A continuous-time approach to constraint satisfaction: Optimization hardness as transient chaos

PN-II-RU-TE-2011-3-0121
Final Sintetic Report

General aims and objectives of the project

Constraint satisfaction problems (such as Boolean satisfiability) constitute one of the hardest classes of optimization problems with many applications in technology and industry. Understanding why these problems are hard is crucially important for the development of efficient algorithms. Before starting this project we provided an exact mapping of Boolean satisfiability (k-SAT) into a deterministic continuous-time dynamical system (CTDS) with a unique correspondence between its set of attractors and the k-SAT solutions [1]. We have shown that optimization hardness is fundamentally equivalent to the phenomenon of chaos and turbulence: after a critical constraint density is reached, the trajectories become transiently chaotic before finding the solutions, signaling the appearance of optimization hardness. In this project we proposed to explore this revolutionary new connection between traditionally separate fields.

The research proposed had two main parts with several objectives:

Aim I: Understanding optimization hardness by studying chaotic properties of the CTDS.

Aim II: Developing a Cellular Nonlinear Network (CNN) model for solving satisfiability problems.

In Aim I we proposed to analyze the chaotic properties of the CTDS presented in [1] trying to reveal why certain problems are hard for all known algorithms. Our goal was to investigate three major questions: A) The chaotic phase transition appearing in NP-complete problems. B) the chaotic behavior of locked occupation problems, these being considered the hardest class of constraint satisfaction problems. C) Distinguishing hard-SAT formulas characterized by long chaotic transients from permanently chaotic UNSAT formulas.

In Aim II we proposed A) to develop a CNN model for solving SAT problems and B) study the effect of noise on these analog algorithms (both the original and the new one), since noise is the main concern in the implementation of all analog devices.

Objectives achieved

The Gantt chart below shows the evolvement of our project. Even if the order of solving the proposed tasks has changed, all aims and objectives were successfully achieved.

The first results came related to Aim IB. We studied the properties of Sudoku, which is a locked occupation problem, but being well known to the general public we used it as an example to present our more general results. We developed a method to measure the hardness of individual problems. This result was published in the open access journal of Nature Publishing Group, the **Scientific Reports** [2] and it generated a large echo in the media (see dissemination of results at the end of the report). Using the hardness measure developed we were able to achieve Aim IA. We proved the existence of a chaotic phase transition in SAT and we studied its properties in details. A paper about these results is under preparation. In the last year of the project we worked mainly on Aim IC. We developed an algorithm for solving the max-SAT problem, which practically means identifying the minimum energy level in unsatisfiable SAT instances. These results were presented at the CNNA conference in 2014, Notre Dame, USA.

In parallel we have also been working on Aim IIA since the beginning of the project. We developed the CNN model, it was published and presented on the **CNNA conference** in Turin 2012 [3]. In 2013

we also started to work on Aim IIB, studying the effects of noise on these analog SAT solver systems. These results have been finalized and published in Europhysics Letters in 2014.

Below we will summarize these results and will also mention some extra research activities and collaborations with foreign researchers, which have not been originally planned, but resulted in excellent publications such as a paper in **Physical Review E**, **Nature Communications**, **PLoS One**, and most importantly a **Science** paper.

At the end we will list all dissemination activities, such as publications, conferences, presentations etc. In total the project resulted in 6 ISI publications, one book chapter, 10 conference participations (4 articles and 6 oral presentations), invited talks at foreign universities and many echoes in the media.

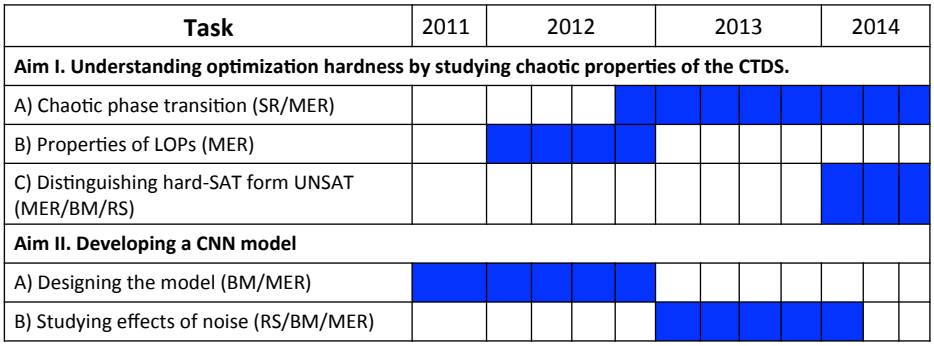

Figure 1: Gantt chart of the project evolution. Initials indicate the team members participating in each part of the project. MER: Dr. Mária Ercsey-Ravasz, project director. RS: Dr. Robert Sumi, postdoctoral researcher. BM: Botond Molnár, PhD student.

Summary of Results

Constraint satisfaction problems (CSP) arise in many domains of computer science, information theory and statistical physics. In these problems we are given a set of constraints and we are asked to find an assignment of the variables which satisfies all constraints. One of the most studied CSPs is Boolean satisfiability, especially the random k-SAT problem, which involves N Boolean variables ($x_i \in \{0,1\}, i = 1,\ldots,N$) and $M = \alpha N$ constraints or clauses. A k-clause is an OR operation ($\vee$) between k randomly chosen literals, meaning it is satisfied if and only if at least one of the literals is true. A literal can be the normal or negated form (marked with overline) of a Boolean variable. A different phrasing could be that a k-clause forbids one random assignment (out of the 2^k possibilities) of the k randomly included variables. For example $x_1 \vee x_3 \vee \overline{x}_9$ forbids ($x_1 = 0, x_3 = 0, x_9 = 1$). k-SAT (for $k \geq 3$) was the first CSP proved to be NP-complete [5, 6] and has major applications in artificial intelligence, electronic design, automation, error-correction, bio-computational applications etc. Random k-SAT has become a standard way of testing algorithmic performance and evaluating complexity and hardness of NP-complete problems.

For solving k-SAT with continuous-time dynamical systems [1] we reformulate the problem on the space of N continuous variables s_i ($i = 1,\ldots,N$)) which can take values in the interval $s_i \in [-1, 1]$. When defining correspondence with the discrete SAT problem, $s_i = -1$ would correspond to the Boolean variable being false and $s_i = +1$ to the variable being true. Similarly to the transformation of SAT to spin-glass models, here we also define an energy function which has to be minimized, its ground state ($E(\boldsymbol{s}^*) = 0$) corresponding to the solution $\boldsymbol{s}^*$ of the SAT instance. We need each constraint to be characterized by a cost function which is 0 if and only if the Boolean versions of these real variables would satisfy the clause. The OR operation which appears in each clause can be mapped into a multiplication in the cost function.

For a clause like $x_1 \vee x_3 \vee \overline{x}_9$ to be satisfied it is enough that one of the literals to be true, so in the cost function each literal will be a factor which is zero if and only if the literal is true. For this clause this function would look like $K(\boldsymbol{s}) = (1-s_1)(1-s_3)(1+s_9)/2^3$, where we introduce the 2^3 factor to normalize the cost function to the interval $K(\boldsymbol{s}) \in [0,1]$. The energy function of the whole SAT instance will be defined as the sum of the squares of cost functions for all M clauses: $E(\boldsymbol{s}) = \sum_{m=1}^{M} K_m(\boldsymbol{s})^2$. Searching for the minimum through continuous-time dynamical systems would naturally call for a gradient descent dynamics. However, this energy function has many local minima and a direct minimization will instantly drive the dynamics in one if these local wells, non-solution attractors, where the energy is not zero.

When designing the dynamics our goal was to achieve the following properties [1]: 1) Deterministic dynamics. 2) The dynamics should stay confined in space $\boldsymbol{s} \in [-1,1]^N$. 3) Every solution should be an attractive fixed point. 4) The only fixed points of satisfiable formulae should be the solutions. 5) Limit cycles should not exist. We could achieve all these by introducing auxiliary variables with a similar role as Lagrange multipliers. Each constraint is associated with an auxiliary variable $a_m \in (0,\infty)$ $(m = 1,\ldots,M)$, a weight which characterizes its importance at a given moment. The new energy function will be defined as: $V(\boldsymbol{s},\boldsymbol{a}) = \sum_{m=1}^{M} a_m K_m(\boldsymbol{s})^2$. We can see that $V(\boldsymbol{s},\boldsymbol{a}) \geq 0$ and $V(\boldsymbol{s^*}) = 0$ if and only if $\boldsymbol{s^*}$ is a solution.

The system of ordinary differential equations (ODEs) which achieves all the desired properties is the following:

$$\frac{ds_i}{dt} = -\frac{\partial}{\partial s_i} V(\boldsymbol{a},\boldsymbol{s}) \tag{1}$$

$$\frac{da_m}{dt} = a_m K_m(\boldsymbol{s}), \tag{2}$$

and we must have $s_i(0) \in [-1,1], \forall i$ and $a_m(0) > 0, \forall m$. The dynamics of the $\boldsymbol{s}$ variables is now defined as a gradient descent on the $V(\boldsymbol{s},\boldsymbol{a})$ energy landscape and the auxiliary variables show an exponential increase whenever the given constraint is not satisfied ($K_m(\boldsymbol{s}) > 0$). The auxiliary variables of the larger $K_m(\boldsymbol{s})$ values grow the fastest, they rapidly dominate the $V(\boldsymbol{s},\boldsymbol{a})$ energy and the dynamics of $\boldsymbol{s}$ moves in a direction reducing the $K_m(\boldsymbol{s})$ cost function of these clauses. While each solution is a stable fixed point of the system, the attractors are in fact much larger subspaces corresponding to the solution clusters (for a more detailed explanation and illustration see the supplementary information of [1]).

Aim I. B) Measuring hardness of constraint satisfaction problems

Because of the evolvement of the research we first were able to achieve Aim IB, and only consequently Aim IA. We present the results in this order.

The mathematical structure of Sudoku puzzles is akin to typical hard constraint satisfaction problems lying at the basis of many applications, including protein folding and the ground-state problem of glassy spin systems. Via an exact mapping of Sudoku into a deterministic, continuous-time dynamical system, we have shown that the difficulty of Sudoku translates into transient chaotic behavior exhibited by the dynamical system. We have also shown that the escape rate κ, an invariant of transient chaos, provides a scalar measure of the puzzle's hardness that correlates well with human difficulty ratings. Accordingly, $\eta = -\log_{10}\kappa$ can be used to define a "Richter"-type scale for puzzle hardness, with easy puzzles having $0 < \eta \leq 1$, medium ones $1 < \eta \leq 2$, hard with $2 < \eta \leq 3$ and ultra-hard with $\eta > 3$. To our best knowledge, there are no known puzzles with $\eta > 4$.

Sudoku as k-SAT. Because our continuous-time dynamical system [1] was designed to solve k-SAT formulae in conjunctive normal form (CNF), we first briefly describe how Sudoku can be interpreted as a +1-in-9-SAT formula, and then how it is transformed into the standard CNF form.

In a Sudoku puzzle we are given a square grid with $9 \times 9 = 81$ cells, each to be filled with one of nine symbols (digits) $D_{ij} \in \{1,\ldots,9\}$, $i,j = 1,\ldots,9$ (with the upper-left corner of the puzzle corresponding

to $i = 1$, $j = 1$). When the puzzle is completed each of the columns, rows and 3×3 sub-grids (blocks partitioned by bold lines, Fig. 2a) must contain all the 9 symbols. Equivalently, all 9 symbols must appear once and only once in each row, column and 3×3 sub-grid.

