so the main practical drive in the mixed-mode implementations was to build a visual microprocessor [14]. In the last decades the size of the engineered chips was constantly growing, the new cellular visual microprocessor EYE-RIS [6] for example has 176×144 processors, each cell hosting also 4 optical sensors. Parallel with increasing the lattice size of the chips, engineers are focusing on developing multi-layered, three-dimensional chips as well.

In the original version of the CNN, the templates (coupling parameters) are identical for all cells. This means that for example $A(i, j; i+1, j)$ is the same for all (i, j) coordinates. In such way, on the two-dimensional CNN chip, all the A couplings are defined by a single 3×3 matrix. Totally, $9+9+1 = 19$ parameters are needed to define the whole, globally valid, template $\{A, B, z\}$. On the latest version of the CNN chips (ACE16K, EYE-RIS) the $z(i, j)$ parameter can already be locally varied. It is expected that on newer chips one will be able to separately control also the $A(i, j; k, l)$ and $B(i, j; k, l)$ connections as well.

Many applications ideal for the analogic and parallel architecture of the CNN-UM were already developed and tested. Studies dealing with partial differential equations [7,8] or cellular automata models [9,10] prove this. It was also proved that CNN computers are suitable for stochastic simulations. The natural noise of the analog chip can be efficiently used to generate random numbers approximately 4 times faster than on digital computers [12]. Experiments in which the site-percolation problem and the two-dimensional Ising model is properly solved were done on the ACE16K chip (with 128×128 cells) [12,13].

3. Optimization of spin-glass systems on CNN architectures

The aim of the present study is to prove that CNN computing is also suitable for efficiently solving complex optimization problems on spin-glass type lattice models. The importance of this problem consists also in the fact that optimization of spin-glasses is in many cases an NP-hard problem. As shown by Barahona [15] and Istrail [16], the problem is polynomial if the simplest two-dimensional square lattice structure – with the first 4 neighbors – is used, but becomes already NP-hard when the 4 nearest and 4 next-nearest neighbors are considered (see Fig. 1). In general the optimization of Ising-type spin-glasses is an NP-hard problem on graphs that have genus scaling faster than the logarithm of the number of vertices. On regular lattices the condition for this is that the edges on which there are interactions should intersect each other, or in other words, the graph should be non-planar. Here a special case of non-planar graphs are considered: as stated before the case of a square lattice with nearest and next-nearest interactions, since this is the lattice structure that is most commonly used on CNN chips. Also, it has to be mentioned that CNN chips with more layers were also built (CACE1k [17]), and our optimization algorithm could be easily generalized for three-dimensional lattices as well.

We consider a two-dimensional space-variant CNN. This means that the templates (parameters of Eq. (1)) can be locally controlled. Matrix A is considered symmetric, and the elements are bounded:

$$A(i, j; k, l) = A(k, l; i, j), \tag{2}$$

$$A(i, j; i, j) = 1, \tag{3}$$

$$A(i, j; k, l) \in [-1, 1], \{i, j\} \neq \{k, l\} \tag{4}$$

where (i, j) and (k, l) denote two neighboring cells. Matrix B, which controls the effect of the input image, will be taken simply as:

$$B(i, j; i, j) = b, \tag{5}$$

$$B(i, j; k, l) = 0, \{i, j\} \neq \{k, l\}. \tag{6}$$

Parameter z is chosen as $z = 0$, so finally our template is defined by $\{A, b\}$ alone. The state equation of the system writes as:

$$\frac{dx_{i,j}(t)}{dt} = -x_{i,j}(t) + \sum_{\langle k,l\rangle \in N(i,j)} A_{i,j;k,l} y_{k,l}(t) + b u_{i,j} \tag{7}$$

where $x_{i,j}$ is the state value, $y_{i,j}$ is the output, and $u_{i,j}$ is the input of the cell (i, j) with neighborhood $N(i, j)$ (8 neighbors and itself).

