

ARTICLE

DOI: 10.1038/s41467-018-07327-2

OPEN

A continuous-time MaxSAT solver with high analog performance

Botond Molnár^{1,2,3}, Ferenc Molnár⁴, Melinda Varga^{4,5}, Zoltán Toroczka⁴ & Mária Ercsey-Ravasz ^{1,3,6}

Many real-life optimization problems can be formulated in Boolean logic as MaxSAT, a class of problems where the task is finding Boolean assignments to variables satisfying the maximum number of logical constraints. Since MaxSAT is NP-hard, no algorithm is known to efficiently solve these problems. Here we present a continuous-time analog solver for MaxSAT and show that the scaling of the escape rate, an invariant of the solver's dynamics, can predict the maximum number of satisfiable constraints, often well before finding the optimal assignment. Simulating the solver, we illustrate its performance on MaxSAT competition problems, then apply it to two-color Ramsey number $R(m, m)$ problems. Although it finds colorings without monochromatic 5-cliques of complete graphs on $N \leq 42$ vertices, the best coloring for $N = 43$ has two monochromatic 5-cliques, supporting the conjecture that $R(5, 5) = 43$. This approach shows the potential of continuous-time analog dynamical systems as algorithms for discrete optimization.

¹Faculty of Physics, Babeş-Bolyai University, Cluj-Napoca 400084, Romania. ²Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca 400084, Romania. ³Transylvanian Institute of Neuroscience, Cluj-Napoca 400157, Romania. ⁴Department of Physics, University of Notre Dame, Notre Dame, IN 46556, USA. ⁵Center for Vascular Biology Research and Department of Medicine, Beth Israel Deaconess Medical Center, Boston, MA 02215, USA. ⁶Romanian Institute of Science and Technology, Cluj-Napoca 400487, Romania. Correspondence and requests for materials should be addressed to Z.T. (email: toro@nd.edu) or to M.E.-R. (email: ercsey.ravasz@phys.ubbcluj.ro)

Digital computing, or Turing's model of universal computing is currently the reigning computational paradigm. However, there are large classes of problems that are apparently intractable on digital computers, requiring resources (time, memory, and/or hardware) for their solution that scale exponentially in the input size of the problem (NP-hard)¹. Such problems, unfortunately, are abundant in sciences and engineering, for example, the ground-state problem of spin-glasses in statistical physics^{2,3}, the traveling salesman problem⁴, protein folding⁵, bioinformatics⁶, medical imaging⁷, scheduling⁸, design debugging, Field Programmable Gate Array routing⁹, probabilistic reasoning¹⁰, etc. It is believed that in order to make progress on solving such problems one might have to look beyond computation with digital Turing machines. Analog computing and quantum computing present two promising and possibly revolutionary approaches, complementing complementary metal-oxide-semiconductor technology¹¹ in solving certain types of hard computational problems. However, quantum computing currently faces fundamental physics and engineering challenges that still need to be solved^{12,13}, leaving analog computing as a possibly more feasible option^{14,15}. Although explored in the 1950s, it was abandoned in favor of the digital approach, due to the technical challenges it posed (for a historical survey see ref. 16). By now, however, technology has matured enough to control much better the physics at the small-scale, making it worthwhile revisiting analog computing, at least at an application-specific level. Accordingly, increasing effort is being dedicated to both problem-driven^{17–29} but also general purpose analog computing^{30–32}, including analog computability theory^{32–34}.

One quintessential family of intractable problems that could potentially be tackled with special-purpose analog devices are Boolean satisfiability problems, both in their decision (SAT) and optimization forms (MaxSAT). In SAT we are given a set of M logical clauses in conjunctive normal form (CNF), $C_1, C_2, \dots, C_M$ over Boolean variables $x_1, \dots, x_N, x_i \in \{0, 1\}$. Typically, one studies k -SAT problems where every clause involves k literals (a literal is a variable or its negation). The task is to set the truth values of all the variables such that all the clauses evaluate to TRUE ("0" = FALSE, "1" = TRUE). It is well known that k -SAT with $k \geq 3$ is NP-complete and thus any efficient solver for 3-SAT implies an efficient solver for all problems in the NP class (Cook-Levin theorem, 1971)^{8,35,36}. The NP class is the set of all decision-type problems where one can check in polynomial time the correctness of a proposed solution (but finding such a solution can be exponentially costly).

MaxSAT (or Max k -SAT) is the optimization version of SAT. It has the same formulation as SAT (or k -SAT), but the task is to maximize the number of satisfied clauses. It is harder than SAT as one cannot guarantee in polynomial time the optimality of the solution (unlike for SAT), for problems that do not admit full satisfiability. Thus, although MaxSAT is NP-Hard, it is not known to be in NP unless $P = NP$. Several discrete algorithms were developed for MaxSAT, including statistical mechanics inspired methods such as Survey Propagation^{37,38}. SAT and MaxSAT have a very large number of applications, with SAT solvers becoming an important back-end technology. Applications include scheduling, planning and automated reasoning, electronic design automation, bounded model checking, design of experiments, coding theory, cryptography, and drug design, see ref. 39–41.

A continuous-time deterministic system (CTDS) based on ordinary differential equations (ODEs), was recently proposed as an analog SAT solver, in ref. 42. It was designed such that all the SAT solutions appear as attractive fixed points for the dynamics while no other attractors exist trapping the dynamics. For hard

problems its behavior becomes chaotic, showing that problem hardness and chaos^{43,44} are related notions within this context, and thus chaos theory can be used to study computational complexity. For fully satisfiable SAT problems the chaos is necessarily transient⁴⁵, with the trajectory eventually settling onto one of its attracting fixed points (a SAT solution). Using numerical experiments, the CTDS was shown to solve hard SAT problems in polynomial continuous-time⁴², but at the expense of auxiliary variables growing exponentially. In a hardware realization, this implies a trade-off between time and energy costs. However, since one can control/generate energy much better than time itself, this presents a viable option for time-critical applications.⁴⁶ proposed an analog circuit design for the CTDS, showing a 10^4 -fold speedup on hard 3-SAT problems, when compared to state-of-the-art SAT solvers^{47,48} on digital machines.

Here, we present an extension of the CTDS such as to solve MaxSAT problems. The idea is based on the observation that the CTDS makes no assumptions about problem satisfiability and thus, even for unsatisfiable SAT problems, the dynamics will still minimize the number of unsatisfied clauses. What we need to determine, however, is the likelihood of the optimality of the best solution found by analog time t , as function of t , which we achieve heuristically, by analyzing the statistics of a dynamical invariant, the escape rate⁴⁵. In the following we will refer to our analog MaxSAT solver as Max-CTDS. We test its performance using hard benchmark MaxSAT problems, in particular, on all the 454 random benchmark problems from the 2016 MaxSAT competition⁴⁹, showing that it achieves the same or very close results to the overall best competition solver. As another application of our approach, we consider the famous problem of Ramsey numbers^{50,51}. The Ramsey number $R(m, m)$ is the smallest order of a complete graph such that any coloring of its edges with two colors has a monochromatic clique of order m . The SAT formulation of this problem, of finding a coloring without monochromatic m -cliques, is fully satisfiable below the Ramsey number, whereas at the Ramsey number, it becomes MaxSAT for the first time. $R(5, 5)$ is still open, only the bounds $43 \leq R(5, 5) \leq 48$ are known^{52,53}. Finding Ramsey numbers is challenging due to the convoluted structure of the search space, and its sheer size: there are $2^{\binom{N}{2}}$ possible colorings of a complete labeled graph on N nodes ($\approx 10^{271}$ for $N = 43$). For $m = 5$ (equivalent to a 10-SAT/MaxSAT problem) Max-CTDS finds good colorings for up to $N = 42$, whereas for $N = 43$ finds a coloring with only two monochromatic 5-cliques sitting on 6 nodes, the lowest energy coloring found so far, to our best knowledge, adding further support to the conjecture that $R(5, 5) = 43$. We conclude with a discussion on analog solvers and their realization in hardware.

Results

A continuous-time dynamical system solver for MaxSAT. Here, we focus on continuous-time systems, in which both the state variables $\mathbf{s} = (s_1, \dots, s_N)$ and the time variable t are real numbers, $s_i \in \mathbb{R}$, $t \in \mathbb{R}$, updated continuously by the algorithm ("software"), in form of a set of ordinary differential equations (ODEs)⁵⁴ $ds/dt = \mathbf{F}(\mathbf{s}(t), t)$, $t \in \mathbb{R}$, see ref. 32 for a review. The process of computation is interpreted as the evolution of the trajectory (the solution to the ODEs) $\mathbf{s}(t) = \Psi_t(\mathbf{s}_0)$, toward an attractive fixed-point state $\mathbf{s}^*$: $\lim_{t \rightarrow \infty} \Psi_t(\mathbf{s}_0) = \mathbf{s}^*$, representing the answer/solution to the problem. Clearly, we want to find $\mathbf{s}^*$, and the challenge is to design $\mathbf{F}$ such that the solutions to the problem (when they exist) appear as attractive fixed points for the dynamics and no other, nonsolution attractors exist that could trap the dynamics.