To formulate Sudoku as a constraint satisfaction problem (CSP) using boolean variables, we associate to each symbol (digit) an ordered set of 9 boolean variables (TRUE="1", FALSE="0"). The digit D_{ij} in cell (i, j) will be represented as the ordered set $(x_{ij}^1, \ldots, x_{ij}^9)$ with $x_{ij}^a \in \{0, 1\}$, $a = 1, \ldots, 9$, such that always one and only one of them is 1 (TRUE). Thus $D_{ij} = a$ is equivalent to writing $x_{ij}^b = \delta_{a,b}$, where $\delta_{a,b}$ is the Kronecker delta function. This way we have in total $9 \times 9 \times 9 = 729$ boolean variables x_{ij}^a, which we can picture as being placed on a 3D grid (Fig. 2b), with a corresponding to the grid index along the vertical direction, and hence a is the digit that is filling the corresponding (i, j) cell in the original puzzle. The corresponding horizontal 9×9 2D layer at height a will be denoted by L_a. Introducing the notion of such (horizontal) layers makes it easier to express the constraints of the Sudoku rules on its representation by 0-s and 1-s as described below. For example in the puzzle shown in Fig. 2a, we have $D_{1,9} = 4$. In the given vertical column the variable in the a^{th} cell (that is in layer L_a) is $x_{1,9}^a = \delta_{a,4}$ (shown as the boolean variable 1 filling the cell next to the digit 4 shown in red, in Fig 2b). This setup allows us to encode the Sudoku constraints in a simple manner. They come from: 1) uniqueness of the digits in all the (i, j) Sudoku cells, 2) a digit must occur once and only once in each row, column and in each of the nine 3×3 subgrids, and 3) obeying the clues. Constraint type 1) was already expressed above, namely that for every cell (i, j), in the set $(x_{ij}^1, \ldots, x_{ij}^9)$ one and only one variable is TRUE, all others must be FALSE. Type 2) constraints are similar, e.g., in row i and layer L_a the set $(x_{i1}^a, \ldots, x_{i9}^a)$ must contain one and only one TRUE variable, all others must be false and this must hold for all rows and layers, etc. Observe that all constraints are in the form of a set of 9 boolean variables of which we demand that one and only one of them be TRUE, all others FALSE. When this is satisfied, we say that the constraint itself (or "clause") is satisfied, or TRUE. Such CSPs are called +1-in-k-SAT and they are part of so-called "locked occupation problems", which is a class of exceptionally hard CSPs [7,8]. Type 3) constraints are generated by the clues (or givens) which are symbols already filled in some of the cells and their number and positioning determines the difficulty of the puzzle. They are also set in a way to guarantee a unique solution to the whole puzzle. If there are given d clues, then this implies setting d boolean variables to TRUE, which means eliminating exactly $4d$ constraints of type 1) and 2) (one vertical or uniqueness constraint, one row, one column and one 3×3 subgrid constraint). Thus, Sudoku is a +1-in-9-SAT type CSP with N boolean variables and $324 - 4d$ constraints of +1-in-k-SAT type ($k \leq 9$). N is a complicated function of the positioning of the clues. The example in Fig. 2a has $d = 22$ clues with $N = 232$ unknown boolean variables. In layer L_4 as illustrated in Fig. 2c there are 28 unknown boolean variables (white cells). These 28 variables appear in a total of 17 constraints of +1-in-k-SAT type. More precisely there is one +1-in-2-SAT, six +1-in-4-SAT, four +1-in-5-SAT and six +1-in-6-SAT type of constraints related to L_4. The other layers generate the remaining $324 - 4d - 17 = 219$ of +1-in-k-SAT type constraints (with $k \leq 9$).

Since our continuous-time SAT solver has been designed to solve boolean satisfiability problems in conjunctive normal form (CNF), we need to bring the +1-in-k-SAT type problems above into this form, and thus formulate it as a k-SAT CNF problem. The CNF is a conjunction (AND, denoted by $\wedge$) of clauses each clause expressed as the disjunction (OR, denoted by $\vee$) of literals. For k-SAT in CNF there are N boolean variables $x_i = \{0, 1\}$ and an instance is given as a propositional formula $\mathcal{F}$, which is the conjunction of M clauses C_m: $\mathcal{F} = C_1 \wedge \cdots \wedge C_m \wedge \cdots \wedge C_M$, with $C_m = z_{m_1} \vee \ldots \vee z_{m_{k_m}}$, $k_m \leq k$ and $z_j \in \{x_j, \bar{x}_j\}$. According to a well known theorem of propositional calculus, all boolean propositions can be formulated in CNF using De Morgan's laws and the distributive law, and thus any +1-in-k-SAT type constraint as well (for details see [2]). Once the transformation to CNF is completed we are left with N variables and M clauses of the type described above, called CNF clauses from here on. In our case the number of variables appearing in a CNF constraint has the property $1 \leq k_m \leq 9$. The parameters N, M and $\{k_m\}_{m=1}^M$ depend on the clues that are difficult to express analytically, but easy

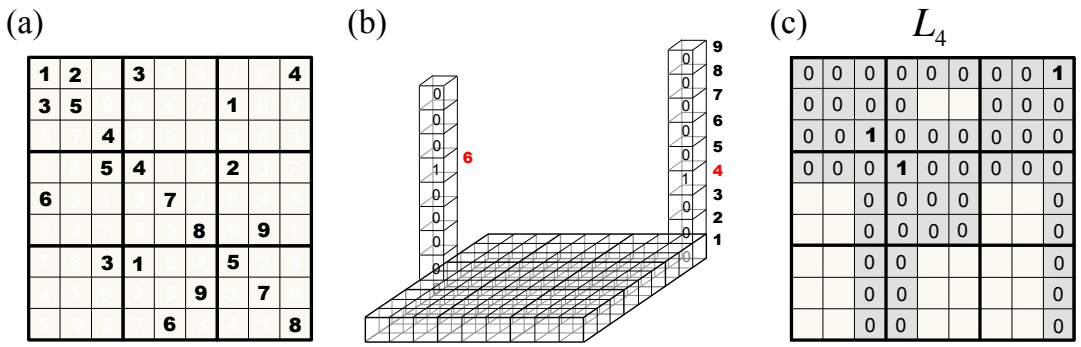

Figure 2: **Sudoku and its boolean representation.** (a) a typical puzzle with bold digits as clues (givens). (b) Setup of the boolean representation in a $9 \times 9 \times 9$ grid. (c) Layer L_4 of the puzzle (the one containing the digit 4) with 1-s in the location of the clues and the regions blocked out for digit 4 by the presence of the clues (shaded area).

to determine computationally. The puzzle from Fig. 2a is ultimately formulated as a CNF SAT problem with $N = 232$ variables and $M = 1718$ CNF clauses. An often used parameter of a satisfiability problem is the number of CNF constraints per variable, or *constraint density*, $\alpha = M/N$, also used as a typical hardness indicator, however, as we show below, this is not an accurate measure of hardness.

Puzzle hardness as transient chaotic dynamics. Since Sudoku puzzles always have a solution, the corresponding boolean SAT CNF formulation also has a solution, and system (1-2) will always find it. The nature of the dynamics, however, will depend on the hardness of the puzzle as we describe next.

In Fig.3a we show an easy puzzle with 34 clues (black numbers). After transforming this problem into SAT CNF, we obtain $N = 126$ and $M = 717$, with a constraint density of $\alpha = M/N = 5.69$. As described above, in our implementation there is a continuous spin variable s_{ij}^a associated to every boolean variable x_{ij}^a in every 3D cell (i, j, a). In the right panels of Fig. 3 we show the dynamics of the spin variables in the cells of the 3×3 grid formed by rows 4-6 and columns 7-9. The $s_{ij}^a(t)$ curves are colored by the digit a they represent ($a = 1, \ldots, 9$) as indicated in the color legend of Fig 3. The dynamics was started from a random initial condition. Indeed, our solver finds the solution very quickly, in about 15 time units, for the easy puzzle in Fig.3a.

In Fig. 3b we show the dynamical evolution of variables for an ultra-hard Sudoku instance with only 21 clues. This puzzle has been listed as one of the world's hardest Sudokus, and even has a special name: "Platinum Blonde" [9–11], and it was the most "difficult" for our solver among all the puzzles we tried. After transforming it into SAT CNF, we obtain $N = 257$ variables and $M = 2085$ constraints. Not only that we have twice as many unknown variables but the constraint density $\alpha = M/N = 8.11$ is also larger than in the previous case, signaling the hardness of the corresponding SAT instance. The complexity of the dynamics in this case is seen in the right panel of Fig. 3b, exhibiting long chaotic transients before the solution is found at around $t \simeq 150$. For an animation of the dynamics for a similarly hard puzzle [12] see Ref [13].

A Richter-type scale for Sudoku hardness. As suggested by the two examples in Fig 3, the hardness of Sudoku puzzles correlates with the length of chaotic transients. A consistent way to characterize these chaotic transients is to plot the distribution of their lifetime. Starting trajectories from many random initial conditions, let $p(t)$ indicate the probability that the dynamics has not found the solution by time t. A characteristic property of transient chaos [14, 15] in hyperbolic dynamical systems is that $p(t)$ shows an asymptotic exponential decay: $p(t) \sim e^{-\kappa t}$, where κ is called the escape rate. The escape rate is an invariant measure of the dynamics in the sense that it characterizes solely the chaotic non-attracting set in the phase space of the system, and it does not depend on the distribution of the

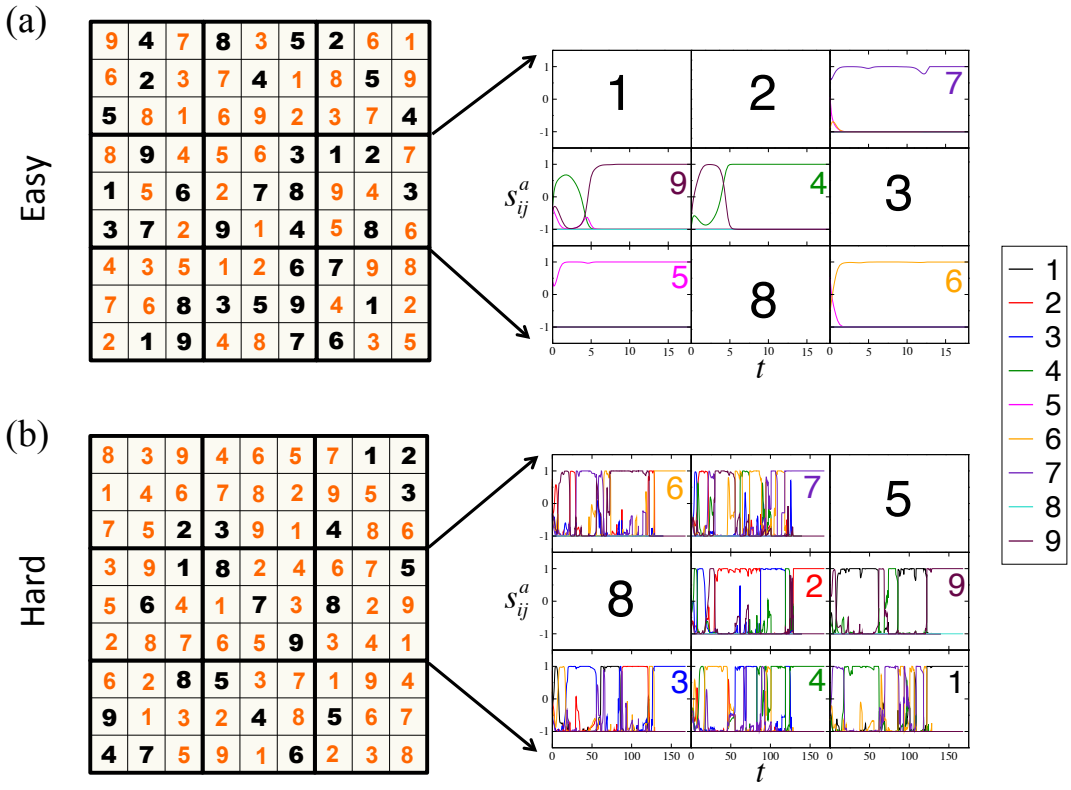

Figure 3: **Solving Sudoku puzzles with the deterministic continuous-time solver (1-2)**. (a) presents an easy puzzle with the evolution of the continuous-time dynamics shown within a 3×3 grid (rows 4-6, columns 7-9). (b) shows the same, but for a known, ultra-hard puzzle called "Platinum Blonde" [9–11]. The trajectories in the right panels show the evolution of the analog variables $s_{ij}^a(t)$ colored by the corresponding digit a. Thus for each cell (i, j) we have 9 such running trajectories, but they cannot always be discerned as many of them are running on top of one another, close to zero as in (a).

initial conditions, its support, or the details of the region from where the escape is measured (as long as it contains the non-attracting set) [14].