In an earlier work Chua et al. [4] defined a Lyapunov function for the CNN, which behaves like the "energy" (Hamiltonian) of the system. For the CNN defined above it can be written simply as

$$E(t) = -\sum_{\langle i,j;k,l\rangle} A_{i,j;k,l} y_{i,j} y_{k,l} - b \sum_{i,j} y_{i,j} u_{i,j}, \tag{8}$$

where $\langle i, j; k, l\rangle$ denotes pairs of Moore neighbors, each pair taken only once in the sum. $y_{i,j}$ denotes the output value of each cell and $u_{i,j}$ stands for an arbitrary input image. We should observe that in the $[-1, 1]$ region, where the state values and output values are equal, $x_{ij} = y_{ij}$, this function is equivalent with the energy of the system, obtained by applying the work-energy theorem (the derivative of this function: dE/dy_{ij} is the same as $-dx_{ij}/dt$ in Eq. (7)). The reason why this Lyapunov function is used instead of the rigorously defined energy function, is that it contains only the output and input values of the cells. This function is also similar to the one used by Hopfield in Ref. [18].

By choosing the parameter $b = 0$, the Lyapunov function of this special CNN is similar with the energy of an Ising type system on a square lattice with locally varying coupling constants. The difference is that Ising spins are defined as ± 1, while here we have continuous values between $[-1, 1]$. Since in the space variant template the $A(i, j; k, l)$ coupling constants can be positive and negative as well, locally coupled spin-glasses can be mapped in such systems. In the following we will be especially interested in the case when the $A(i, j; k, l)$ couplings lead to a frustration and the quenched disorder in the system is similar with that of spin-glasses [19,20].

As demonstrated by Chua et al. this Lyapunov function has two important properties [4]:

(1) it is always a monotone decreasing function in time, $dE/dt \leq 0$, so starting from an initial condition E can only decrease during the dynamics of the CNN.
(2) the final steady state is a local minimum of the energy: $dE/dt = 0$.

In addition to these, our CNN has also another important property: due to the fact that all self-interaction parameters are $A(i, j; i, j) = 1$, it was shown in Ref. [4] that the output values of the cells in the final steady state will be always either 1 or -1. The local minima achieved by the CNN is thus an Ising-like configuration. Exceptions could be the cells (spins) with 0 local energy. It might very well happen that these cells will not achieve a ± 1 state (although on real hardwares, in presence of real noise this is highly improbable). The state of these will not affect the energy of the Ising configuration. One can randomly choose -1 or 1 for them. The result given by the CNN for these variables is equivalent with a kind of mixed state. We can conclude thus, that starting from any initial condition the final steady state of the template – meaning the result of a single operation – will be always a local minimum of the spin-glass type Ising system with local connections defined by the A matrix. The important result that one single operation is needed for finding a local minimum of the energy, gives hope to develop a fast optimization algorithm.

As already emphasized, the complex frustrated case (locally coupled spin-glass type system), where the A coupling parameters generates a non-trivial quenched disorder, will be considered here. The minimum energy configuration of such system is searched by an algorithm which is similar with the well-known simulated annealing method [21]. The noise is included with random input images ($u_{i,j}$ values in Eq. (8)) acting as an external locally variable magnetic field. The strength of this field is governed through parameter b. Whenever b is different from zero, our CNN template minimizes the energy expressed in Eq. (8). The first term in this expression is the energy of the considered spin-glass type model and the second part an additional term, which becomes minimal when the state of the system is equal to the input image (the external magnetic field). If b is large, the result will be the input image itself, if $b = 0$ the result is a local minimum of the pure Ising-type system. For values in between, the result is a "compromise" between the two cases. Decreasing slowly the value of b will result in a process similar with simulated annealing, where the temperature of the system is consecutively lowered. First big fluctuations of the energy are allowed, and by decreasing these we slowly drive the system to a lower energy state. Since the method is stochastic, one can never be totally sure that the global minimum will be ever achieved, but certainly good approximations can be obtained.

The outline of the algorithm is the following:

(1) The $A(i, j; k, l)$ connection parameters are fixed.
(2) One starts from a random initial condition x, and $b_0 = 5$ (with this value the result of the template is almost exactly the same as the input image).
(3) A binary random input image u is generated with $1/2$ probability of black ($+1$ value) pixels.
(4) Using the x initial state and the u input image the CNN template, $\{A, b\}$, is applied.
(5) The value of b is decreased with values Δb.
(6) Items (3)–(5) are repeated until $b = 0$ is reached. The result of the previous CNN operations are considered as the initial state for the next operation.