Our MaxSAT solver is based on a previously introduced SAT solver⁴², which we now briefly describe; more details are given in

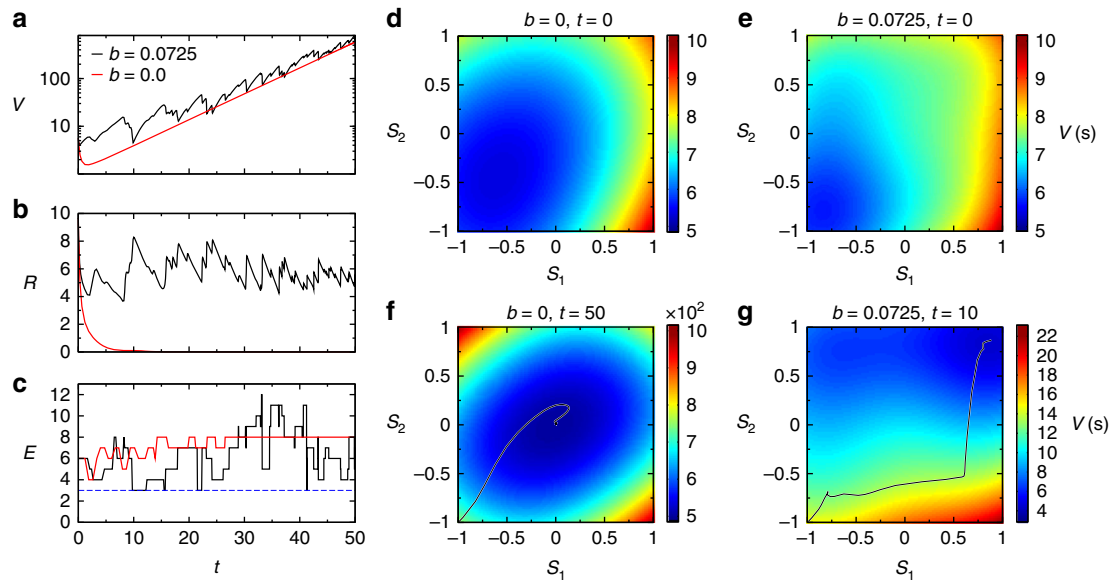


Fig. 1 MaxSAT solver dynamics. The Max 3-SAT formula used here has $N = 10$, $M = 80$ (clauses given in the Supplementary Data 1). **a** The potential V , **b** the radius $R = \sqrt{\sum_i s_i^2}$, and **c** the number of unsatisfied clauses (energy) E as function of analog time t for the original dynamics corresponding to $b = 0$ (red) and the modified dynamics with $b = 0.0725$ (black). **d–g** Colormaps of the potential $V(\mathbf{s}(t), \mathbf{a}(t))$ in the plane (s_1, s_2) . At a given time instant t we fix all values $s_j(t)$, $j = 3, \dots, N$ and $a_m(t)$, $\forall m$ and change only s_1, s_2 in the $[-1, 1] \times [-1, 1]$ plane, showing the instantaneous potential energy landscape V in this plane. The curves indicate the projection of the trajectory onto (s_1, s_2) up to the indicated time t . In $t = 0$, $s_1 = s_2 = -1$. For $b = 0$, the dynamics converges to $\mathbf{s} = \mathbf{0}$, which is the centre of a deep well in the potential landscape. For $b = 0.0725$, the centre is not a minimum anymore and at time $t = 10$ the orthant with minimal energy $E_{\min} = 3$ is found (the solution), shown as a blue dotted line in the $E(t)$ figure

Methods. Let us assign the variable $s_i = 2x_i - 1$ to every Boolean variable x_i (when $x_i = 0$, $s_i = -1$ and when $x_i = 1$, $s_i = 1$), but allow s_i to vary continuously in the $[-1, 1]$ interval. The continuous dynamical system $\frac{d\mathbf{s}}{dt} = \dot{\mathbf{s}} = \mathbf{F}$ thus generates a trajectory confined to the hypercube $\mathcal{H}_N = [-1, 1]^N$ with the SAT solutions $\mathbf{s}^*$ all located in its corners. To every clause C_m (constraint) we associate the analog clause function $K_m(\mathbf{s}) = 2^{-k} \prod_{j=1}^N (1 - c_{mj}s_j)$, where $c_{mj} = 1$ (-1) if variable x_i appears in normal (negated) form in clause C_m , and $c_{mj} = 0$ if it is missing (in either form) from C_m . The normalization 2^{-k} ensures that $K_m \in [0, 1]$. One can easily check that $K_m = 0$ in a corner $\mathbf{s}$ of $\mathcal{H}_N$ if and only if (iff) clause C_m is satisfied at $\mathbf{s}$. We then introduce a “potential energy” function V that depends on the K_m -s such that $V = 0$ iff all the clauses are satisfied, that is, $K_m = 0$, $\forall m = 1, \dots, M$:

$$V(\mathbf{s}, \mathbf{a}) = \sum_{m=1}^M a_m K_m(\mathbf{s})^2. \quad (1)$$

Here, the a_m are time-dependent, positive weights, or auxiliary variables, $a_m(t) > 0$, $\forall m = 1, \dots, M$, $\forall t \geq 0$. If they were constants, the dynamics would easily get stuck in nonsolution attractors. To prevent that, the dynamics of the auxiliary variables a_m is coupled with the evolution of the clause functions K_m . The dynamics of the full system is defined via

$$\begin{cases} \dot{s}_i = \frac{ds_i}{dt} = -(\nabla_{s_i} V)_i = -\frac{\partial}{\partial s_i} V(\mathbf{s}, \mathbf{a}) = \sum_{m=1}^M 2a_m c_{mi} K_m(\mathbf{s}) K_m, i = 1, \dots, N \\ \dot{a}_m = \frac{da_m}{dt} = a_m K_m, m = 1, \dots, M, \end{cases} \quad (2)$$

where $K_{mi} = 2^{-k} \prod_{j=1, j \neq i}^N (1 - c_{mj}s_j)$. Note that (2) is just a gradient descent in $\mathbf{s}$ -space on V , $\dot{\mathbf{s}} = -\nabla_{\mathbf{s}} V$. For hard (but satisfiable) SAT formulas the dynamics is transiently chaotic, but

eventually all trajectories converge to a solution. Since the dynamics is hyperbolic⁴², the probability $p(t)$ of a trajectory not finding a solution by analog time t decreases exponentially: $p(t) \sim e^{-\kappa t}$. The decay rate κ is an invariant of transient chaos, called the escape rate^{55,56}, and it characterizes the hardness of the given SAT formula/instance.

Next, we introduce a modified version of the above solver to solve MaxSAT. Note that if the global optimum $\mathbf{s}^*$ is not a solution with $V = 0$ (a true MaxSAT problem), then V will keep changing in time as function of the auxiliary variables. The dynamics is still biased to flow toward the orthants of $\mathcal{H}_N$ with low energy, and as shown, in Fig. 1a, it will find the global optimum, but it will never halt there. Naturally, the question arises: how do we know when we have hit an optimal assignment? For that we use a heuristic based on a statistical approach: we start many (relatively short) trajectories from random initial conditions, look for the lowest energy found by each trajectory and then exploit this statistic to help predict the lowest energy state and the time needed to get there by the solver.

However, Eqs. (1) and (2) cannot directly be applied to MaxSAT problems, one needs to modify the potential energy function, first. To see why, notice that the potential V in the center of $\mathcal{H}_N$, in $\mathbf{s} = \mathbf{0}$, is always $V(\mathbf{0}, \mathbf{a}) = 2^{-2k} \sum_{m=1}^M a_m$, because $K_m(\mathbf{0}) = 2^{-k}$, $\forall m$. On the other hand, in a corner $\mathbf{s}'$ of the hypercube, where $|s_i| = 1 \forall i$, the value of each $K_m(\mathbf{s}')$ clause function is 0 if the clause is satisfied or 1 if it is unsatisfied, so the potential V in a corner is just the sum of auxiliary variables corresponding to the unsatisfied clauses, i.e., $V(\mathbf{s}', \mathbf{a}) = \sum_{\{m: K_m \neq 0\}} a_m$. Let a be the average value of the auxiliary variables in a given time instance t , $a = \frac{1}{M} \sum_{m=1}^M a_m$. Thus $V(\mathbf{0}, \mathbf{a}) = 2^{-2k} aM$ and $V(\mathbf{s}', \mathbf{a}) \simeq aE(\mathbf{s}')$, where we introduced $E(\mathbf{x})$ to denote the number of unsatisfied clauses for an assignment $\mathbf{x}$, which we will call “energy”, from here on. If $\mathbf{s}'$ is the global optimum and its energy is large enough (typically at large constraint densities $\alpha = M/N$), the center of the hypercube may

have a smaller potential energy value (due to the 2^{-2k} factor), than any of the corners of the hypercube, and it may become a stable attractor, trapping the dynamics. Figure 1 shows an example of this trapping on a small MaxSAT problem with $N = 10$ variables and $M = 80$ clauses, given in Supplementary Data 1. To prevent this, we need to modify the potential energy function. We do this by adding a term $V'(\mathbf{s}, \mathbf{a})$ to $V(\mathbf{s}, \mathbf{a})$ such that it satisfies the following conditions: (1) it is symmetric in all s_i so that there is no bias introduced in the search dynamics, (2) the energy in $\mathbf{s} = \mathbf{0}$ is always sufficiently large so that it never becomes an attractor, (3) the added term does not modify the energy in the corners of the hypercube, and (4) similarly to the original dynamics, $\mathbf{s}$ always stays within the hypercube $\mathcal{H}_N$, which demands that $\partial V'/\partial s_i = 0$ for all i along the boundary of $\mathcal{H}_N$. We may imagine this added term in the form of a “hat” function: it has a maximum at $\mathbf{s} = \mathbf{0}$ that keeps growing together with the time-dependent auxiliary variables (never to become permanently smaller than the potential energy in the global optimum), but vanishing at the boundary surface of the hypercube.