In Fig. 4a we plot the distribution $p(t)$ in log-linear scale for several puzzles gathered form the literature. The distributions were obtained from over 10^4 random initial conditions. The decay shows a wide range of variation between the puzzles. For easy puzzles the transients are very short, $p(t)$ decays fast resulting in large escape rates but for hard puzzles κ can be very small. Fig. 4b shows a zoom onto the $p(t)$ of hard puzzles. For example, for the puzzle in Fig.3a we obtain $\kappa = 0.156$, whereas for Fig.3b (Platinum Blonde) the escape rate is $\kappa = 0.00026$. In spite of the large variability of the decay rates, we see that in all cases the escape is exponentially fast.

The several orders of magnitude variability of κ naturally suggests the use of a logarithmic measure of κ for puzzle hardness, see Fig.4c, which shows the escape rates on a semilog scale as function of the number of clues, d. Thus, the escape rate can be used to define a kind of "Richter"-type scale for Sudoku hardness:

$$\eta = -\log_{10}(\kappa) \tag{3}$$

with easy puzzles falling in the range $0 < \eta \leq 1$, medium ones in $1 < \eta \leq 2$, hard ones in $2 < \eta \leq 3$ and for ultra-hard puzzles $\eta > 3$. We chose several instances from the "Sudoku of the Day" website [16] in four of the categories defined there: easy (black square), medium (red circle), hard (green x) and absurd (blue star). These ratings on the website try to estimate the hardness of puzzles when solved by humans. These ratings correlate very well with our hardness measure η. Occasionally, daily newspapers present puzzles claimed to be the hardest Sudoku puzzles of the year. In particular, the escape rate for the Caveman Circus 2009 winner [17] (turquoise diamond) and the Guardian 2010 hardest puzzle [18] (maroon diamond) are indeed one order of magnitude smaller than the hardest puzzles on the daily Sudoku websites, placing them at $\eta = 2.93$ and $\eta = 2.82$ on the hardness scale. The USA Today 2006 hardest puzzle [19], however, does not seem to be that hard for our algorithm having $\eta = 2.17$ (magenta diamond). Eppstein [20] gives two Sudoku examples (orange left-pointing triangles) while describing his algorithm, one with $\eta = 1.288$ and a much harder one with $\eta = 2.017$. Elser *et al.* [12] present an extremely hard Sudoku (black filled circle), which has an escape rate of $\kappa = 0.0023$ resulting in $\eta = 2.639$. The smallest escape rates we have found are for the Sudokus listed as the hardest on Wikipedia [10, 11] (red triangles).

While the escape rate correlates surprisingly well with human ratings of Sudoku hardness, it is natural to expect a correlation with the number of clues, d. Indeed, as a general rule of thumb, the fewer clues are given, the harder the puzzle, however, this is not universally true [21]. In Fig.4d we then plot the escape rate as function of the constraint density $\alpha = M/N$, leading to practically the same conclusion. This is because the constraint density α is essentially linearly correlated with the number of givens d, as shown in Fig.4e. The apparent non-monotonic behavior of puzzle hardness with the number of givens, (or constraint density) is due to the fact that hardness cannot simply be characterized by a global, static variable such as d or α, but it also depends on the *positioning* pattern of the clues, as also shown by concrete examples in Ref [21].

Using the world of Sudoku puzzles, we have presented further evidence that optimization hardness translates into complex dynamical behavior by an algorithm searching for solutions in an optimal fashion. Namely, there seems to be a trade-off between algorithmic performance and the complexity of the algorithm and/or its behavior [2]. Simple, sequential search algorithms have a trivial description and simple dynamics, but an abysmal worst-case performance (2^N), whereas algorithms that are among the best performers are complex in their description (instruction-list) and/or behavior (dynamics). This happens because in order to improve performance, algorithms have to exploit the structure of the problem one way or another. As hard problems have complex structures, the dynamics of the algorithms should be indicative of the problem's hardness. The continuous-time dynamical system [1] (1-2) as a *deterministic algorithm* does have these features: 1) the search happens on an energy landscape $V = \sum_m a_m K_m^2$ that incorporates simultaneously *all* the constraints (problem structure) 2) it solves easy problems efficiently

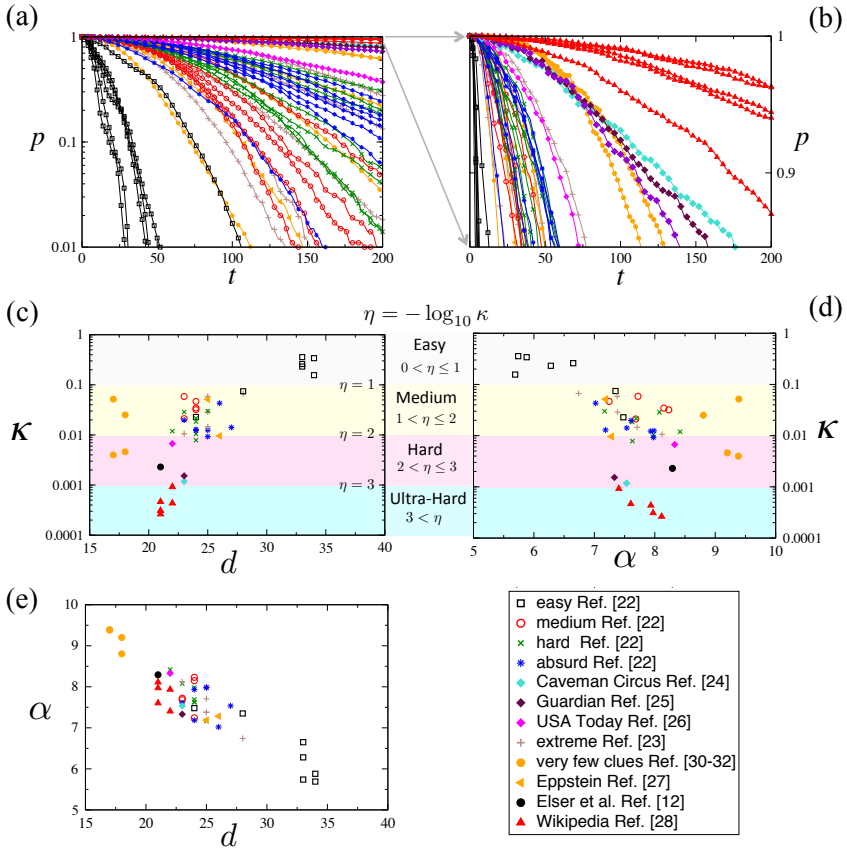

Figure 4: **Escape rate as hardness indicator.** (a) shows the distribution in log-linear scale of the fraction $p(t)$ of 10^4 randomly started trajectories of (1-2) that have not yet found a solution by analog time t for a number of Sudoku puzzles taken from the literature (see legend and text) with a wide range of human difficulty ratings. The escape rate is obtained from the best fit to the tail of the distributions. (b) is a magnification of (a) for hard puzzles. (c) and (d) show the escape rate κ in semilog scale vs the number of clues d and constraint density α indicating good correlations with human ratings (color bands). (e) shows the relationship between the number of clues d and α for the puzzles considered.

(polynomial time, both analog and discrete) and 3) it guarantees to find solutions to hard problems even for solvable cases where many other algorithms fail. Although it is not a polynomial cost algorithm, it seems to find solutions in continuous-time t that scales polynomially with N [1]. These features and the fact that the algorithm is formulated as a deterministic dynamical system with continuous variables, allows us to apply the theory of nonlinear dynamical systems on CTDS (1-2) to characterize the hardness of boolean satisfiability problems. In particular, via the measurable escape rate κ, or its negative log-value η, we can provide a single-scalar measure of hardness, well defined for any *finite instance*. We have illustrated this here on Sudoku puzzles, but the analysis can be repeated on any other ensemble from NP. Having a mathematically well-defined number to characterize optimization hardness for specific problems in NP provides more information than the polynomial/exponential-time solvability classification, or knowing what the constraint density $\alpha = M/N$ is (the latter being a non-dynamic/static measure). Moreover, within the framework of CTDS (1-2), dynamical systems and chaos theory methods can now be brought forth to help develop a novel understanding of optimization hardness.

These results have been published in Nature Scientific Reports [2] and generated a large echo in the scientific community and even the media. In the first week after publication 50,000 people accessed our paper, and several journals wrote news article about our results, such as Daily Mail, Huffington Post etc.

Aim I. A) Chaotic phase transition

Aim IA has now been completed, but results are still under publication. For this reason the summary of results is not included in this online version of our report.

Aim I. C) Distinguishing hard-SAT problems with long chaotic transients from unsatisfiable instances

This Aim is practically equivalent with solving the max-SAT problem, which asks one to find the maximum possible number of satisfiable constraints in a given SAT instance. This means that even in unsatisfiable (UNSAT) instances we have to find the best solution, so we need to be able to identify UNSAT instances and distinguish them from satisfiable but hard instances.

We again used the same dynamics as presented and discussed above. We have shown, that even if trajectories never get trapped, the most efficient strategy is not to start one single trajectory and wait until it finds a solution, but to start more trajectories and stop them after a relatively small amount of time. To show this, here we will use a new discrete "energy function" of the system which describes the exact number of total constraints remained unsolved after running the simulation for time t ($t \in [0, t_{max}]$). Taking a max-SAT problem and starting N_{init} trajectories from random initial conditions, one can measure for each time t and discrete energy E_d the number of trajectories $p(t, E_d)$, which have found a state with energy E_d or smaller up to time t. Then we define the average time-cost of finding an energy level E_d by running trajectories up to time t as $k(t, E_d) = t \cdot N_{init}/p(t, E_d)$ (Fig. 5). We can see there is an optimal t value for each energy level (red curve). To efficiently solve max-SAT problems a small learning algorithm with the following main steps was developed: 1) running the dynamics from different initial conditions with small t_{max}; 2) recording which energy levels were reached in every time step, before reaching t_{max} ($p(t, E_d)$); 3) using the formula $k(t, E_d) = t \cdot N_{init}/p(t, E_d)$, the cost of reaching energy level E_d before time t is calculated (Fig. 5); 4) the minimum cost for every energy level is selected; 5) using the Levenberg-Marquardt non-linear curve fitting method the energy curve as a function of the costs is fitted in real-time (Fig. 6) to approximate the curves parameters; 6) these steps are repeated until a good approximation is obtained, increasing the number of instances if needed.