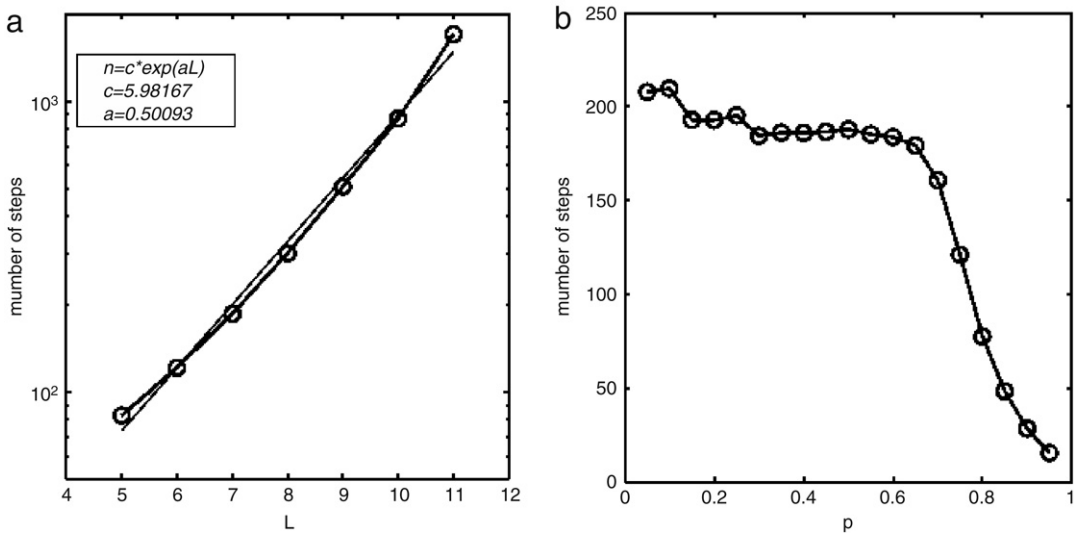

Fig. 2. (a) The number of steps (each optimization attempt consists of $b_0/\Delta b = 100$ steps) needed to find the optimal energy as a function of the lattice size L. The density of positive connections is fixed at $p = 0.4$, and the parameter $\Delta b = 0.05$ is used. (b) For a system with size $L = 7$ the number of steps needed for getting the presumed global minima is plotted as a function of the probability of positive connections p.

(7) When reaching $b = 0$ the image (Ising spin configuration) is saved and the energy is calculated. This concludes one optimization attempt.

In the classical simulated annealing algorithm several thousands of Monte Carlo steps are needed for a single temperature. Here, the CNN template working totally in parallel, one operation replaces all these steps. Similarly with the problem of choosing appropriate cooling rate in simulated annealing, choosing the value of Δb is also a delicate task. A proper value providing an acceptable compromise between the quality of the results and speed of the algorithm has to be found. For each system size one can find an optimal value of Δb, but as one would naturally expect, this is rapidly decreasing by increasing the system size. It is much more effective, both for performance (meaning the probability of finding the real global optimum) and speed, to choose a constant $\Delta b = 0.05$ step and repeat the whole cooling process several times. As a result, several different final states will be obtained, and we have a higher probability to get the right global minima between these.

For testing the efficiency of the algorithm one needs to measure the number of optimization attempts necessary for finding the right global minima. To do this, one needs a previous knowledge of this global minima. For small systems with $L = 5, 6$ this can be obtained through a quick exhaustive search in the phase-space. For bigger systems the classical simulated annealing algorithm can be used. In such cases the temperature was decreased with a rate of 0.99 ($T_{\text{final}}/T_{\text{ini}}$) and 1000 Monte Carlo steps were performed for each temperature.