There are several possibilities for such terms, here we focus on one version that works well in simulations:

$$V(\mathbf{s}, \mathbf{a}) = \sum_{m=1}^M a_m K_m(\mathbf{s})^2 + b\alpha a \sum_{i=1}^N \cos^2\left(\frac{\pi}{2}s_i\right), \quad (3)$$

where a is the average value of the auxiliary variables, $\alpha = M/N$ is the constraint density and b is a constant factor tuning the strength of the last term to be always larger than the first, when chosen properly. The sum with the $\cos^2(\pi s_i/2)$ terms ensures the symmetric hat form, vanishing in the corners of $\mathcal{H}_N$. Note that the first term on the rhs of (3) is never larger than aM . We now have $V(\mathbf{0}, \mathbf{a}) = (2^{-2k} + b)\alpha aM$ and b can be chosen such as to avoid the trapping phenomenon by the origin as described above, see also Fig. 1b. To do that, we simply demand that the potential in the origin $V(\mathbf{0}, \mathbf{a})$ keeps growing approximately at the same rate as the potentials in the corners of the hypercube, never getting smaller than the potential in the global minimum (the smallest potential value in the corners). Thus, as long as $V(\mathbf{0}, \mathbf{a}) \geq V(\mathbf{s}', \mathbf{a})$, where $\mathbf{s}'$ is some corner of the hypercube accessed by the dynamics, the dynamics will not get attracted by the origin of the hypercube. Since $V(\mathbf{s}', \mathbf{a}) \simeq aE(\mathbf{s}')$, this implies that $b \geq \frac{1}{M}E(\mathbf{s}') - 2^{-2k}$, where $E(\mathbf{s}')/M$ is the fraction of unsatisfied clauses in $\mathbf{s}'$. Clearly, the b value can be chosen arbitrarily large, however, if it is too large, then it forces the dynamics to keep running close to the boundary of the hypercube, somewhat lowering its performance. In practice, an $E' = E(\mathbf{s}')$ is easily found by running a trajectory with a sufficiently large b value for some short time, then resetting $b \geq \frac{E'}{M} - 2^{-2k}$. If chosen this way, the search dynamics would not be too sensitive to this parameter b . The new dynamical system is therefore:

$$\begin{cases} \dot{s}_i = -\frac{\partial V}{\partial s_i} = \sum_{m=1}^M 2a_m c_{mi} K_{mi}(\mathbf{s}) K_m(\mathbf{s}) + \frac{\pi}{2} b \alpha a \sin(\pi s_i), \forall i = 1, \dots, N \\ \dot{a}_m = a_m K_m, \forall m = 1, \dots, m. \end{cases} \quad (4)$$

Figure 1 illustrates the difference between the two dynamics (see also Supplementary Fig. 1). While for $b = 0$ (original system) the dynamics converges rapidly to $\mathbf{s} = \mathbf{0}$ (seen, e.g., by monitoring the radius $R^2 = \sum_i s_i^2 \rightarrow 0$), the modified system with a properly chosen $b > 0$ continues the search. It finds an orthant with the minimum energy quite quickly (by $t = 10$), but it does not halt there, it continues the dynamics and returns to this minimum repeatedly (e.g., around $t \approx 16, 22, 41$). Figure 1d–g shows the potential energy function landscape $V(\mathbf{s}, \mathbf{a})$ in the (s_1, s_2) plane.

An energy-dependent escape rate. The escape rate is an invariant measure of the dynamics introduced for characterizing transiently chaotic systems^{55,56}. In a transiently chaotic system the asymptotic dynamics is not chaotic, but, for example, settles onto a simple attractor, or escapes to infinity (in open systems), however, the nonasymptotic dynamics is chaotic, usually governed by a chaotic repeller. It is well known that for hyperbolic, transiently chaotic dynamical systems the probability of a randomly started trajectory not converging to an attractor by time t (i.e., not finding a SAT solution in our case) decreases exponentially in time: $p(t) \sim e^{-\kappa t}$, where κ is the escape rate^{45,55}. The escape rate can also be interpreted as the inverse of the average lifetime τ of trajectories $\kappa = 1/\tau$. For permanently chaotic systems, such as our MaxSAT solver, however, this definition does not work, as there is no simple asymptotic attractor in the dynamics and the system is closed. To be able to use a similar notion also for MaxSAT, we use a thresholding on the energy of the visited states. More precisely, we monitor the probability $p(E, t)$ that a trajectory has not yet found an orthant of energy smaller than E by analog time t . Here, E acts as a parameter of the distribution. This can be measured by starting many trajectories from random initial conditions and monitoring the fraction of those that have not yet found a state with an energy less than E by analog time t . In Fig. 2a we show these distributions for different E values for a MaxSAT problem. For large E , all trajectories almost immediately find orthants with fewer unsatisfied clauses, but for lower E values the distributions decay exponentially. We call their decay rates energy-dependent escape rates $\kappa(E)$. Naturally, if an energy level does not exist in the system (e.g., for $E < E_{\min}$), the escape rate for that energy level is meaningless (extrapolates to zero or a negative number). This suggests that the $\kappa(E)$ dependence could be used to predict where this minimum energy is reached. However, to capture this energy limit, it is more convenient to plot the $E(\kappa)$ function, instead (see Fig. 2b). From extensive simulations, we observe a power-law behavior with an intercept E_0 :

$$E = E_0 + c\kappa^\beta. \quad (5)$$

Since E_0 is not an integer in general, we have $E_{\min} = \lfloor E_0 \rfloor + 1$. This observation is at the basis of our method to predict the global energy minimum for MaxSAT.

Procedure for predicting the global minimum. Here, we describe the algorithm along with the halting criterion for system (4) with details presented in the Methods section along with a flowchart shown in Supplementary Fig. 2. The exponentially decaying nature of the $p(E, t)$ distributions implies that sooner or later every trajectory will visit the orthant with the lowest energy. Nevertheless, instead of leaving one trajectory to run for a very long time, it is more efficient starting many shorter trajectories from random initial conditions and tracking the lowest energy reached by each trajectory (see Supplementary Fig. 3). This also generates good statistics for $p(E, t)$ and for obtaining the properties of the chaotic dynamics that are then exploited along with (5) to predict the value of the global minimum and to decide on the additional number of trajectories needed to find a lower energy state with high probability.

The basic step of the algorithm is to run a trajectory ω from a random initial condition up to a given time $t_{\max}$ and record the lowest energy found by this particular trajectory, denoted by $E_s(\omega)$. Let Γ denote the total number of trajectories run so far, $\mathcal{T}$ the set of these trajectories (thus $\Gamma = |\mathcal{T}|$), and $\bar{E}(\Gamma) = \min_{\omega \in \mathcal{T}} E_s(\omega)$ be the lowest energy found by all these trajectories. Using statistical methods and the relation between energy and escape rate $\kappa(E)$ (shown in (5)), the algorithm repeatedly predicts (as Γ grows) the expected number of trajectories we need to run

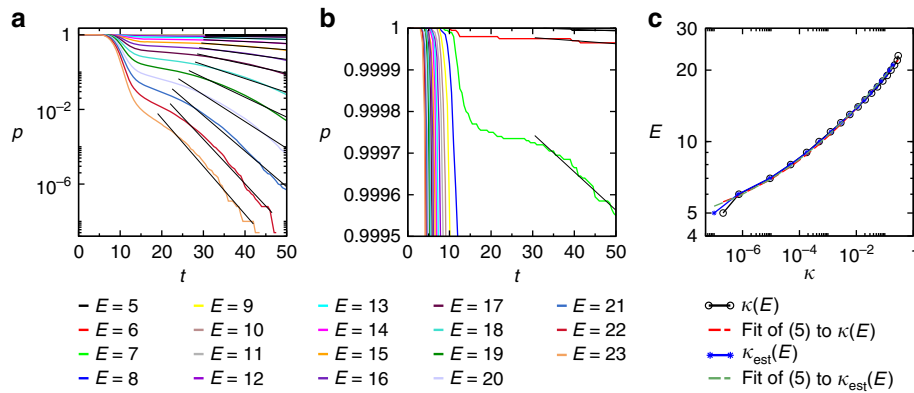


Fig. 2 Energy dependent escape rate. **a** The $p(E, t)$ distributions for a hard, benchmark MaxSAT competition problem with $N = 250$, $M = 1000$ ($\alpha = 4.0$), “HG-3SAT-V250-C1000-1.cnf”, from ref. ⁴⁹. We find $E_{\min} = 5$ with our algorithm after running $\Gamma = 2 \times 10^5$ trajectories with $b = 0.002375$. The escape rates are obtained from fitting $p \sim e^{-\kappa t}$ onto the last section of the distributions (black lines), see Methods. **b** A zoom into the upper section of **(a)**. **c** E vs. the escape rate κ using the values obtained from the fitting shown in **(a)** (black) and using the rough estimate for the escape rates $\kappa_{\text{est}}(E) \simeq -\ln(p(E, t_{\max}))/t_{\max}$ (blue). This estimation is convenient, as it is easier to automate in the algorithm than the fitting procedure (see Methods). The dashed lines show the fitting of Eq. (5) to the $\kappa(E)$ points from **(a)** (red curve: $E = 4.25 + 22.87\kappa^{0.184}$) and to the $\kappa_{\text{est}}(E)$ points (green curve: $E = 4.06 + 22.31\kappa^{0.178}$), respectively. Both curves result in $E_0 \in (4, 5]$ thus predicting the global optimum $E_{\min}^{\text{pred}} = 5$

in total to find the lower energy value $\bar{E} - 1$, i.e., $\Gamma^{\text{pred}}(\bar{E} - 1)$ and the global minimum energy $E_{\min}^{\text{pred}}$. We then monitor $E_{\min}^{\text{pred}}$ for saturation and once the saturation criterion is reached, it outputs a decision $E_{\min}^{\text{dec}}$, representing the final energy value predicted by the algorithm as the global minimum. If this energy value has already been attained (found at least one assignment for it), the algorithm outputs the corresponding assignment(s). If it did not attain it then it keeps running until finds such an assignment or reaches the preset maximum limit $\Gamma_{\max}$ on the number of runs. In the latter case it outputs the lowest energy value attained and the corresponding assignment(s) and the consistency status of the predicted value.

Performance on random Max 3-SAT problems. We first test our algorithm and its prediction power on a large set (in total 4000) of random Max 3-SAT problems with $N = 30, 50, 100$ variables and constraint densities $\alpha = 8, 10$. (In 3-SAT the SAT-UNSAT transition is around $\alpha \simeq 4.267$ ⁵⁷). We compare our results with the true minimum values ($E_{\min}$) provided by the exact algorithm MaxSATZ^{58,59}. In Fig. 3, we compare the lowest energy found by the algorithm $\bar{E}$, the predicted minimum $E_{\min}^{\text{pred}}$ and the final decision by the algorithm $E_{\min}^{\text{dec}}$ with the true optimum $E_{\min}$, by showing the distribution of their deviations from $E_{\min}$ across many random problem instances. We use $t_{\max} = 25$ and at most $\Gamma_{\max} = 150,000$ runs, after which we stop the algorithm even if the prediction is not final. Thus, one expects that the performance of the algorithm decreases as N increases, (e.g., at $N = 100$), so that we would need to run more trajectories to obtain the same performance. Nevertheless, the results show that all three distributions have a large peak at 0. Most errors occur in the prediction phase, but many of these can be significantly reduced through simple decision rules (see Methods), because they occur most of the time at easy/small problems, where the statistics is insufficient (e.g., too few points since there are only few energy values). To show how the error in prediction depends on the hardness of problems, we studied the correlation between the error $E_{\min}^{\text{pred}} - E_{\min}$ and the hardness measure applicable to individual instances $\eta = -\ln \kappa / \ln N$ (see ref. ⁴³), see Fig. 3d (and Supplementary Fig. 4). Interestingly, larger errors occur mainly at the easiest problems with $\eta < 2$. Calculating the Pearson correlation coefficient between $|E_{\min}^{\text{pred}} - E_{\min}|$ and η (excluding instances where the

prediction is correct) we obtain a clear indication that often smaller η (thus for easier problems) generates larger errors. Positive errors are much smaller and shifted toward harder problems. Negative errors mean that the algorithm consistently predicts a slightly lower energy value than the optimum, which is good as this gives an increased assurance that we have found the optimum state. In Supplementary Fig. 4b, we show the correlation coefficients calculated separately for problems with different N and α .