On Fig. 5 the mean cost (blue color) of reaching the energy level E_d by running N_{init} trajectories up to time t is represented. Until now relatively small instances have been considered with $N = 70$ variables and $M = 800$ constraints. After selecting the lowest cost for every energy level (red line in Fig. 5) one can

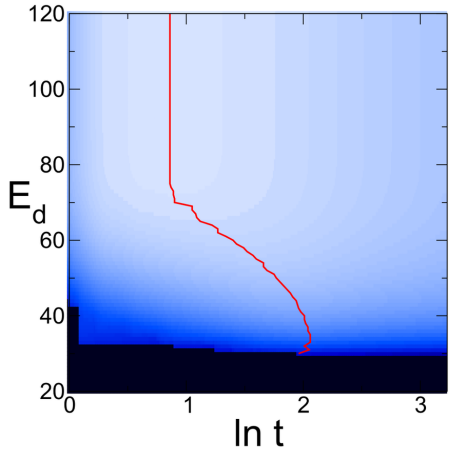

Figure 5: The average cost $k(t, E_d)$ (blue colour scale) of reaching energy E_d by running trajectories up to time t in a *max*-SAT problem with $N = 70$ literals and $M = 800$ constraints. Darker colour represents increased cost, black colour corresponding to infinitely large costs (there is no state with that low energy). The red curve shows the optimal time limit until trajectories should run to reach a given energy.

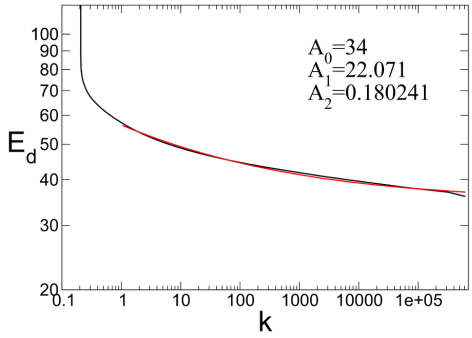

Figure 6: The reached energy levels as a function of the necessary (optimal) cost is represented of a *max*-SAT system with $N = 70$ literals and $M = 800$ constraints with the red line representing the Levenberg-Marquardt fit. The parameters of the non-linear curve fitting gives us the expected energy minimum and the hardness of the problem.

see that there is no need for running the simulations longer than $t_{max} = 25$, but increasing the statistics lower energy levels become reachable.

The energy level as a function of the cost of reaching that level has a power-law decay, which converges to a constant digit, described by: $f(x) = A_0 + A_1 \cdot x^{-A_2}$. The Levenberg-Marquardt non-linear curve fitting algorithm was used in order to calculate the parameters of the power-law decay. We observed that the predicted lowest possible energy state of the system is represented by the constant to which our system is converging (A_0) and the exponent (A_2) of the power-law can be considered as a hardness of the problem, providing a potential tool for classification of NP-hard problems according to their difficulty. In order to verify our findings the results were compared to the ones obtained by the *maxsatz* algorithm, the winner of the 8^{th} International *max*-SAT Solving Competition (2013). This algorithm is an exact solver. The comparison can be seen in the following table:

$problem$	$maxsatz$	$Our-prediction$	$Our-reached$
1	31	31	31
2	34	34	35
3	30	30	30
4	28	26	28
5	31	31	32
6	30	30	30
7	31	30	32
8	34	34	35
9	32	30	33
10	30	31	32

These were 10 benchmark *max*-SAT instances were used and ran from around $1.0-2.5M$ separate initial conditions, depending on the hardness of the problem.

We developed a method based on the analog dynamical system presented in [1] to approximate the global minima of NP-hard *max*-SAT problems. Results show that the optimal strategy is to run many trajectories up to a relatively small time. By extrapolating the parameters of the power-law decay of the reached energy levels as function of the needed cost we can approximate the global minimum of the system and the hardness of the problem. Even if this approximation method requires fine tuning, the results are in good agreement with the ones obtained using the *maxsatz* exact solver. We presented and published these results at the CNNA 2014 conference. We are still working on refining the method and our future goal is to apply it also on the CNN SAT-solver presented in [3, 4].

Aim II. A) CNN model for k-SAT

One of the main back draws of our analog system is that it cannot be implemented by analog circuits (the auxiliary variables are unbounded). For that reason in Aim IIA we proposed to develop a deterministic continuous-time recurrent neural network similar to CNN models, which can solve Boolean satisfiability (k-SAT) problems without getting trapped in non-solution fixed points. The model can be implemented by analog circuits, in which case the algorithm would take a single operation: the template (connection weights) is set by the k-SAT instance and starting from any initial condition the system converges to a solution. We proved that there is a one-to-one correspondence between the stable fixed points of the model and the k-SAT solutions and present numerical evidence that limit cycles may also be avoided by appropriately choosing the parameters of the model [3].

The continuous-time recurrent neural network model for solving k-SAT with the following key properties:

- It has a deterministic continuos-time dynamics.
- All variables remain bounded.
- The dynamics can be implemented with analog circuits (has almost the same form as used in CNN computers).
- There is a one-to-one correspondence between the stable fixed-points of the system and the solutions of the k-SAT problem.
- On an analog device this would be a one-operation (one single template) algorithm for solving k-SAT: the connection weights (template) are defined by the k-SAT instance and starting from any arbitrary initial condition the system converges to a solution without getting trapped.

The model. The simplest form of dynamics used in CTRNNs is:

$$\frac{dx_i(t)}{dt} = -x_i(t) + \sum_j w_{ij} f(x_j(t)) + u_i \tag{4}$$

where x_i is the state value, or activation potential of the cell, $f(x)$ is the output function of the neuron (usually a sigmoid), u_i is the input, or bias of the neuron and w_{ij} are connection weights. There are many variants used in the literature, where in some cases the input my be time dependent $u_i(t)$; or it is allowed for a cell (in CNNs) to be influenced by the inputs of neighboring cells, etc. Here we use this simple form.

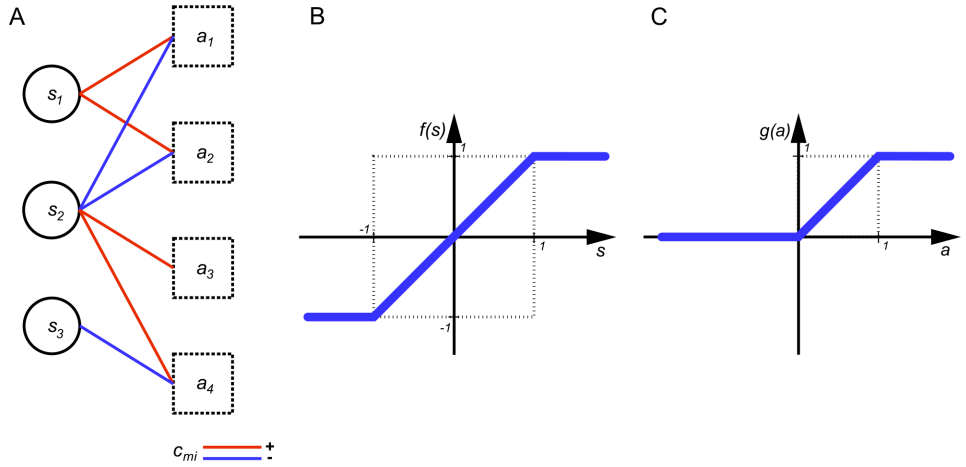

Figure 7: a) The bipartite graph representation of k-SAT: variables $\boldsymbol{s}$ represent the variables of the SAT formula, cells $\boldsymbol{a}$ correspond to the constraints. The sign of connections is given by the matrix c_{mi} encoding the k-SAT instance. b) The output function of s-type cells. c) the output function of a-type cells.

We define a CTRNN on a bipartite graph with two types of nodes (cells) (Fig.7a). One type ("s-type") represents the variables of k-SAT, whose state value will be denoted by s_i, $i = 1, \ldots, N$ and their output function defined via (see Fig.7b):

$$f(s_i) = \frac{1}{2}\left(|s_i + 1| - |s_i - 1|\right). \tag{5}$$

Here we assign to $f(s_i) = 1$ the Boolean variable x_i being true ($x_i = 1$) and to $f(s_i) = -1$ the variable being false ($x_i = 0$). However, in the dynamics we also allow any continuous value $f(s_i) \in [-1, 1]$. For simplicity we say that $f(\boldsymbol{s})$ is a solution of k-SAT, whenever $\boldsymbol{x} = [f(\boldsymbol{s}) + 1]/2$ is a solution. The self-coupling parameter will be a fixed value $w_{ii} = A$ and the input to these s-type cells is $u_i = 0\ \forall i$.

The second type of cells represent the clauses with state value a_m, $m = 1, \ldots, M$ and output function (Fig.7c):

$$g(a_m) = \frac{1}{2}(1 + |a_m| - |1 - a_m|). \tag{6}$$

These variables, or "a-type" cells will play a similar role as the Lagrange multipliers in [23–25] or the auxiliary variables in [1]. They determine the impact a clause has at a given moment on the dynamics of the $\boldsymbol{s}$ variables. For this reason $g(a_m) = 0$ will correspond to the clause being true, and $g(a_m) = 1$ to the clause being false. For these cells the self-coupling is $w_{mm} = B$ and the input is $u_m = u = 1 - k$ where k represents the number of variables in the clause ($k = 3$ for 3-SAT). This is needed in order to achieve the correspondence between k-SAT solutions and stable fixed points. The connection weights between the cells are determined by the c_{mi} matrix elements of the given k-SAT problem. The dynamical system

is defined via:

$$\dot{s}_i(t) = \frac{ds_i(t)}{dt} = -s_i(t) + Af(s_i(t)) + \sum_m c_{mi} g(a_m(t)) \quad (7)$$

$$\dot{a}_m(t) = \frac{da_m(t)}{dt} = -a_m(t) + Bg(a_m(t)) - \sum_i c_{mi} f(s_i(t)) + 1 - k \quad (8)$$

Properties of the CTRNN. We proved the following theorems [3]: *Theorem 1. Variables remain bounded: If initially* $|s_i(0)| \leq 1$ *and* $0 \leq a_m(0) \leq 1$*, then the state values of cells* $s_i(t)$ *and* $a_m(t)$ *remain bounded for all* $t > 0$*,* $\forall i, m$*. The following bounds are tight:*

$$|s_i(t)| \leq 1 + A + \sum_m |c_{mi}| \quad (9)$$

$$-2k \leq a_m(t) \leq 2 + B \quad (10)$$

Theorem 2. Every k-SAT solution has a corresponding stable fixed point: Given a k-SAT formula $\mathcal{F}$*, if* $f(s_i^*) = \pm 1$*,* $i = 1, \ldots, N$ *is a solution of* $\mathcal{F}$ *and* $A > 1$*,* $B > 1$ *then the* $(\boldsymbol{s^*}, \boldsymbol{a^*})$ *point:*

$$s_i^* = Af(s_i^*) \ , \ a_m^* = -\sum_j c_{mj} f(s_j^*) + 1 - k \quad (11)$$

$i = 1, \ldots, N$*,* $m = 1, \ldots, M$ *is a stable fixed point of the system (25-26).*
Theorem 3. A stable fixed point always corresponds to a solution. If $1 < A < 2$*,* $1 < B < 2$ *and* $(\boldsymbol{s^*}, \boldsymbol{a^*})$ *is a stable fixed point, then* $f(\boldsymbol{s^*})$ *must be a solution of the k-SAT formula.*

Numerical results. While we proved analytically that all stable fixed points of the system correspond to k-SAT solutions, it does not guarantee that there are no other attractors (such as limit cycles or chaotic attractors) in the system. The existence or non-existence of such attractors is very difficult to show analytically, but here we present numerical evidence, showing that by appropriately choosing the parameters A, B the dynamics avoids getting trapped in non-solution attractors and converges to a k-SAT solution.