4. Results

In the present work spin-glass systems with $A(i, j; k, l) = \pm 1$ local connections were studied. The p probability of the positive bonds was varied (influencing the amount of frustration in the system), and local interactions with the 8 Moore neighbors were considered. For several p densities of the positive links and various system sizes, we calculated the average number of steps (each optimization attempt consists of $b_0/\Delta b = 100$ steps) needed for finding the energy minimum. As naturally is expected for the non-trivial frustrated cases, the number of simulation steps needed increases exponentially with the system size. This means that the problem remains NP-hard on the CNN-UM as well. As an example, the $p = 0.4$ case is shown on Fig. 2(a). As it is observable on the figure, we could made estimates for relatively small system sizes only ($L < 12$). The reason is that we had to simulate also the CNN operations, and for large lattices a huge system of partial differential equations had to be solved. This process gets quite slow for bigger lattices.

The average number of steps needed to reach the estimated energy minima depends also on the p probability of the positive connections in the system. On Fig. 2(b) we illustrate this for a system with size $L = 7$. To obtain this data for each p value 5000 different systems were analyzed. As observable on Fig. 2(b) the system is almost equally hard to solve for all p values in the range of $p \in (0, 0.6)$.

Let us have now some thoughts about the estimated speed of such an optimization algorithm. As mentioned earlier, on the nowadays available CNN chips, only parameter z can be locally varied, parameters A and B are 3×3 matrices, uniformly applied for all cells. The reason for no local control of A and B seems to be simply the lack of practical motivations. In image processing applications no really useful algorithms were developed, which would require these space variant connections. Realizing the local control of A and B is technically possible and is expected to be included in the newer versions of the CNN chips. This modification would not change the basic properties and the speed of the chip, only the control unit and template

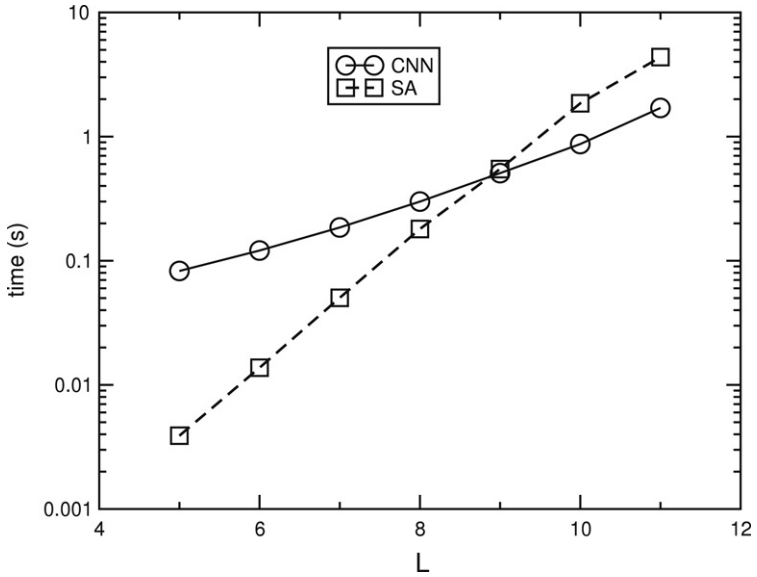

Fig. 3. Time needed to reach the minimum energy as a function of the lattice size L. Circles are for estimates on CNN computers and stars are simulated annealing results on 3.4 GHz Pentium 4. Results are averaged on 10 000 different configurations with $p = 0.4$ probability of positive bonds. For the CNN algorithm $\Delta b = 0.05$ was chosen. For simulated annealing the initial temperature was $T_0 = 0.9$, final temperature $T_f = 0.2$ and the decreasing rate of the temperature was fixed at 0.99.

memories would become more complicated. Also, introducing the connection parameters in the local memories of the chip would take a slightly longer time. In the specific problem considered here the connection parameters have to be introduced only once for each problem, so this would not influence in a detectable manner the speed of calculations.

Based on our previous experience with the ACE16K chip (with sizes 128×128) [12,13] we can make an estimation of the speed for the presented optimization algorithm. This chip with its parallel architecture solves one template in a very short time - of the order of microseconds. For each step in the algorithm one also needs to generate a random binary image. This process is already 4 times faster on the ACE16K chip than on a 3.4 GHz Pentium 4 computer and needs around 100 μs (see Ref. [12]). It is also helpful for the speed, that in the present algorithm it is not needed to save information at each step, only once at the end of each optimization attempt. Saving an image takes roughly 10 ms on the ACE16K, but this is done only once after several thousands of CNN operations. Making thus a rough estimate for our algorithm, a chip with similar properties like the ACE16K should be able to compute between 1000 and 5000 steps in one second, *independently of the lattice size*.