Performance evaluation on hard MaxSAT competition problems. Next, we present the performance of our solver on MaxSAT competition problems, from 2016⁴⁹. We are foremost interested if Max-CTDS is capable of predicting the global minimum energy value and finding assignments corresponding to that energy value for hard problems, within a reasonable time.

For illustration purposes, here we first discuss an extremely hard competition problem instance with $N = 250$ variables and $M = 1000$ clauses, called “HG-3SAT-V250-C1000-1.cnf”, which was reposted for several years. This problem was also used in Fig. 2. No complete competition algorithm could solve this problem. The best complete solver in 2016, the CCLS2akms has found a state with energy value 5, but could not prove that it is optimal within the allotted time (30 min). We ran our algorithm on a regular 2012 iMac 21.5, 3.1 GHz, Intel Core i7 computer and it predicted the lowest energy of 5 (unsatisfied clauses), after 21 min 24 s of running time and produced an assignment for it after 9.168 h of running time. The minimum energy prediction was achieved already after $\Gamma = 7000$ trajectories, whereas finding an assignment with this minimum energy took a total of $\Gamma = 189,562$ trajectories to run. The minimum energy assignment corresponding to $E_{\min}^{\text{dec}} = 5$ is provided in Supplementary Data 2. (The problem can be downloaded from the competition site⁴⁹). We ran the complete and exact algorithm, MaxSATZ^{58,59} for over 5 weeks on this problem and the smallest energy it found was $E = 9$. The details of how the Max-CTDS algorithm performs are shown in Fig. 4. Similar figures for other hard problems such as for a 4-SAT problem and a spin-glass problem are shown in Supplementary Figs. 5 and 6.

Figure 5a shows the lowest energy values (+ symbol) for all the 454 random MaxSAT problems of the 2016 competition⁴⁹, obtained by the (incomplete/heuristic) competition solvers, and

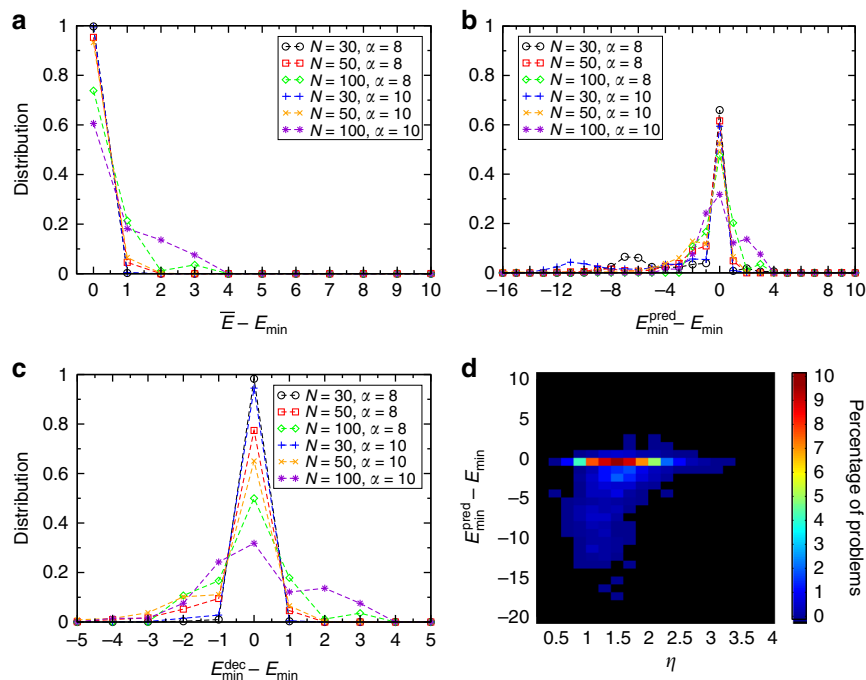


Fig. 3 Algorithm statistics over random Max 3-SAT problems. Distribution of differences between the real global minimum $E_{\min}$ obtained with the exact algorithm MaxSatz and **a** the smallest energy found by the algorithm $\bar{E}$, **b** the predicted minimum value $E_{\min}^{\text{pred}}$, and **c** the final decision of the algorithm $E_{\min}^{\text{dec}}$ shown for problems with different N and α values (see legends). **d** The percentage of instances indicated by color (see color bar) for different values of the error $E_{\min}^{\text{pred}} - E_{\min}$ and hardness η . Most instances are in the $E_{\min}^{\text{pred}} - E_{\min} = 0$ row indicating correct prediction. Large errors occur mainly at smaller η values, and are dominantly negative

the predicted and found lowest energies attained by Max-CTDS (green star and solid red circle symbol, respectively). See Supplementary Fig. 7 for more comparisons with other competition solvers. One can see that Max-CTDS is capable of predicting and achieving the same energy levels as the overall best, for almost all of the high-girth MaxSAT problems “HG3”, “HG4”, and “s3” (Abrame-Habet). There are, however, small deviations in the “s2” group, where Max-CTDS achieves energy levels close to the overall best. These deviations are presented in detail in Fig. 5b, while in Fig. 5c we are showing them normalized by the total number of clauses M , for every problem, indicating that Max-CTDS solved all the problems within 0.8% of the best value. The main reason for the discrepancy in “s2” is the stiffness of the equations⁶⁰, in this category. Since in Max 2-SAT there are only two variables that can satisfy a clause, the problems are more constrained, and additionally, the problems in the “s2” category also have a high minimum energy value (“s2” had also the largest problems with $N \in [120, 200]$, $M \in [1200, 2600]$). This causes the effects of stiffness to appear earlier in the simulations than for the other problems, slowing them down; see Supplementary Note 1 and Supplementary Fig. 8 for a more detailed description. The fact that the Max-CTDS solver still finds very close solutions even for these problems shows, that it has a “smart” search dynamics.

In terms of computation time (wall-clock time) on digital machines (using standard computers), Max-CTDS typically took on the order of hours to find an assignment for the minimum energy value. The average time over all the problems was 4.35 h, with a standard deviation of 5.05 h: in the “s2” category the average time was 10.93h, in “s3”: 0.52h, in “HG3”: 5.69h, and in “HG4”: 0.28 h. The lowest search time was 1.08s, for problem “s3v70c700-5.cnf”, with the best heuristic competition solver taking 0.64 seconds for this problem. The worst search time by Max-CTDS was for a very stiff problem, “s2v120c2500-2.cnf” at 4 days and 14.47h, with the best heuristic solver for this problem

taking only 1.36s. These numbers certainly depend on the digital hardware used. Note that in an analog circuit implementation, the current flow or voltage behavior would correspond to the equations of the solver, eliminating numerical integration issues and thus the algorithm should run much faster (⁴⁶ shows a possible 10^4 speedup).

Application to Ramsey numbers. Ramsey theory deals with the unavoidable appearance of order in large sets of objects partitioned into few classes, with deep implications in many areas of mathematics^{51,61} but also with practical applications⁶². Although it has several variants, in the standard, two-color Ramsey number problem we have to find the order for the smallest complete graph for which no matter how we color its edges with two colors (red and blue), we cannot avoid creating a monochromatic m -clique. The number of nodes for the smallest such complete graph is denoted by $R(m, m)$. The proof that $R(3, 3) = 6$ is trivial. For $m = 4$ the answer is $R(4, 4) = 18$ and it is harder to prove⁶³. The $m = 5$ case is still open, only the bounds $43 \leq R(5, 5) \leq 48$ are known⁵². The best lower bound of 43 was first found in 1989 by Exoo⁶⁴, and the upper bound was only recently reduced from 49⁵³ to 48 by Angeltveit and McKay⁶⁵. Using various heuristic methods, researchers have found in total 656 solutions (328 graphs and their complements) for the complete graph on 42 nodes⁵³. It has been conjectured by McKay and Radziszowski⁵³ that there are no other solutions for $N = 42$. Starting from these solutions they searched for a 5-clique-free coloring in 43. As no solution was found, McKay, Radziszowski, and Exoo make the strong conjecture that $R(5, 5) = 43$ ⁵³.

To tackle Ramsey number problems with our algorithm, we first transform them into k -SAT^{66,67}: every edge i ($i = 1, \dots, N(N-1)/2$) to be colored is represented by a Boolean variable x_i (with $x_i \in \{0, 1\}$, 1 = blue, 0 = red). A clique of size m has $m(m-1)/2$