We investigated in more depth how the efficiency of the system in finding solutions depends on the A, B parameters. In Fig. 8 we show two maps covering the $A \in (1, 2)$, $B \in (1, 3)$ parameter region, depicting the performance of the system. For each point of the maps we ran 100 different 3-SAT problems (we use only satisfiable instances) with $N = 20$ and $\alpha = 4.25$, this being in the hardest region for 3-SAT. As preparing such maps are computationally costly, we had to use small instances. On Fig8a the color gives the fraction of solved problems in the given time $t_{max} = 10000$ and on Fig8b we show the average continuous-time (not the simulation running time) the system takes to solve them (see color bars). There is a large parameter region where the solution is found efficiently. It is interesting that the optimal parameter setting seems to be around $A = 1.4$, $B = 2.24$ and it is fairly independent from the size of the system N (not shown). This suggests that Theorem 3 could be extended for larger values of $B > 2$, a statement that we will investigate in future studies.

Contrary to the system presented in [1] we expect an exponential time complexity, because the variables remain bounded and there is no extra energy introduced into the system. On Fig9a we can see the fraction $p(t)$ of problems which remain unsolved after a time t for various sizes ($N = 20, 30, \ldots, 100$) of randomly chosen 3-SAT instances at the optimal parameter values $A = 1.4$, $B = 2.24$. The distributions are decreasing as a power law ($p(t) \sim t^{-\beta(N)}$), where the power $\beta(N)$ depends on the size of the k-SAT instances. Because $\beta(N)$ is again a power law (very close to $\beta(N) \sim 1/N$) it can be shown that indeed, the time complexity of the model is exponential for solving a fixed fraction of problems. This power-law decrease of $p(t)$ shows that the probability of not finding the solution goes to zero (not a positive constant), supporting the claim that the dynamics does not get trapped in limit cycles.

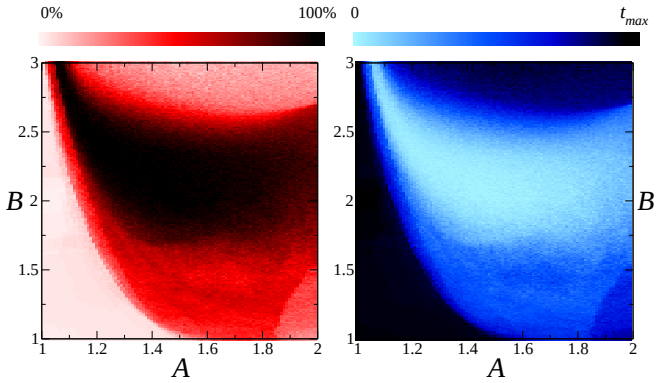

Figure 8: For each point $A \in (1,2), B \in (1,3)$ on the map we solve 100 randomly chosen satisfiable 3-SAT instances, with $N = 20$, $\alpha = 4.25$, $t_{max} = 10000$. a) The fraction of solved problems, b) the average continuos-time (see color bars).

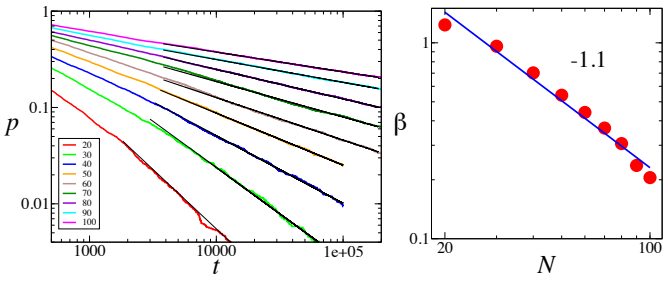

Figure 9: The number of 3-SAT problems $p(t)$ which remain unsolved as function of the continuous-time t of the system, for different values of N (see the legend). The distributions are fitted with a power law $p(t) \sim t^{-\beta(N)}$, and the right panel shows the dependence of the exponent β on N. This can be fitted with a power law $\beta(N) \sim N^{-1.1}$.

Discussion. We presented an implementation friendly dynamical system similar to CNN models, which solves Boolean satisfiability problems by converging to a stable fixed point of the dynamical system. On an analog device this algorithm would take a single operation: the connection weights (template) are based on the c_{mi} matrix corresponding to the given k-SAT instance and starting from any initial condition the system converges to a solution, without the need of any further intervention by the user. We have shown numerically that there is a large and robust parameter region where limit cycles do not appear, and the solution is always found even if the dynamics goes through a transiently chaotic phase. Although this CTDS is very different from the one presented in [1], this model also shows chaotic behavior, especially in the hard-SAT phase. This confirms the qualitative equivalence between optimization hardness and chaotic behavior exhibited by analog search algorithms.

The result has been published in the conference proceedings of the Cellular Nanoscale Networks and Applications Conference in 2012, Turin [3]. It generated a great interest, the project director has been later invited to the Péter Pázmany Catholic University in Budapest to give a talk to the engineers, in February 2013. Later in 2013 we also published a more detailed study in journal PLoS One.

Aim II. B) Robustness to noise of the analog SAT solvers

When implementing these analog SAT solvers with analog circuits, the great question comes if these will be robust to noise. Noise is characteristic of all electronic circuits, and is caused by small fluctuations of current and voltage. In electronic circuits different noise types can be identified: shot, thermal, flicker, burst, avalanche noise etc. [43]. The effect of different noise sources is introduced by adding a random term on the right hand side of equations 1 and 2, transforming them into stochastic differentail equations (SDEs). Simulating SDEs needs completely new techniques, it is part of a relatively new evolving field, which we needed to get familiar with.

We considered two type of noise: white and colored noise. White noise random variable is defined as the limit:

$$\xi(t) = \lim_{dt \to 0} N(0, 1/dt), \tag{12}$$

where $N(0, 1/dt)$ denotes a normal random variable with mean 0 and variance $1/dt$ [44]. In order to have the noise intensity equal to D we multiply $\xi(t)$ by $\sqrt{2D}$. Equations 1 and 2 become:

$$\dot{s_i} = -\frac{\partial}{\partial s_i} V(\boldsymbol{a}, \boldsymbol{s}) + \sqrt{2D}\xi_i(t), \qquad i = 1, \ldots, N \tag{13}$$

$$\dot{a_m} = a_m K_m + \sqrt{2D}\xi_m(t), \qquad m = 1, \ldots, M \tag{14}$$

White noise has the following statistical properties:

$$\langle \xi(t) \rangle = 0, \quad \langle \xi(t)\xi(t') \rangle = \delta(t - t'), \tag{15}$$

where δ denotes the Dirac-delta function.

In contrast with white noise, which is uncorrellated in time, colored noise is a correlated process. It can be obtained by solving the Langevin equation [46–49]:

$$\dot{\epsilon(t)} = -\gamma\epsilon(t) + \sqrt{2D}\xi(t), \tag{16}$$

The solution $\epsilon(t)$ is exponentially colored noise (Ornstein-Uhlenbeck process) with properties:

$$\langle \epsilon(t) \rangle = 0, \quad \langle \epsilon(t)\epsilon(t') \rangle = \frac{D}{\gamma} e^{-\gamma|t-t'|}) \tag{17}$$

$\tau = 1/\gamma$ is the correlation time of the colored noise, and $D\tau^2$ is the intensity. Thus the colored noise driven system is described by the following SDE system:

$$\dot{s_i} = \sum_{m=1}^{M} 2a_m c_{mi} K_{mi} K_m + \epsilon_i(t), \qquad i = 1, \ldots, N \tag{18}$$

$$\dot{\epsilon_i} = -\gamma\epsilon_i + \sqrt{2D}\xi(t), \qquad i = 1, \ldots, N \tag{19}$$

$$\dot{a_m} = a_m K_m + \epsilon_m(t), \qquad m = 1, \ldots, M \tag{20}$$

$$\dot{\epsilon_m} = -\gamma\epsilon_m + \sqrt{2D}\xi(t), \qquad m = 1, \ldots, M \tag{21}$$

Numerical solution. To solve our equations we used the Euler-Maruyama method, which is one of the simplest numerical method for solving SDEs [45]. Consider the stochastic process X on the time interval $[t_0, T]$ and the following SDE:

$$\dot{X}_t = a(t, X_t) + b(t, X_t)\xi_t \tag{22}$$

First rewrite this equation to

$$dX_t = a(t, X_t)dt + b(t, X_t)\sqrt{dt}N(0, 1), \tag{23}$$

where $N(0, 1)$ denotes the standard normal random variable. The Euler-Maruyama approximation considers the discretization $t_0 = \tau_0 < \tau_1 < \cdots < \tau_n < \cdots < \tau_N = T$ of the time interval, and approximates X by a continuous time stochastic process Y satisfying:

$$Y_{n+1} = Y_n + a(\tau_n, Y_n)\Delta_n + b(\tau_n, Y_n)\sqrt{\Delta_n}N(0, 1), \tag{24}$$

where $Y_n = Y(\tau_n)$ and $\Delta_n = \tau_{n+1} - \tau_n$.

Simulation results. We have simulated the effect of noise on a large number of random SAT instances with different number of variables and constraint density parameters, $\alpha = M/N$. To illustrate how noise can affect trajectories in Fig. 10 we show the dynamics of one chosen s_i and a_m variable in a 3-SAT problem with $N = 50$ variables and $\alpha = 4.25$ constraint density (which falls in the hard-SAT region of 3-SAT). We plot the trajectory of the variables for three different noise intensities $I = 0.0005, 0.0015, 0.004$ and several correlation times. In Fig. 10a for example we show the trajectory of a variable at noise intensity $I = 0.0005$. Zooming on the trajectories (see inset) we see that white noise shows the largest fluctuations (red). Increasing the correlation time the curves seem to be much smoother (D is smaller), however the effect of noise is still there because of the correlations ($I = D\tau^2$ for colored noise). The figure shows that trajectories follow the same path up to a relatively long time ($\simeq 30$ even at large intensity Fig. 10e), when some of them suddenly get diverted. Nevertheless, the solution is still found, sometimes even much earlier than without noise. It is interesting that even if curves are plotted for 18 different parameter settings in total, there appear only 4 characteristic paths. It seems that noise is just switching between several strongly attracting trajectories.