Using the lower estimate (1000 steps /second) and following the number of optimization steps needed in case of $p = 0.4$, the estimated average time for solving one problem is plotted as a function of the lattice size on Fig. 3 (circles). Comparing this with the speed of simulated annealing (SA) performed on a 3.4 GHz Pentium 4 (squares on Fig. 3), the results for larger lattice sizes are clearly in favor of the CNN chips. For testing the speed of simulated annealing we used the following parameters: initial temperature $T_0 = 0.9$, final temperature $T_f = 0.2$, decreasing rate of the temperature 0.99. Results were averaged for 10000 different bond distributions. From Fig. 3 it results that the estimated time needed for the presented algorithm on a CNN chip would be smaller than simulated annealing already at 10×10 lattice sizes. Before getting however too enthusiastic about this let us remind that the problem still remains NP-hard and the necessary time will increase exponentially with system size. After fitting the results, on the CNN-UM we predict an increase of the simulation time (t_{CNN}) as a function of system (L) as: $t_{\mathrm{CNN}} = 0.0079 \cdot \exp(0.47 \cdot N)$. This can be compared with the results obtained by simulated annealing on a 3.4 GHz Pentium 4 PC: $t_{\mathrm{PC}} = 0.000012 \cdot \exp(1.24 \cdot L)$. The factor in the exponent is almost three times lower for the CNN! This means for example that if we would like to study the problem on a lattice with sizes $L = 50$ on PC type computers 10^{13} years of computing would be necessary, in comparison with this on a CNN-UM 4 years of computations are enough. The speed-up is significant, although from this example one will quickly realize that due to the exponential increase of the computation time with system size, CNN is not enough for solving such NP-hard optimization problems on larger lattices. Besides its importance in condensed matter physics, spin-glass theory has in the time acquired a strongly interdisciplinary character, with applications to neural network theory [22], computer science [23], theoretical biology [24], econophysics [25] etc. There are many applications in which global minimum is not crucial to be exactly found, the minimization is needed only with a margin of error. In such cases the number of requested operations will decrease drastically. As an example, it has been shown that using spin-glass models as error-correcting codes, their cost-performance is excellent [26], and the systems are usually not even in the spin-glass phase. In this manner by using the CNN chip, finding acceptable results could be very fast, even on big lattices.

We also have to mention here that during our simulations with the above proposed algorithm the templates are run always for a fixed 10 time-units (time-unit set by the discharge time of the relevant capacitor in the circuit). This means that

in most of the cases the template settling time is not reached, in other words the output is not completely stabilized. There is simply no need for this, since anyhow a random noise is finally applied on the output. Due to this cutoff on the template settling time, the time necessary for one CNN operation is not affected by the system size. No additional scaling with system size is thus expected.

Finally, let us make some remarks regarding the influence of typical template errors that can be in the range of 3%–5% for the presently available analog implementations. The question that it worth looking for is if the asymmetric nature of A under such conditions (i.e. $A(i, j; kl)$ will not exactly be the same as $A(k, l; i, j)$) will cause any problems for the optimization. Simulations with noises up to $\pm 10\%$ values show that the results remain similar. The simulations presented on Fig. 2 were redone simulating such template errors and the obtained curves were practically the same as the ones plotted on Fig. 2.

5. Conclusion

A cellular neural/nonlinear network with locally variable parameters was used for developing a stochastic optimization algorithm for two-dimensional, Ising type spin-glass systems. By simulating the proposed optimization algorithm, promising perspectives for solving such problems were predicted: CNN computers could be faster than digital computers already at a 10×10 lattice size. The problem remains however still NP-hard and in the spin-glass regime the necessary optimization time will grow exponentially with increasing system size. Although the actual size of the commercial CNN chips is already much larger (128×128), one will not benefit too much from the increased system size, since reasonable computing times (of the order of weeks) are reached already for 35×35 lattice sizes. Chips with 2 and 3 layers of cells were also produced (for ex. CACE1k [17]) and increase in the number of layers is also expected. The presented algorithm can be generalized for three-dimensional lattices, and also other applications may become possible (the connection matrix of a globally coupled graph could also be mapped using more layers). There seems to be no technical impediment in building space variant CNN chips and we hope that this work can further motivate the development of hardwares in this direction.