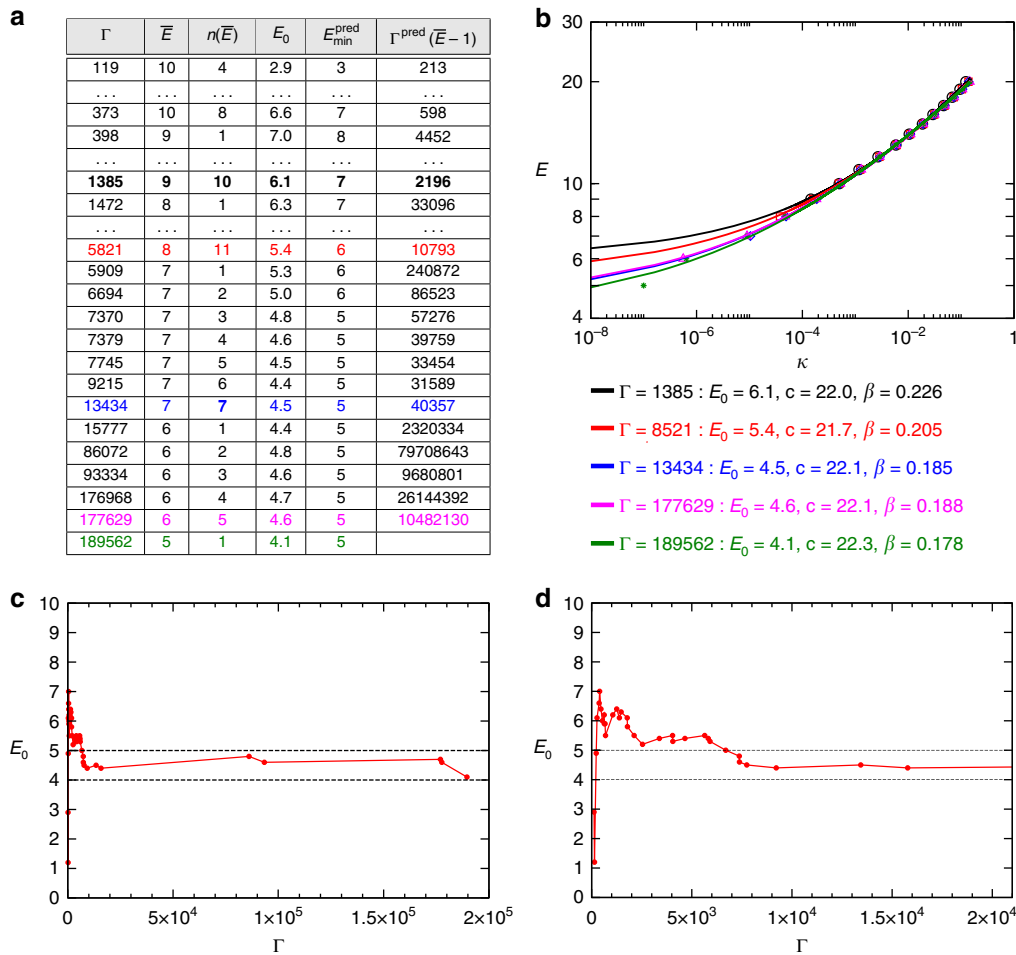


Fig. 4 Algorithm performance on a hard benchmark problem. We use the same problem as in Fig. 2. **a** Γ is the number of trajectories, $\bar{E}$ the lowest energy found until that point, $n(\bar{E})$ is the number of times this energy has been found, E_0 is the parameter obtained from fitting of Eq. (5), $E_{\min}^{\text{pred}}$ and estimating $\Gamma^{\text{pred}}(\bar{E} - 1)$. The algorithm estimates the escape rate and performs a prediction at each Γ shown in the table and for the colored lines we show the fitting curves in **(b)**. **c** The relevant parameter E_0 is shown as function of Γ . While it wildly fluctuates at the beginning when the statistics is small, it remains in the $E_0 \in [4, 5]$ interval, convincingly predicting $E_{\min}^{\text{pred}} = 5$ already after $\Gamma = 7000$ up until the point that it finds this energy at $\Gamma = 189,562$. At this point it could be expected that we do not have a good estimate for $\kappa(5)$ because it has been found only once ($n(5) = 1$), nevertheless the estimation $E_{\min}^{\text{pred}}$ remains consistently the same, convincing our algorithm to accept $E_{\min}^{\text{dec}} = 5$ and stop. **d** A zoom into the $[0, 2 \times 10^4]$ interval of **(c)**

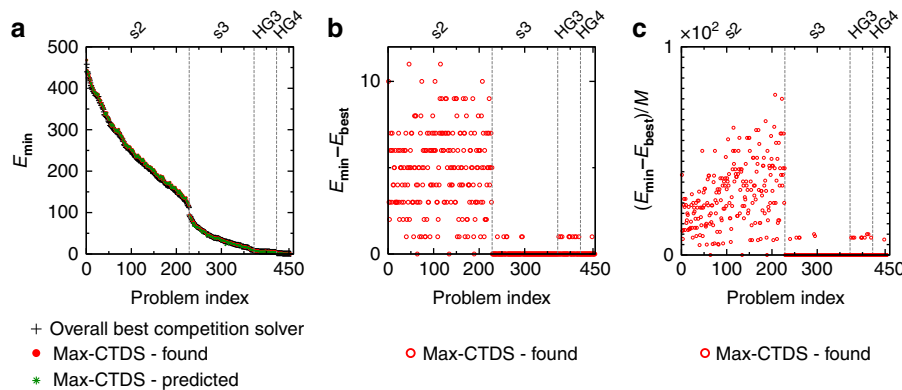


Fig. 5 Algorithm performance on competition MaxSAT problems. **a** Using the 454 random benchmark problems from the 2016 SAT competition⁴⁹, we compared the overall best (lowest) energy found by the competition solvers (black + symbols) and the minimum found/predicted by our Max-CTDS. There are four categories of problems, separated by vertical dashed lines: “s2” (Abrame-Habet) are Max 2-SAT with $N \in [100, 200]$, $M \in [1200, 2600]$; “s3” (Abrame-Habet) are Max 3-SAT with $N \in [70, 110]$, $M \in [700, 1500]$; “HG3” (high-girth) are Max 3-SAT problems with $N \in [250, 300]$, $M \in [1000, 1200]$, and “HG4” are Max 4-SAT problems with $N \in [100, 150]$, $M \in [900, 1350]$. **b** There are some small energy differences especially in the “s2” category, where the ODEs are more stiff and the integration becomes very slow. **c** The energy differences normalized by the number of clauses, $(E_{\min} - E_{\text{best}})/M$. Max-CTDS solves all problems within 0.8% of the best energy value

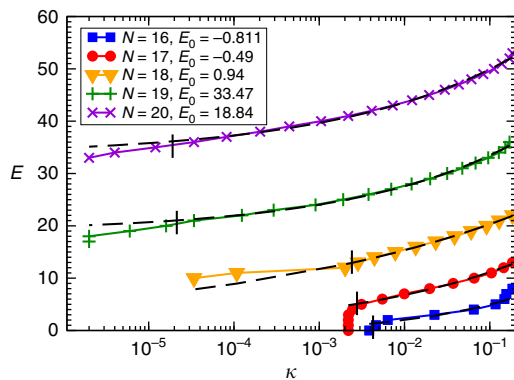


Fig. 6 Finding the Ramsey number $R(4, 4)$. The $E(\kappa)$ relationship for the 6-SAT problems corresponding to the K_N complete graph colorings with two colors. E_0 is the extrapolated value based on the fit from Eq. (5) (dashed lines). The long vertical bars indicate the lower end of the fitting range. Note that for $N=16, 17$, E_0 is a negative value indicating full colorability (the corresponding 6-SAT problem is fully satisfiable), whereas for $N \geq 18$, $E_0 > 0$, and thus the 6-SAT problem becomes MaxSAT

edges. We are satisfied with a coloring (a solution) when no m -clique is monochromatic, i.e., every m -clique with set of edges $\{i_1, \dots, i_{m(m-1)/2}\}$ must have both colors, expressed as the statement formed by the conjunction of the two clauses

$$\left(x_{i_1} \vee \dots \vee x_{i_{m(m-1)/2}}\right) \wedge \left(\bar{x}_{i_1} \vee \dots \vee \bar{x}_{i_{m(m-1)/2}}\right) \quad (6)$$

being true. This means that for every m -clique we have two clauses and thus there are a total of $2^{\binom{N}{m}}$ clauses to satisfy. Since the number of clauses ($O(N^m)$) for $m \geq 2$ grows faster in N than the number of variables N , there will be a lowest N value corresponding to UNSAT, which is the sought $R(m, m)$ Ramsey number. Thus, for $m=3$ we have a 3-SAT problem, for $m=4$ a 6-SAT problem and for $m=5$ a 10-SAT problem. For graphs with $N=42$ nodes the number of clauses is $2 \times \binom{42}{5} = 1,701,336$ and the search space has $2^{\binom{42}{2}} = 2^{861} \simeq 1.5 \times 10^{259}$ colorings. If we were to compute the familiar constraint density α , it would be $\alpha = 2^{\binom{42}{5}} / \binom{42}{2} = 1976$, indeed above the SAT/UNSAT transition point for random 10-SAT, which is estimated to be $\alpha_{10\text{-SAT}} \simeq 707^{68}$.

Applying our algorithm for the $m=4$ Ramsey problem, we can easily find coloring solutions for $N \leq 17$, while for $N=18$ it predicts that there is no solution, indeed confirming that $R(4, 4) = 18$. This is seen from the plot of E vs. κ in Fig. 6. For $N \leq 17$ the smooth portion of the curve fitted by (5) suddenly cuts off, κ being the same for all energy values lower than a threshold value, meaning that after reaching a state corresponding to the threshold energy level, the solution (i.e., $E=0$) is immediately found. This is simply due to the fact that (5) is a statistical average behavior characteristic of the chaotic trajectory, from the neighborhood of the chaotic repeller of the dynamics and away from the region in which the solution resides. However, once the trajectory enters the basin of attraction and nears the solution, the dynamics becomes simple, nonchaotic, and runs into the solution, reflected by the sudden drop in energy. This is not due to statistical errors, because the curve remains consistent when plotting it using 10^3 , 10^4 , or 10^5 initial conditions (the figure shows 10^5 initial conditions).

Searching for the value of $R(5, 5)$ one can relatively easily find coloring solutions without 5-cliques up to $N=35$ for which the number of variables is 595 and the number of clauses 649,264, already huge for a 10-SAT problem for other types of SAT and

MaxSAT solvers. To find solutions faster for $N \geq 36$, however, we employ a strategy based on circulant matrices⁶⁹ helping us find solutions (proper colorings) up to and including $N=42$ in a relatively short time (on the order of hours), see the description in the Methods. This approach places the trajectories relatively close to the solution and a proper coloring can be found in hours even for $N=42$, (see Fig. 7a, b, and Supplementary Data 3 for an easily readable list of edge colorings), for which other heuristic algorithms take many days of computational time⁵³, even with the circulant matrix strategy. Applying the same strategy for $N=43$ we did not find any complete coloring solutions, however, we did find a coloring that creates only two (out of $\binom{43}{5} = 962,598$ possible) monochromatic 5-cliques, see Fig. 7c, d, and the specific coloring provided in Supplementary Data 4.