Another way to investigate the sensibility of trajectories to noise is to plot the attractor basin maps. Fixing a random initial condition in a 3-SAT problem with $N = 50, \alpha = 4.25$ we vary only two randomly chosen variables (s_4 and s_5) along a 400×400 grid and colour the points according to the solutions they flow to up to time $t_{max} = 130$. Black colour indicates that no solution has been found. In Fig. 11 we show this map with 4 different noise-parameter settings. The intensity of noise clearly affects the trajectories. The map shows significant changes and becomes more and more blurry, but the solutions are still found with high probability. Estimating the size of attractor basins (probability of finding a solution) in the whole phase space (not only along the maps shown) indicates that noise does not induce significant changes. For the 4 settings used in Fig. 11 these show very small variability: 0.2078, 0.2126, 0.2050, 0.1970 (attractor basin marked with green), 0.066, 0.0604, 0.0728, 0.0686 (orange),0.3456, 0.3504,

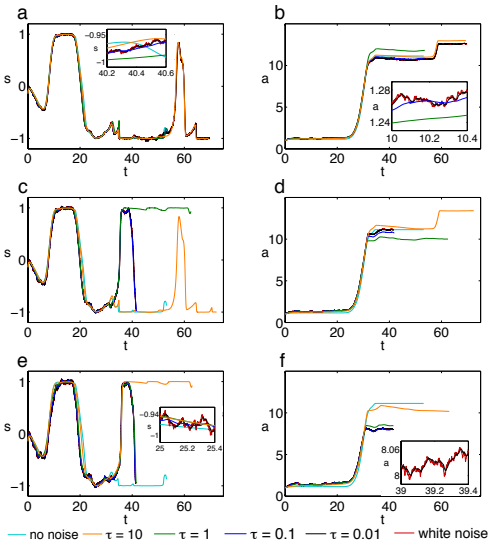

Figure 10: Time evolution of a randomly chosen s (left column) and a variable (right column) in a problem with $N = 50$ variables and $\alpha = 4.25$. Colors correspond to different τ correlation times (see legend). The intensity is a, b) $I = 0.0005$, c, d) $I = 0.0015$, e, f) $I = 0.004$. Trajectories terminate when a solution is found. A magnified portion is shown in the insets.

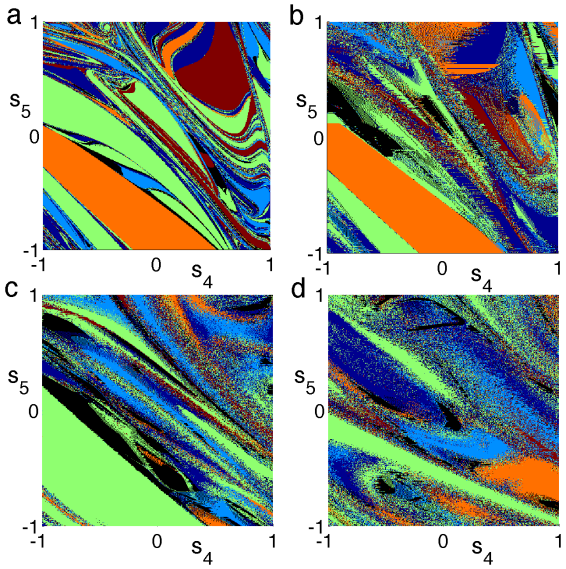

Figure 11: Attractor basins in a 3-SAT problem with $N = 50$, $\alpha = 4.25$. Fixing a random initial condition we change s_4 and s_5 along a 400×400 grid. Each point is colored according to the solution the dynamics flows to, and black when no solution has been found up to time $t_{max} = 130$. Noise-parameter settings: a) No noise, b) $\tau = 100$, $I = 0.0005$, c) $\tau = 0.1$, $I = 0.0005$, d) $\tau = 0.1$, $I = 0.002$. Larger intensity and smaller correlation time makes the maps blurrier.

0.3454, 0.3202 (navy blue),0.1266, 0.1266, 0.1210, 0.1312 (red),0.1562, 0.1540, 0.1568, 0.1574 (royal blue).

Concerning the viability of our SAT solving method in presence of noise the most important question is not whether trajectories get diverted by noise, but whether the probability of finding a solution changes. When there are more solutions no algorithm can control which solution is found, it will depend on initial conditions or other parameters. In our case there is a mapping between the initial conditions and the solutions found (Fig. 11). Noise changes this mapping, but if the expected time for finding a solution is not changed the goal of the algorithm is achieved. For this reason we measured the distribution of transient times. Taking $20,000$ random 3-SAT problems both in the easy- ($\alpha = 3.5$) and in the hard-SAT phase ($\alpha = 4.25$), each time we start the dynamics from a random initial condition and we measure what is the probability $p(t)$ that the solution has not been found up to time t. In Fig. 12a,b we plot this distribution in case of white noise for several parameter settings. Without noise this distribution follows an exponential decay $p(t) \sim e^{-\kappa t}$, characteristic to transient chaos, κ denoting the escape rate [14, 33]. In the presence of white noise mainly the end of the distribution changes. Fitting the curves (not shown) indicates that they follow $p(t) = c + be^{-\kappa t}$, meaning they saturate to a constant c in $t \to \infty$. The fitted parameters κ and c are shown as function of the noise intensity in Fig. 12c,d. Up to $I = 1.5 \times 10^{-3}$ almost no change can be noticed, $c \simeq 0$. Above this threshold, which is fairly independent of N or α, both c and κ suddenly increase. Nevertheless, even at larger intensities the first part of the distribution ($t < 60$) is unaffected, meaning there are still many short trajectories finding solutions not disturbed by noise. $c \neq 0$ means there will be trajectories never finding a solution. This happens, because noise can sometimes push out trajectories from the $H = [-1, 1]^N$ hypercube, from where they cannot return. The effect of colored noise is very similar (Fig. 13a,c), the intensity threshold is approximately the same. Studying the effect of correlation time the statistics was made on 20000 random 3-SAT problems with $N = 50$ variables at two different constraint densities and noise intensity $I = 0.002$ (Fig. 13b,d). The effect is largest when τ is small, the threshold being somewhere around $\tau \simeq 5$.

Robustness of the CNN model Even if the tolerated noise level is relatively high, implementing this system in electronic circuits poses some challenges, most important being the unbounded nature of the $\boldsymbol{a}$ auxiliary variables. Subsequently the polynomial-time complexity can be achieved only up to a limit of problem sizes, depending on the possible range of auxiliary variables assured by the implementation. In hope of more straightforward implementation we recently developed a cellular neural network type model, which can also solve SAT problems [3, 4]. CNN models have already been implemented and used in engineering applications [41]. In this case all variables are bounded, no extra energy can be introduced into the system to assure the polynomial efficiency, but the speed of computing with actual analog circuits could still offer unusually high computational efficiency.

In the CNN model we again have two types of variables, also called cells [**?**], representing the boolean variables (s-type) and clauses (a-type) respectively. As in any recurrent neural network model, each cell has a state value and an output value. The output function of s-type variables is $f(s_i) = \frac{1}{2}(|s_i + 1| - |s_i - 1|)$ assuring that $f(s_i) \in [-1, 1]$, and the output of a-type variables will now be limited to $[0, 1]$ interval through the output function $g(a_m) = \frac{1}{2}(1 + |a_m| - |1 - a_m|)$. The connections between cells are based on the SAT problem and are again determined by the c_{mi} values as defined above (e.g. if variable s_i appears in its normal form in clause a_m there will be a connection between these two cells with weight $c_{mi} = +1$). The CNN solving a k-SAT problem is:

$$\dot{s}_i = -s_i + Af(s_i) + \sum_m c_{mi} g(a_m) \tag{25}$$

$$\dot{a}_m = -a_m + Bg(a_m) - \sum_i c_{mi} f(s_i) + 1 - k \tag{26}$$

where A, B are self-connecting parameters of the two types of cells. In [**?**, 4] we have shown that this model also presents a one-to-one correspondence between its stable fixed points and the SAT solutions.

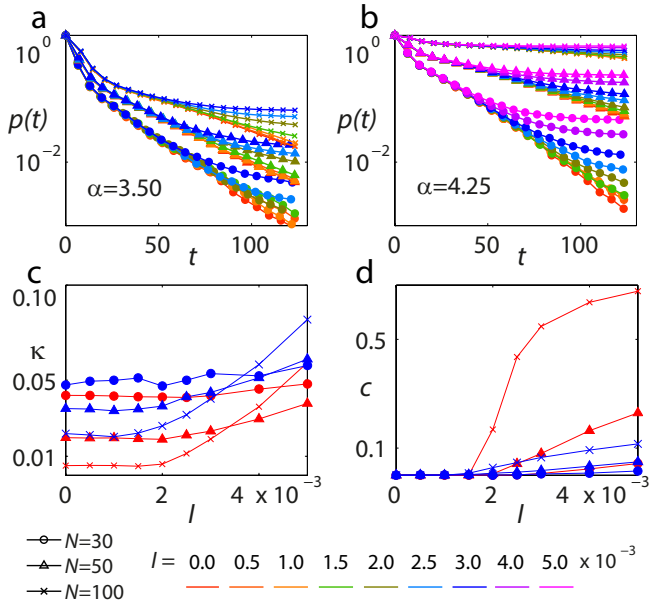

Figure 12: Fraction of unsolved problems $p(t)$ as function of time in the presence of white noise with varying intensities I (see legend) for system sizes $N = 30$ (circle), $N = 50$ (triangle) and $N = 100$ (star) and constraint densities a) $\alpha = 3.5$ b) $\alpha = 4.25$. After fitting $p(t) = c + be^{-\kappa t}$ we plot c) κ and d) c as function of the intensity for the two different α: 3.5 (blue), 4.25 (red).

While limit cycles do exist, these can be avoided by properly choosing the values of A and B, which have an optimal region fairly independent of the particular SAT problem (the values of N, M, and even k) [4].

To study the effect of noise in this dynamical system, we used the same methods as described for the previous model. Each variable is assigned a corresponding random variable $\xi_i^{(s)}(t)$ and $\xi_m^{(a)}(t)$ and the terms $\sqrt{2D}\xi_i^{(s)}(t)$ and $\sqrt{2D}\xi_m^{(a)}(t)$ are added to the right hand side of equations. For coloured noise the Langevin equations must also be included the same way as in Eqs. (**??**-**??**). Similar analysis have been performed as in the previous case. The trajectories being chaotic are naturally sensitive to noise, but plotting the $p(t)$ distribution of transient times it remains fairly unaffected up to a large intensity. Here the output functions of cells do not allow the system to escape from the searching space and distributions do not saturate, they keep a power-law decay $p(t) \sim t^{-\beta}$ [4]. This is different compared to the exponential decay seen in the previous model and it indicates in fact the qualitative difference between the two models, this is why the CNN model cannot achieve polynomial continuous-time efficiency (see [4]).

In Fig. 14a, b we plot these distributions at two different A, B parameter settings, one of them being in the optimal region (where the system works most efficiently, as shown in [4]), and the other being non-optimal. Only results for white noise are shown, but colored noise has very similar effects. The insets show the power β, which shows a linear decay as function of the intensity. It is interesting that in case of the non-optimal parameter setting up to a surprisingly large noise intensity $I \simeq 10^{-2}$ the system becomes more efficient. This is promising for actual implementations, and decreases even more the sensitivity of the model to the parameters A, B. This intensity is well tolerated also in case of the optimal parameters.