Acknowledgments

This work was supported by a Romanian CNCSIS No. 1571 (contract 84/2007) and PNCDI2/ElBioArch research grant and also a Hungarian ONR grant (N00014-07-1-0350). Illuminating discussions with Dr. Ákos Zarándy are gratefully acknowledged.

References

[1] G.E. Moore, Electronics 38 (1965) 114.
[2] T. Roska, Computational and computer complexity of analogic cellular wave computers, in: Proceedings of the 7th IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA 2002, Frankfurt, Germany, 2002, pp. 323–335.
[3] T. Roska, Circuits, International Journal of Circuit Theory and Applications 35 (5–6) (2007) 485.
[4] L.O. Chua, L. Yang, IEEE Transactions on Circuits and Systems 35 (1988) 1257.
[5] T. Roska, L.O. Chua, IEEE Transactions on Circuits and Systems 40 (1993) 163.
[6] www.anafocus.com.
[7] T. Roska, L.O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, F. Puffer, IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications 42 (10) (1995) 807.
[8] T. Kozek, L.O. Chua, T. Roska, D. Wolf, R. Tetzlaff, F. Puffer, K. Lotz, IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications 42 (10) (1995) 816.
[9] J.M. Cruz, L.O. Chua, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 42 (10) (1995) 715.
[10] K.R. Crounse, T. Yang, L.O. Chua, Pseudo-random sequence generation using the CNN universal machine, in: Fourth IEEE International Workshop on Cellular Neural Networks and their Applications, Seville, Spain, 1996.
[11] K.R. Crounse, L.O. Chua, IEEE Transactions on Circuits and Systems 42 (10) (1995) 583.
[12] M. Ercsey-Ravasz, T. Roska, Z. Néda, International Journal of Modern Physics C 17 (6) (2006) 909.
[13] M. Ercsey-Ravasz, T. Roska, Z. Néda, European Physical Journal B 51 (2006) 407.
[14] L.O. Chua, T. Roska, Cellular Neural Networks and Visual Computing, Foundations and Applications, Cambridge University Press, 2002.
[15] F. Barahona, Journal of Physics A-Mathematical General 15 (1982) 3241.
[16] S. Istrail, Universality of intractability of the partition functions of the Ising model across non-planar lattices, in: Proceedings of the Symposium on the Theory of Computation STOC, Portland Oregon, USA, 2000, pp. 87–96.
[17] T. Roska, A. Rodriguez-Vasquez, Proceedings of the IEEE 90 (2000) 1244.
[18] J.J. Hopfield, Proceedings of the National Academy of Sciences USA 81 (1984) 3088.
[19] S.F. Edwards, P.W. Anderson, Journal of Physics F: Metal Physics 5 (1975) 965.
[20] D. Sherrington, S. Kirkpatrick, Physical Review Letters 35 (26) (1975) 1792.
[21] S. Kirckpatrick, C.D. Gelatt, M.P. Vecchi, Science 220 (4598) (1983) 671.
[22] M. Mezard, G. Parisi, M.A. Virasoro, Spin Glass Theory and Beyond, World Scientific, Singapore, 1987.
[23] H. Nishimori, Statistical Physics of Spin Glasses and Information Processing. An Introduction, Clarendon Press, Oxford, 2001.
[24] G. Rowe, Theoretical Models in Biology: The Origin of Life, the Immune System and the Brain, New edition, Clarendon Press, Oxford, 1997.
[25] R.N. Mantegna, H.E. Stanley, An Introduction to Econophysics - Correlations and Complexity in Finance, Cambridge University Press, Cambridge, England, 2000.
[26] N. Sourlas, Nature 339 (1989) 693.