Discussion

In summary, we presented a continuous-time dynamical system approach to solve a quintessential discrete optimization problem, MaxSAT. The solver is based on a deterministic set of ordinary differential equations and a heuristic method that is used to predict the likelihood that the optimal solution has been found by analog time t . The prediction part of the algorithm exploits the statistics of the ensemble of trajectories started from random initial conditions, by introducing the notion of energy-dependent escape rate and extrapolating this dependence to predict both the minimum energy value (lowest number of unsatisfied clauses) and the expected time needed by the algorithm to reach that value. This statistical analysis is very simple; it is quite possible that more sophisticated methods can be used to better predict minima values and time lengths. Due to its general character, the presented approach can be extended to other optimization problems as well, to be presented in forthcoming publications.

Instead of a numerical implementation on a digital computer, one would ideally like to use a direct implementation by analog circuits, the feasibility of which has been shown in ref. ⁴⁶, as it promises to be a faster (by orders of magnitude) and more efficient approach. One reason for this is that in such analog circuits the von Neumann bottleneck is eliminated, with the circuit itself serving its own processor and memory, see ref. ⁴⁶ for details. Implementation on a digital computer, however, (as it was done here) requires the use of an ODE integrator algorithm, which discretizes the continuous-time equations and evolves them step by step, while controlling for errors. Note that in this case the continuous time variable t is also simulated and evolved in steps $t \rightarrow t + \Delta t$. The time-cost of the dynamics in this case is the wall-clock time (not t), which also depends on the computer hardware and the numerical integration method used. However, in a physical implementation, the t variable would be the real time-cost of the “computation”. In digitized form, the solver is not performing better than current MaxSAT competition solvers simply because the dynamics evolves many (several thousands or more) coupled ODEs, and this integration is time consuming on digital machines. Additionally, to manage the occasional stiffness of the differential equations, one needs to use implicit or higher-order integration methods, also contributing to the slowing down of the simulations. Note that this would not be an issue for analog circuit implementations, as there are no discretization schemes or numerical integration methods; the physical system evolves its currents and voltages according to the ODEs, flowing toward a halting condition, solving the problem. Nevertheless, even when simulated on a digital machine, the solver finds very good solutions to hard problems in reasonable time. This is because continuous-time analog dynamical systems represent an entirely novel family of solvers and search dynamics, and for this reason

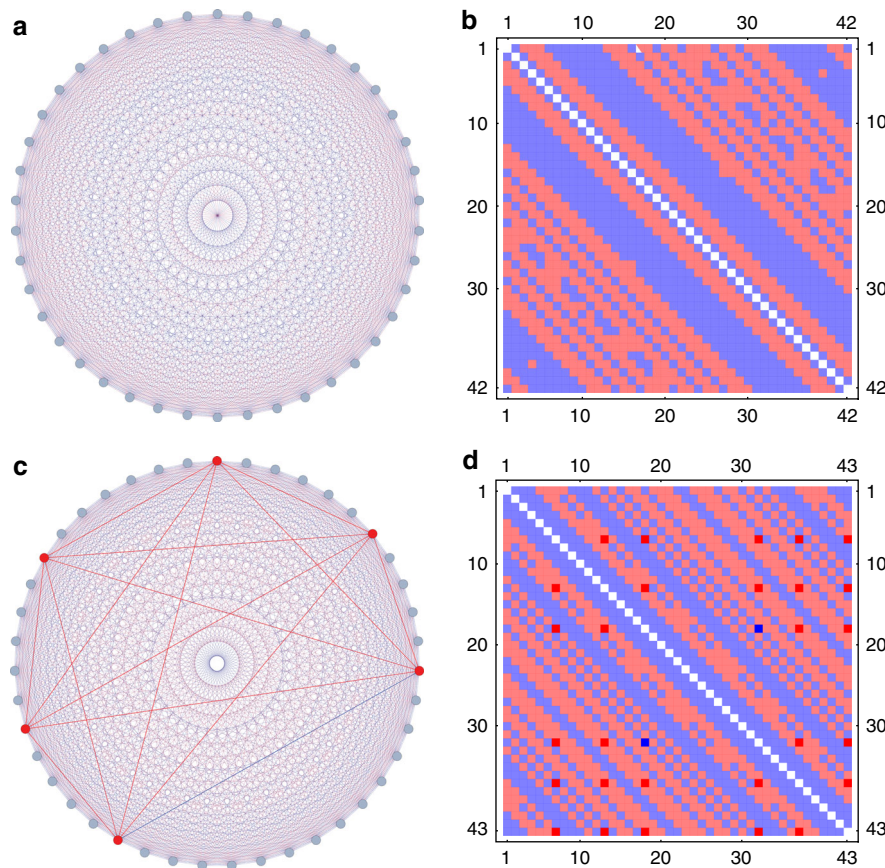


Fig. 7 Colorings for the $R(5, 5)$ Ramsey number problem. **a** A coloring of the complete graph on $N = 42$ nodes that avoids monochromatic 5-cliques. **b** The adjacency matrix corresponding to the coloring in **(a)**, using the same colors. **c** The best coloring of the complete graph on $N = 43$ nodes containing only 2 monochromatic (red) 5-cliques, sitting on 6 nodes (highlighted with thicker edges). **d** The adjacency matrix corresponding to the coloring in **(c)**, using the same colors. The thicker red (blue) edges from **(c)** are represented with darker red (blue) cells. Supplementary Fig. 9 shows a reordered version of this matrix such that the 5-cliques can be seen in the upper left corner of the matrix

they behave differently and thus may perform better, than existing algorithms on certain classes of hard problems.

It is also important to note that the system (4) is not unique, other ODEs can be designed with similar or even better properties. This is useful, because the form given in (4) is not readily amenable to simple hardware implementations, due to the constantly growing auxiliary variable dynamics (all variables represent a physical characteristic such as a voltage or a current and thus they will have to have an upper limit value for a given device). However, the auxiliary variables do not need to grow always exponentially, one can devise other variants in which they grow exponentially as needed, otherwise they can decay (to be presented in a future publication), allowing for better hardware implementations.

To illustrate the effectiveness of our solver, we applied it to the famous two-color Ramsey problem and in particular for $R(5, 5)$, which is still open. We have shown, that the two-color Ramsey problem avoiding monochromatic m -cliques can be translated into an $\frac{m(m-1)}{2}$ -SAT problem and thus a 10-SAT for $m = 5$. Note that digital SAT solving algorithms focus on 3-SAT or 4-SAT problems, and usually are unable to handle directly the much harder 10-SAT. Our solver when run on the corresponding 10-SAT problem was able to find colorings of the complete graph of order 42 without monochromatic 5-cliques, and a coloring with only two monochromatic 5-cliques on 6 nodes for the complete graph on 43 vertices (colorings in the literature for $N = 43$ quote 500+ monochromatic 5-cliques⁷⁰). Note that after posting our paper to arxiv, Geoffrey Exoo in a private communication

mentioned that he also found the same, smallest energy coloring as presented here. This adds further support to the conjecture that $R(5, 5) = 43$.

Methods

SAT/MaxSAT definitions. Boolean satisfiability in conjunctive normal form (CNF) is a constraint satisfaction problem formulated on N Boolean variables $x_i \in \{0, 1\}$, $i = 1, \dots, N$ and M clauses $C_1, \dots, C_M$. A clause is the disjunction (OR operation) of a set of literals, a literal being either the normal (x_i) or the negated (NOT) form ($\bar{x}_i$) of a variable, an example clause being: $C_4 = (x_9 \vee \bar{x}_{10} \vee x_{27})$. The task is to find an assignment for the variables such that all clauses are satisfied, or alternatively, the conjunctive formula $\mathcal{F} = C_1 \wedge \dots \wedge C_M$ evaluates to 1 (TRUE). If all clauses contain exactly k literals, the problem is k -SAT. For $k \geq 3$ this is an NP-complete decision problem³⁵, meaning that a candidate solution is easily (polynomial) checked for satisfiability, but finding a solution can be hard (exp-time). Oftentimes, when studying the performance of algorithms over sets of randomly chosen problems the constraint density $\alpha = M/N$ is used as a statistical guide to problem hardness^{71,72}. MaxSAT has the same formulation as SAT, but the task is to maximize the number of satisfied clauses. For both SAT and MaxSAT, all known algorithms require exponentially many computational steps (in N) in the worst case, to find a solution. However, guaranteeing optimality of solution for MaxSAT is as hard as finding the solution itself, and thus MaxSAT is harder than SAT (NP-hard). Max 2-SAT (i.e., $k = 2$) is already NP-hard.

Properties of the analog SAT solver. Eq. (2) preserves the positivity of the auxiliary variables at all times, since $K_m \geq 0$. According to (2), the auxiliary variables grow exponentially whenever the corresponding clause functions are not satisfied, however, once $K_m = 0$, $\dot{a}_m = 0$, they stop growing. Eq. (2) ensures that whenever the dynamics would get stuck in a local, nonsolution minimum of V , the exponential acceleration changes the shape of V such as to eliminate that local minimum. This can be seen by first solving formally (2): $a_m(t) = a_{m0} \exp\left(\int_0^t d\tau K_m(s(\tau))\right)$, then

inserting it into (1): $V = \sum_{m=1}^M a_m e^{\int_0^{t_{\max}} dt K_m} K_m^2$. Due to the exponentially growing weights, the changes in V are dominated by the clause that was unsatisfied the longest. Keeping only that term in V and inserting it into (2), it is easily seen that the dynamics drives the corresponding clause function toward zero exponentially fast, until another clause function takes over. This is repeated until all clauses are satisfied, for solvable SAT problems. The properties and performance of this solver have been discussed in previous publications^{42–44}. In⁴³ we show that the notion of escape rate κ can be used to characterize the hardness of individual problem instances. We demonstrate this on Sudoku puzzles (all Sudoku problems can easily be translated into SAT), showing that $\eta = -\log_{10} \kappa$ indeed provides a good hardness measure that also correlates well with human ratings of puzzle hardness.