In circuit implementations of CNN models another concern is the precision of connection weights. When producing circuit elements (resistors, capacitors, etc.) the parameters of these elements will show errors compared to the theoretically proposed values. For that reason we also investigated the effect

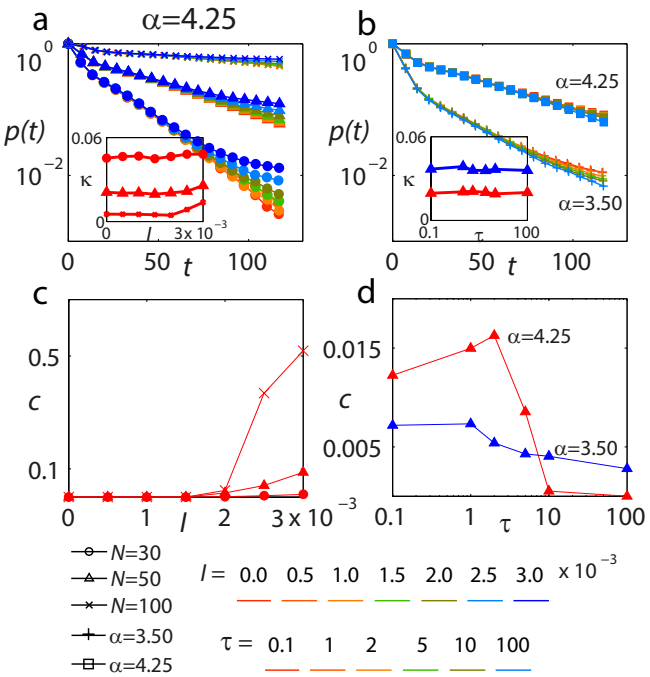

Figure 13: Fraction of unsolved problems $p(t)$ for a) $\alpha = 4.25$, $\tau = 1$ and varying N and I (see legends); b) $N = 50$, $\alpha = 3.50$, 4.25, $I = 0.002$ and varying correlation time τ (legend). Insets show the variation of κ. c,d) show the variation of the constant c as function of I and τ respectively.

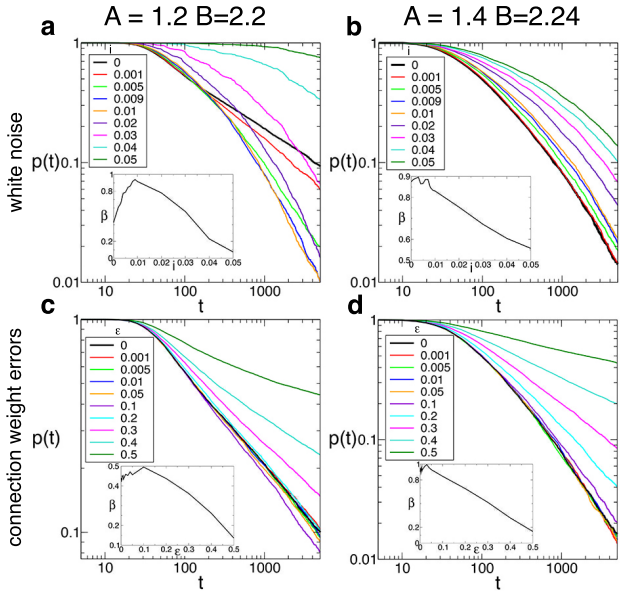

Figure 14: Fraction of unsolved problems for $N = 20$, $\alpha = 4.25$ with parameters a,c) $A = 1.2$, $B = 2.2$ and b,d) $A = 1.4$, $B = 2.24$ in presence of a,b) white noise with different intensities (see legend) and c,d) connection weight errors of different magnitude (see legend). Insets show the variation of β as function of noise intensity.

of errors in the c_{mi} connection weights. Each connection weight is assigned a random deviation value between $[-\epsilon, \epsilon]$, constant in time. In Fig. 14c, d we plot the $p(t)$ distributions for the two A, B parameter settings used above, for different values of ϵ. A very similar behavior can be observed. Surprisingly large errors up to $\epsilon \approx 0.2$ still allow the system to function.

Discussion It is necessary to ask how the noise intensities shown here to be acceptable compare to actual noise observed in circuits and analog computer devices. In CNN computers the amplitude of noise has been shown to be at least two orders of magnitude smaller than the values of voltages (this has even improved in later implementations). For example in the Q-eye chip [40, 41] the maximal signal-to-noise ratio (SNR) is 45dB. By definition SNR$= 20 \log_{10} \frac{A_{signal}}{A_{noise}}$, meaning that the amplitude of noise on the chip is at least $10^{2.25}$ times smaller than the amplitude of the signal. In case of white noise and for values changing between $[-1, 1]$ this would mean fluctuations $A_{noise} = \sqrt{2D} < 10^{-2.25}$, corresponding to a white noise intensity $I < 1.6 \times 10^{-5}$. In our CNN model the acceptable intensity is around $I = 10^{-2}$, and even in the first, more sensitive model it is around $I = 1.5 \times 10^{-3}$.

The greatest problem in analog computing has always been the presence of noise. Until now developing robust dynamical algorithms have been possible only for simple converging systems with Lyapunov dynamics. Solving complex optimization problems such simple dynamics can minimize an energy function only by converging to the closest local minimum [**?**, 50]. In order to find the global optimum this Lyapunov dynamics was used as a basic step of more complex algorithms similar to those used in digital computing (such as simulated annealing etc.) [51–53]. This significantly reduced the power of analog computing. Our models offer two great advantages: 1) the optimization is realized through one single dynamical process: the dynamics is started from a random initial condition and searches until finding the solution, without further intervention needed by the user; 2) In spite of the fact that hard optimization problems necessarily present transiently chaotic dynamics, the probability of finding solutions is robust to noise. The large noise intensity levels tolerated promise the possibility for highly robust and efficient physical implementations.

These results were published in Europhysics Letters (2014).

Research and collaborations beside the originally planned activities

During these three years some great opportunities came to collaborate with foreign researchers. As the director of a young Romanian research team I considered it important to maintain these relations and participate in these research projects even if these were out of planning. These were extra activities and did not alter the course of the originally planned project.

Collaborating with my previous supervisor, Prof. Zoltán Toroczkai, at the University of Notre Dame, IN, USA we finalized some publications, which were then accepted in Physical Review E (2012) and Nature Communications 2014 (see dissemination activities). These are theoretical works related to complex networks and graph theory.

Also in 2012, collaborating with Prof. Toroczkai, Prof. József Baranyi from UK, and Zoltán Lakner form Hungary, we published a paper in PLoS One about the complexity of the international agro-food trade network.

The greatest opportunity came as a collaboration with the team of Dr. Henry Kennedy, a neuroscientist form the Stem Cell and Brain Research Institute from Lyon. While I previously collaborated with them as part of other projects, now a new opportunity came. They asked me to participate in writing a review paper in the neuroscience special issue of **Science**. This is related to the inter-areal network of the brain, which is a highly complex system. I participated in analyzing their data form a physicist point of view. This paper appeared in November 2013 (see dissemination of results).

Dissemination of results

Also see our webpage: http://sirius.phys.ubbcluj.ro:33380/Ercsey-Ravasz/TurbComp/

Publications

ISI Journal articles:

- R. Sumi, M. Varga, Z. Toroczkai, M. Ercsey-Ravasz, "From order to chaos in random satisfiability problems", to be submitted soon

- Y. Ren, M. Ercsey-Ravasz, P. Wang, M.C. Gonzalez, Z. Toroczkai, "Predicting commuter flows in spatial networks using a radiation model based on temporal ranges", **Nature Communications**, 5, 5347 (2014) (impact factor: 10.742)

-R. Sumi, B. Molnár, M. Ercsey-Ravasz, "Robust optimization with transiently chaotic dynamical systems", **Europhysics Letters**, 106, 40002, (2014) (impact factor 2.269, relative infl. score 2.603)

- N.T. Markov, M. Ercsey-Ravasz, D.C. Van Essen, K. Knoblauch, Z. Toroczkai, H. Kennedy, "Cortical High-density Counter-stream Architectures", **Science**, 342, 1238406, 2013 (impact factor 31.027, relative influence score 30.90676)

- M. Ercsey-Ravasz, Z. Toroczkai, "The Chaos Within Sudoku", **Nature Scientific Reports** 2, 755 (2012) doi:10.1038/srep00725 (impact factor: 2.927)

- M. Ercsey-Ravasz, R. Lichtenwalter, N.W. Chawla, Z. Toroczkai, "Range-limited Centrality Measures in Non-weighted and Weighted Complex Networks, **Physical Review E** 85, 066103 (2012). arxiv:1111.5382 (impact factor 2.313, relative influence score 1.43073)

- M. Ercsey-Ravasz, Z. Toroczkai, Z. Lakner, J. Baranyi, "Complexity of the International Agro-Food Trade Network", **PLoS ONE** 7(5), e37810 (2012). doi:10.1371/journal.pone.0037810 (impact factor 3.73, relative influence score 3.73269)

ISI Conference Proceedings:

- B. Molnár, M. Ercsey-Ravasz, "Analog dynamics for solving max-SAT problems", **Proceedings of CNNA 2014**, Notre Dame, IN, USA (2014)

- B. Molnár, R. Sumi, M. Ercsey-Ravasz, "A CNN SAT-solver robust to noise", **Proceedings of CNNA 2014**, Notre Dame, IN, USA (2014)

- K. Knoblauch, M. Ercsey-Ravasz, H. Kennedy, Z. Toroczkai, "The Brain in Space", **Proc. of IPSEN**, Paris, May (2014).

- B. Molnár, M. Ercsey-Ravasz, Z. Toroczkai, "Continuous-time Neural Networks Without Local Traps for Solving Boolean Satisfiability", **Proceedings of CNNA 2012**, p. 4012, Torino, Italy (2012)

Book chapters:

- M. Ercsey-Ravasz, Z. Toroczkai, "Döntesek fizikája és rejtvények káosza" ("Physics of decision making and chaos of puzzles") in A fizika, matematika és művészet találkozása az oktatásban, kutatásban (Physics, mathematics and arts in education and research), Ed.: A. Juhász, T. Tél, Publisher: Science Department of the Eötvös Lóránd University, Hungary, 2013.

Other publications:

- M. Ercsey-Ravasz, Z. Toroczkai, "A döntéshozatal és a Sudoku káosza" ("The Chaos Within Sudoku and Decision Making"), Természet Világa (World of Nature), invited paper in the special issue "Káosz, Környezet, Komplexitás" ("Chaos, Environment, Complexity"), Budapest, Hungary, October 2013

Conferences and Workshops

- B. Molnar, R. Sumi, M. Ercsey-Ravasz, "CNN computers for solving SAT problems" (poster), Summer School on Statistical Physics of Complex and Small Systems, Palma de Mallorca, Spain, 2014 Sept.