Algorithm description. Here, we give a simple, nonoptimized variant of the algorithm (see flowchart in Supplementary Fig. 2). Better implementations can be devised, for example with better fitting routines, however the description below is easier to follow and works well. Given a SAT problem, we first determine the b parameter as described previously. Step 1: initially we set $\bar{E} = M$, $\Gamma_{\min}$, $\Gamma_{\max} \gg \Gamma_{\min}$, $\Gamma^{\text{pred}}(\bar{E} - 1) = \Gamma_{\min} + 1$ and $t_{\max}$. Unless specified otherwise, in our simulations we used $\Gamma_{\min} = 100$, $\Gamma_{\max} = 2 \times 10^6$, $t_{\max} = 50$. Step 2: to initialize our statistics, we run $\Gamma_{\min}$ trajectories up to $t_{\max}$, each from a random initial condition. For every such trajectory ω we update the $p(E, t)$ distributions as function of the energies of the orthants visited by ω . We record the lowest energy value found $\bar{E}(\Gamma_{\min})$. Step 3: starting from $\Gamma = \Gamma_{\min} + 1$ and up to $\Gamma_{\max}$, we continue running trajectories in the same way and for each one of them check: (a) If $E_s \leq \bar{E}$, set $\bar{E}_{\min} := (E_s, \bar{E})$, update $p(E, t)$ and go to Step 4. (b) If Γ just reached $\Gamma^{\text{pred}}(\bar{E} - 1)$, go to Step 4. (c) If $\Gamma = \Gamma_{\max}$, output “Maximum number of steps reached, increase $\Gamma_{\max}$ ”, output the lowest energy value found, the predicted $E_{\min}^{\text{pred}}$ and the quality of fit for $E_{\min}^{\text{pred}}$, then halt. Step 4: using the $p(E, t)$ distributions, estimate the escape rates $\kappa(E)$ as described in the corresponding Methods section. Step 5: the $\kappa(E)$ curve is extrapolated to the $E - 1$ value obtaining $\kappa(E - 1)$ and then using this we predict $\Gamma^{\text{pred}}(\bar{E} - 1)$ (as described in another Methods section). Further extrapolating the $\kappa(E)$ curve to $\kappa = 0$ we obtain $E_{\min}^{\text{pred}}$ (see the corresponding Methods section). Step 6: we check the consistency of the prediction defined here as saturation of the predicted values. We call it consistent, if $E_{\min}^{\text{pred}}$ has not changed during the last 5 predictions. If it is not consistent yet, we continue running new trajectories (Step 4). If the prediction is consistent, we check for the following halting conditions: (i) If $E_{\min}^{\text{pred}} = \bar{E}(\Gamma)$ then we decide the global optimum has been found: $E_{\min}^{\text{dec}} = E_{\min}^{\text{pred}} = \bar{E}(\Gamma)$ and skip to Step 7. (ii) If the fitting is consistently predicting $E_{\min}^{\text{pred}} > \bar{E}(\Gamma)$ (usually it is very close, $\bar{E}(\Gamma) + 1$) we check the number of trajectories that has attained states with $\bar{E}(\Gamma)$, i.e., $n(\bar{E}) = [1 - p(\bar{E}(\Gamma), t_{\max})]\Gamma$. If it is large enough (e.g. > 100), we decide to stop running new trajectories and set $E_{\min}^{\text{dec}} = \bar{E}(\Gamma)$ and go to Step 7. (iii) If $E_{\min}^{\text{pred}} < \bar{E}(\Gamma)$ then we most probably have not found the global optimum yet and we go to Step 4. We added additional stopping conditions that can shorten the algorithm in case of easy problems, see Methods corresponding section, but these are not so relevant. Step 7: the algorithm ends and outputs $E_{\min}^{\text{pred}}$, $E_{\min}^{\text{dec}}$, $\bar{E}$ values, the Boolean variables corresponding to the optimal state found, along with the quality of fit.

Estimation of the escape rates. As seen in Fig. 2 and Supplementary Fig. 1 the exponential decay of the $p(E, t)$ distribution settles in after a short transient period. Theoretically the escape rate can be obtained by fitting the exponential on that last part of the curves (Fig. 2a). However, while running the algorithm it would be difficult to automatically estimate the region where the exponential should be fitted. The simple approach that works well is to estimate the escape rates as $\kappa(E) \simeq -\ln(p(E, t_{\max})/t_{\max})$ which practically would correspond to the exponential behavior being valid on the whole $(0, t_{\max})$ interval. Note, the $p(E, t)$ is a cumulative distribution with $p(E, 0) = 1$. Usually this estimation is very close to the fitted values (Fig. 2c), but notice that what matters here is the scaling behavior of the escape rates, and this is quite precisely obtained this way because it simply uses the scaling behavior of the $p(E, t_{\max})$ values, instead of the fittings, which is sensitive to the chosen interval.

Predicting the number of trajectories. After calculating the escape rates $\kappa(E)$ one can estimate the number of expected trajectories needed to find a lower energy value: $\Gamma^{\text{pred}}(\bar{E} - 1)$ as described below. Clearly, $p(E, t_{\max})$ is the probability that a trajectory has not reached the energy level E up to time $t_{\max}$. This means that $1 - p(E, t_{\max})$ is the probability that a trajectory did reach energy E , up to time $t_{\max}$. Running Γ trajectories, thus $n(\bar{E}) = [1 - p(\bar{E}, t_{\max})]\Gamma$ will give the expected number of trajectories that reached energy E . Thus, the expected number of trajectories we need to run in total to find the $\bar{E} - 1$ energy value at least once is:

$$\Gamma^{\text{pred}}(\bar{E} - 1) = \frac{1}{1 - p(t_{\max}, \bar{E} - 1)}. \quad (7)$$

However, no trajectory has reached energy $\bar{E} - 1$ yet, and thus we don't have

$p(\bar{E} - 1, t_{\max})$. Instead, it is computed from $p(\bar{E} - 1, t_{\max}) \simeq e^{-\kappa(\bar{E}-1)t_{\max}}$, after extrapolating the $\kappa(E)$ curve to obtain $\kappa(\bar{E} - 1)$.

Predicting the global optimum. When fitting the curve $E = E_0 + a\kappa^\beta$ on our data points we used the Numerical Recipes implementation⁷³ of the Levenberg–Marquardt nonlinear curve fitting method^{74,75}. This implementation has some weaknesses, one could choose to use other implementations or other methods. Sometimes a three-parameter fitting is too sensitive and does not give good results. Because we do not need a very precise value for E_0 (we just need to find the integer interval it falls into, because $E_{\min}^{\text{pred}} = [E_0] + 1$, $[\cdot]$ meaning the integer part) we perform a series of fittings always fixing E_0 and leaving only 2 unknown parameters (a, β) . For each $E_0 = \bar{E}, \bar{E} - 0.1, \bar{E} - 0.2, \dots$ we then perform a fitting and check the χ^2 error. The fitting with minimal error is chosen as the final E_0 and final fitted curve.

Additional stopping conditions. There are cases usually for easy problems, when the fitting using the form (5) does not work well, but based on certain simple conditions we can trust that the global optimum has been found. For example, if there aren't enough (e.g., less than 5) data points in the $\kappa(E)$ curve fitted (this partly explains why the fitting does not give good prediction), but the lowest energy has already been found many times ($n(\bar{E}) > n_{\max}$, e.g., $n_{\max} = 1000$). This happens for very easy problems. Or, if the fitting is consistently predicting another $E_{\min}^{\text{pred}} \neq \bar{E}$, but $n(\bar{E})$ is very large and $\Gamma > \Gamma^{\text{pred}}(\bar{E} - 1)$, so according to the dynamics, it seems a lower energy should have been found already.

In such cases we exit the algorithm (Step 7) with the decision: $\bar{E} \neq E_{\min}^{\text{pred}}$, $E_{\min}^{\text{dec}} = \bar{E}$.

Circulant matrices for the Ramsey number problem. Kalbfleisch⁶⁹ argued that there should be coloring solutions of complete graphs for the Ramsey problem that can be described with a circulant form adjacency matrix (e.g., all red edges are 0-s, blue edges are 1-s in this matrix), or matrices that are close to such a circulant form. Although there is no formal proof of this statement, one expects this to be true also from the SAT formulation of the Ramsey problem. In the SAT formulation, the clauses have a very high degree of symmetry: all variables participate in the same way (4) in all the clauses, which run over all the possible m -cliques. This observation on symmetry can be exploited, allowing us to do part of the search in a much smaller space than the original space, where all the variables could in principle change independently from one another. More precisely, we first define a MaxSAT problem which has only $N - 1$ variables (instead of the full $N(N - 1)/2$ by choosing, e.g., those associated with the links of the first node: $x_1 = a_{1,2}$, $x_2 = a_{1,3}$, ..., $x_{N-1} = a_{1,N}$ as problem variables (here a_{ij} denotes the adjacency matrix) and defining the variables of the links of the other nodes by the circular permutation of this vector $\mathbf{x}$, to obtain a circulant matrix (e.g., $a_{2,3} = a_{1,2}$, $a_{ij} = a_{i-1, j-1}$). Taking the SAT form of the Ramsey problem we replace the variable of a link a_{ij} with x_{j-i} , thus reducing the number of independent variables from $N(N - 1)/2$ to $N - 1$. The number of clauses will also be reduced, because we can now eliminate the repeated ones. This way we obtain a much smaller MaxSAT problem, on which we apply our solver, and starting from random initial conditions we search for low-energy states, which are relatively easily found. We save the $\mathbf{x}$ vectors (the Boolean values) corresponding to such low-energy circulant matrix states. For $N = 42$ we have found circulant type matrices having only 6, 14, 20, 26, etc. monochromatic 5-cliques, indicating that they may already be close to a solution. After saving these circulant matrix states (with small number of monochromatic 5-cliques) we return to the original 10-SAT problem (with $N(N - 1)/2$ variables, without the symmetry constraint), and start a new trajectory from the corner of the hypercube corresponding to the saved matrix state, but now without symmetry restriction.

Data availability

Datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request. The source code for the analog MaxSAT solver can be accessed at <https://github.com/molnarb14/analog-maxSAT-solver>.