- R. Sumi, M. Varga, Z. Toroczkai, M. Ercsey-Ravasz, "Transient Chaos Provides Hardmess Measure for Constraint Satisfaction Problems" (oral presentation), International Conference CHAOS 2014, Lisabon, Portugal, June 2014

- M. Ercsey-Ravasz "Analog approaches to hard optimization: from Sudoku to CNNs" (oral presentation), Statistical Mechanics of Unsatisfiability and Glasses, Mariehamn, Finland, May, 2012

- R. Sumi "Chaotic phase transition in an analog approach to constraint satisfaction" (oral presentation), CHAOS 2012 -5th Chaotic Modeling and Simulation International Conference , Athen, Greece, 12-15 June, 2012

- M. Ercsey-Ravasz "Döntések fizikája és rejtvények káosza" ("Physics of decisions and chaos of puzzles")(invited talk), A fizika, matematika és művészet találkozása az oktatásban (Physics, mathematics and arts in high school) Tg. Mures, Romania, 15 August, 2012

- M. Ercsey-Ravasz "Solving constraint satisfaction problems via transiently chaotic analog systems and CNN dynamics" (invited talk), CNNA 2012, Torino, Italy (2012)

- B. Molnár, "Continuous-time Neural Networks Without Local Traps for Solving Boolean Satisfiability" (oral presentation), CNNA 2012, Torino, Italy (2012)

- M. Ercsey-Ravasz has been invited to the Information Technology Department of the Pázmány Péter Catholic University in Hungary, where she gave a talk with the title: "Asymmetric neural networks for solving constraint satisfaction", 22 February, 2013

- R. Sumi was accepted to participate on the Summer Programming and Tuning Massively Parallel systems (PUMPS) Summer School, Barcelona, July 2013.

Echoes in the media:

The article "The chaos within Sudoku" which appeared in Nature Scientific Reports was accessed by more than 50,000 times only in the first week and generated a large echo in the international media. Articles and comments appeared in prestigiuous newspapers such as: Huffington Post, Daily Mail, Technology review:

- http://www.huffingtonpost.com/william-g-gilroy/notre-dame-researcher-sudoku_b_1971656.html
- http://www.dailymail.co.uk/sciencetech/article-2216642/Problem-solved–sort-Mathematicians-come-formula-complete-Sudoku-trying-understand-longer-doing-puzzle.html
- http://www.technologyreview.com/view/428729/mathematics-of-sudoku-leads-to-richter-scale-of/

The romanian news media has also published short news and comments about our results: Income Magazine, Antena 3, Ad astra, Szabadsag.

- http://www.incomemagazine.ro/articol_86565/ce-legatura-e-intre-haos-sudoku-si-doi-fizicieni-romani.html
- http://jurnalul.ro/stiri/observator/ce-legatura-e-intre-haos-sudoku-si-doi-fizicieni-romani-627519.html
- http://www.antena3.ro/romania/articolul-scris-de-doi-romani-a-facut-inconjurul-lumii-au-reusit-sa-faca-ordine-in-haos-190186.html
- http://www.ad-astra.ro/posts/view_post.php?post_id=2039&lang=ro
- http://www.szabadsag.ro/szabadsag/servlet/szabadsag/template/archive%2CPArchiveArticleSelectedScreen.vm/id/81245/mainarticle/false/web/false/quick/false/ynews/false

Our article in PLoS One has also generated some echoes in the international media on Science Daily and Wired.com:

- http://www.sciencedaily.com/releases/2012/06/120607180241.htm
- http://www.wired.com/wiredscience/2012/06/food-trade-complex/

Other dissemination activities

We consider it important to reach out to the general public and especially high school students.

- Maria Ercsey-Ravasz gave a talk on the "Szekely Tehetseg Napok" (Szekler Talent Day) organised in Targu Mures, Sapientia University, for high school teachers and students of the Szekler region of Transylvania, September 2014. Title of the talk: "Kozelkep az agy halozatarol" (Snapshot on the Network of the Brain).

- Botond Molnar gave a talk for students of the Szekely Miko Theoretical Highschool in Sfintu Gheorghe, with title "A Sudokutol a cellularis hullamszamitogepekig" (From Sudoku to Cellular Wave Computers), December 2013.

- Maria Ercsey-Ravasz gave a talk on the "Weekend for Hungarian High-school students" organized by Babes-Bolyai University, with title: "Káosz a Sudokuban" ("The Chaos Within Sudoku"), October 11 2013

- Maria Ercsey-Ravasz gave a talk on the "Weekend for Hungarian High-school students" organized by Babes-Bolyai University, with title: "Fizika a majmok agyában" ("Physics in the brain of monkeys"), October 12 2013

- Botond Molnár gave a talk on the "Weekend for Hungarian High-school students" organized by Babes-Bolyai University, with title: "Analog számítógépek" ("Analog computing"), October 12 2013

-Maria Ercsey-Ravasz gave a talk with title "The chaos within Sudoku" to the participants of the Scientific Conference of Highschool Students, May 2013

-Maria Ercsey-Ravasz gave a talk at a conference organized for highschool teachers in Transylvania: "Döntések fizikája és rejtvények káosza" ("Physics of decisions and chaos of puzzles"), A fizika, matematika és művészet találkozása az oktatásban (Physics, mathematics and arts in high school) Tg. Mures, Romania, August 15, 2012.

-We participated on the conference "Vándoregyetem" organized for college students, October 2012

-Maria Ercsey-Ravasz was invited to the Unitarian Highschool János Zsigmond, where she gave a talk with title "This Is All Physics", April 2012

-Botond Molnár and Maria Ercsey-Ravasz have also gave disseminating talks to the highschool students who participated on the Vermes Miklós Physics Competition, 2012.

References

1. Ercsey-Ravasz M, Toroczkai Z (2011) Optimization hardness as transient chaos in an analog approach to constraint satisfaction. Nature Physics 7: 966-970.

2. M. Ercsey-Ravasz, Z. Toroczkai, The Chaos Within Sudoku, *Nature Scientific Reports* 2, 755 (2012) doi:10.1038/srep00725

3. B. Molnár, M. Ercsey-Ravasz, Z. Toroczkai, Continuous-time Neural Networks Without Local Traps for Solving Boolean Satisfiability", *Proceedings of CNNA 2012*, p. 4012, Torino, Italy (2012)

4. Molnár B., Ercsey-Ravasz M.PLOS ONE8,2013,e73400.

5. Cook SA (1971) The complexity of theorem-proving procedures. In: STOC. pp. 151-158.

6. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). W. H. Freeman & Co Ltd, first edition edition.

7. Zdeborová, L. & Mézard, M. Locked constraint satisfaction problems. *Phys. Rev. Lett.* **101**, 078702 (2008).

8. Zdeborová, L. & Mézard, M. Constraint satisfaction problems with isolated solutions are hard. *J. Stat. Mech.: Theor. Exp.* P12004 (2008).

9. http://forum.enjoysudoku.com/the-hardest-sudokus-new-thread-t6539.html

10. http://wiki.karadimov.info/index.php/Sudoku_algorithms#Exceptionally_difficult_Sudokus_.28hardest_Sudokus.29

11. http://en.wikipedia.org/wiki/Algorithmics_of_sudoku#Exceptionally_difficult_Sudokus_.28hardest_Sudokus.29

12. Elser, V., Rankenburg, I. & Thibault, P. Searching with iterated maps. *Proc. Natl. Acad. Sci. USA* **104**, 418-423 (2007).

13. http://www.youtube.com/watch?v=y4_aSLP9g_w

14. Lai, Y.-C. & Tél, T. *Transient Chaos: Complex Dynamics on Finite-Time Scales* (Springer 2011).

15. Cvitanović, P., Artuso, R., Mainieri, R. Tanner, G. & Vattay, G. *Chaos: Classical and Quantum*, ChaosBook.org/version13 (Niels Bohr Institute, Copenhagen 2010).

16. http://www.sudokuoftheday.co.uk

17. http://cavemancircus.com/2009/11/05/the-hardest-sudoku-puzzle-ever/

18. http://www.guardian.co.uk/media/2010/aug/22/worlds-hard est-sudoku

19. http://www.usatoday.com/news/offbeat/2006-11-06-sudoku_x.htm

20. Eppstein, D. Solving Single-digit Sudoku Subproblems, http://arxiv.org/abs/1202.5074v2

21. Rosenhouse, J. & Taalman, L. *Taking Sudoku Seriously: The Math Behind the World's most Popular Pencil Puzzle* (Oxford University Press, New York, 2011).

22. Kirkpatrick S, Selman B (1994) Critical-Behavior in the Satisfiability of Random Boolean Expressions. Science 264: 1297-1301.

23. Wah B, Chang Y (1997) Trace-based methods for solving nonlinear global optimization and satisfiability problems. J of Global Optimization 10: 107-141.

24. Wah B, Wang T, Shang Y, Wu Z (2000) Improving the performance of weighted Lagrange-multiplier methods for nonlinear constrained optimization. Information Sciences 124: 241-272.

25. Nagamatu M, Yanaru T (1996) On the stability of Lagrange programming neural networks for satisfiability problems of propositional calculus. Neurocomputing 13: 119-133.

26. Gray P. R., Hurst P. J., Lewis S. H., Meyer R. G. Analysis and Design of Analog Integrated Circuits, John Wiley & Sons, Inc., 2009.

27. Gillespie D. T.,*Markov Processes: An Introduction for Physical Scientists*, Academic Press, Inc, San Diego, 1992

28. Gillespie D. T., *Am J Phys*, **64**, 225 (1996).

29. Gardiner C., Stochastic Methods: A Handbook for the Natural and Social Sciences, Springer 2009.

30. Laing C., Lord G. J., Stochastic Methods in Neuroscience, Oxford University Press 2010.

31. Hänggi P., Jung P., *Advances in Chemical Physics*, **89** (1995).

32. Kloeden P. E., Platen E., *Numerical solution of stochastic differential equations*, Springer-Verlag, Berlin, 1999.

33. Tél T., Lai Y.-C.Physics Reports, 460, 2008, 245

34. Seoane J.M., Huang L., Sanjuán M.A.F., Lai Y.-C. Phys. Rev. E, 97, 2009, 047202.

35. Reimann P. J. Stat. Phys., 85, 1996, 403.

36. Franaszek M.Phys. Rev. A, 44, 1991, 4065.

37. Bernal J.D., Seoane J., Sanjuán M.A.F. Phys. Rev. E, 88, 2013, 032914.

38. Altmann E.G., Endler A. Phys. Rev. Lett., 105, 2010, 244102.

39. Faisst H., Eckhardt B. Phys. Rev. E, 68, 2003, 026215.

40. Zarandy A., Rekeczky C. IEEE Circ. and Syst. Magazine, 5, 2005, 36.

41. http://www.anafocus.com

42. Kirkpatrick S., Selman B. Science, 264, 1994, 1297.

43. Gray P. R. Analysis and Design of Analog Integrated Circuits 5th Edition (Wiley) 2009.

44. Gillespie D. T. Markov Processes: An Introduction for Physical Scientists (Academic Press) 1991.

45. Kloeden P. E., Platen E. Numerical Solution of Stochastic Differential Equations (Springer, NY) 2011.

46. Gillespie D. American J. of Physics, 64, 1996, 225.

47. Gardiner C. Handbook of Stochastic Methods: for Phys., Chem. and the Natural Sci. (Springer) 2004.

48. Laing C., Lord G. J.Stochastic Methods in Neuroscience (Oxford University Press, USA) 2009.

49. Hanggi P., Jung P. Adv. in Chemical Physics, 89 , 1995, 239.

50. Hopfield J., Tank D. Bio. Cyber., 52, 1985, 141.

51. Sima J., Orponen P. Neur. Comp., 15, 2003, 693.

52. Ercsey-Ravasz M., Roska T., Néda Z. Physica A-Stat. Mech. and its Appl., 388, 2009, 1024.

53. Ercsey-Ravasz M., Roska T. , Néda Z. Physica D-Nonlinear Phenomena, 237, 2008, 1226.

PN-II-RU-TE-2011-3-0121
Project director,
Dr. Maria-Magdolna Ercsey-Ravasz