Received: 19 January 2018 Accepted: 26 October 2018

Published online: 19 November 2018

References

- Moore, C. & Mertens, S. *The Nature of Computation* (Oxford University Press, Oxford, UK, 2011).
- Barahona, F. On the computational complexity of Ising spin glass models. *J. Phys. A Math. Gen.* **15**, 3241–3253 (1982).
- Istrail, S. Statistical mechanics, three-dimensionality and NP-completeness: I. Universality of intractability for the partition function of the Ising model

- across non-planar surfaces. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing (STOC00)*, 87–96 (ACM Press, Portland, OR, 2000).
4. Lawler, E. L. et al. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. (Wiley, New York, 1985).
 5. Fraenkel, A. S. Complexity of protein folding. *Bull. Math. Biol.* **55**, 1199–1210 (1993).
 6. Sperschneider, V. *Bioinformatics: Problem Solving Paradigms* (Springer Science & Business Media, Berlin, Germany, 2008).
 7. Wang, J. et al. Segmenting subcellular structures in histology tissue images. In *Biomedical Imaging (ISBI)*, 2015 IEEE 12th International Symposium, 556–559 (IEEE, Brooklyn, NY, 2015).
 8. Garey, M. & Johnson, D. *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co Ltd., New York, NY, 1990).
 9. Wu, Y.-L., Tsukiyama, S. & Marek-Sadowska, M. On computational complexity of a detailed routing problem in two dimensional FPGAs. In *Design Automation of High Performance VLSI Systems. GLSV'94*, 70–75 (IEEE, Notre Dame, IN, 1994).
 10. Dagum, P. & Luby, M. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artif. Intell.* **60**, 141–153 (1993).
 11. Waldrop, M. M. More than Moore. *Nature* **530**, 144–147 (2016).
 12. Kalai, G. The quantum computer puzzle. *Not. AMS* **63**, 508–516 (2016).
 13. Almudever, C. G. et al. The engineering challenges in quantum computing. In *Design, Automation and Test in Europe Conference and Exhibition*, 836–845 (IEEE, Lausanne, Switzerland, 2017).
 14. Tsividis, Y. Not your father's analog computer. *IEEE Spectr.* **55**, 38–43 (2017).
 15. Guo, N. et al. Energy-efficient hybrid analog/digital approximate computation in continuous time. *IEEE J. Solid-State Circuits* **51**, 1514–1524 (2016).
 16. Ulmann, B. *Analog Computing* (Oldenbourg Wissenschaftsverlag, München, Germany, 2013).
 17. Subramaniam, A. A neuromorphic approach to image processing and machine vision. In *2017 Fourth International Conference on Image Information Processing (ICIIP)*, Eds: Gupta, P. K., Singh, A. K., Tyagi, V. & Ghreer, S. T., 1–6 (IEEE proc., Wagnaghat, India, 2017).
 18. Inagaki, T. et al. A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
 19. McMahon, P. L. et al. A fully-programmable 100-spin coherent Ising machine with all-to-all connections. *Science* **354**, 614–617 (2016).
 20. Joshi, S., Kim, C., Ha, S. & Cauwenberghs, G. From algorithms to devices: enabling machine learning through ultra-low-power VLSI mixed-signal array processing. In *Custom Integrated Circuits Conference (CICC)*, 1–9 (IEEE, Austin, TX, 2017).
 21. Mostafa, H., Müller, L. K. & Indiveri, G. An event-based architecture for solving constraint satisfaction problems. *Nat. Commun.* **6**, 8941 (2015).
 22. Parihar, A., Shukla, N., Jerry, M., Datta, S. & Raychowdhury, A. Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Sci. Rep.* **7**, 911 (2017).
 23. Parihar, A., Shukla, N., Datta, S. & Raychowdhury, A. Exploiting synchronization properties of correlated electron devices in a non-Boolean computing fabric for template matching. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **4**, 450–459 (2014).
 24. Neftci, E. O., Augustine, C., Paul, S. & Detorakis, G. Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* **11**, 324 (2017).
 25. Pershin, Y. V. & Di Ventra, M. Practical approach to programmable analog circuits with memristors. *IEEE Trans. Circuits Syst. I Regul. Pap.* **57**, 1857–1864 (2010).
 26. Ahmadyan, S. N. & Vasudevan, S. Duplex: Simultaneous parameter-performance exploration for optimizing analog circuits. In *Computer-Aided Design (ICCAD)*, 1–8 (IEEE, Austin, TX, 2016).
 27. Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I. & Gauthier, D. J. Reservoir computing with a single time-delay autonomous Boolean node. *Phys. Rev. E* **91**, 020801 (2015).
 28. Kumar, S., Strachan, J. P. & Williams, R. S. Chaotic dynamics in nanoscale NbO₂ Mott memristors for analogue computing. *Nature* **548**, 318 (2017).
 29. Achour, S., Sarpeshkar, R. & Rinard, M. C. Configuration synthesis for programmable analog devices with Arco. In *ACM SIGPLAN Notices*, vol. 51, 177–193. Eds: Krintz, C. & Berger, E. (ACM, Santa Barbara, CA, 2016).
 30. Orponen, P. A survey of continuous-time computation theory. In *Advances in Algorithms, Languages, and Complexity*, 209–224. Eds: Du, D.-Z. & Ko, K.-I. (Springer, Boston, MA, 1997).
 31. MacLennan, B. J. Analog computation. In *Encyclopedia of Complexity and Systems Science*, 271–294. Ed: Meyers, R. A. (Springer, New York, NY, 2009).
 32. Bournez, O. & Pouly, A. *Handbook of Computability and Complexity in Analysis*, chapter a survey on analog models of computation (Springer in cooperation with the Association Computability in Europe, 2018, in press).
 33. Bournez, O. & Pouly, A. A Universal Ordinary Differential Equation. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)* vol. 80, 116:1–116:14 (2017).
 34. Kia, B., Lindner, J. F. & Ditto, W. L. A simple nonlinear circuit contains an infinite number of functions. *IEEE Trans. Circuits Syst. II Express Briefs* **63**, 944–948 (2016).
 35. Cook, S. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 151–158 (ACM, Shaker Heights, OH, 1971).
 36. Levin, L. A. Universal sequential search problems. *Probl. Inf. Transm.* **9**, 265–266 (1973).
 37. Battaglia, D., Kolar, M. & Zecchina, R. Minimizing energy below the glass thresholds. *Phys. Rev. E* **70**, 036107 (2004).
 38. Chieu, H. L. & Lee, W. S. Relaxed survey propagation for the weighted maximum satisfiability problem. *J. Artif. Intell. Res.* **36**, 229–266 (2009).
 39. Claessen, K. et al. SAT-solving in practice, with a tutorial example from supervisory control. *Discret Event Dyn. Syst.* **19**, 495 (2009).
 40. Kautz, H. & Selman, B. The state of SAT. *Discret Appl. Math.* **155**, 1514–1524 (2007).
 41. Biere, A., Heule, M. & van Maaren, H. *Handbook of Satisfiability*, vol. 185 (IOS Press, Amsterdam, The Netherlands, 2009).
 42. Ercsey-Ravasz, M. & Toroczkai, Z. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nat. Phys.* **7**, 966–970 (2011).
 43. Ercsey-Ravasz, M. & Toroczkai, Z. The chaos within Sudoku. *Sci. Rep.* **2**, 725 (2012).
 44. Varga, M., Sumi, R., Toroczkai, Z. & Ercsey-Ravasz, M. Order-to-chaos transition in the hardness of random Boolean satisfiability problems. *Phys. Rev. E* **93**, 052211 (2016).
 45. Lai, Y.-C. & Tél, T. *Transient Chaos: Complex Dynamics on Finite-Time Scales* (Springer, New York, NY, 2011).
 46. Yin, X. et al. Efficient analog circuits for Boolean satisfiability. *IEEE Trans. Very Large Scale Integr. Syst.* **26**, 155–167 (2017).
 47. Eén, N. & Sörensson, N. An extensible SAT-solver. In *International Conference on Theory and Applications of Satisfiability Testing*, 502–518. Eds: Giunchiglia, E. & Tacchella, A. (Springer, Santa Margherita Ligure, Italy, 2003).
 48. Van Harmelen, F., Lifschitz, V. & Porter, B. *Handbook of Knowledge Representation*, vol. 1 (Elsevier, Amsterdam, The Netherlands, 2008).
 49. Benchmarks. *Evaluation of Max-SAT Solvers*. <http://www.maxsat.udl.cat/16/benchmarks/index.html> (2016).
 50. Graham, R. & Spencer, J. Ramsey theory. *Sci. Am.* **263**, 112–117 (1990).
 51. Graham, R. L., Rothschild, B. L. & Spencer, J. H. *Ramsey Theory*, vol. 20 (John Wiley & Sons, New York, NY, 1990).
 52. Radziszowski, S. P. Small Ramsey numbers. *The Electronic Journal of Combinatorics*, #DS1, Version 15, 1–104 (Mar 3, 2017).
 53. McKay, B. D. & Radziszowski, S. P. Subgraph counting identities and Ramsey numbers. *J. Comb. Theory, Ser. B* **69**, 193–209 (1997).
 54. Arnold, V. *Ordinary Differential Equations* (MIT Press, New York, NY, 1978).
 55. Kadanoff, L. & Tang, C. Escape rate from strange repellers. *Proc. Natl Acad. Sci.* **81**, 1276 (1984).
 56. Cvitanović, P., Artuso, R., Mainieri, R., Tanner, G. & Vattay, G. *Chaos: Classical and Quantum*. (Niels Bohr Institute, Copenhagen, 2009).
 57. Mézard, M. & Zecchina, R. Random k-satisfiability problem: from an analytic solution to an efficient algorithm. *Phys. Rev. E* **66**, 056126 (2002).
 58. Li, C., Manyà, F. & Planes, J. Detecting disjoint inconsistent subformulas for computing lower bounds for max-SAT. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, 86–91 (AAAI Press, Cambridge, MA, 2006).
 59. Li, C., Manyà, F. & Planes, J. New inference rules for max-SAT. *J. Artif. Intell. Res.* **30**, 321–359 (2007).
 60. Hairer, E. & Wanner, G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd edn. (Springer-Verlag, Berlin, Germany, 1996).
 61. Bollobás, B. *Modern Graph Theory*, vol. 184 (Springer Science & Business Media, New York, NY, 2013).
 62. Roberts, F. S. Applications of Ramsey theory. *Discret Appl. Math.* **9**, 251–261 (1984).
 63. Greenwood, R. E. & Gleason, A. M. Combinatorial relations and chromatic graphs. *Canad. J. Math.* **7**, 7 (1955).
 64. Exoo, G. A lower bound for $R(5, 5)$. *J. Graph Theory* **13**, 97–98 (1989).
 65. Angeltveit, V. & McKay, B. D. $R(5, 5) \leq 48$. Preprint at: <https://arxiv.org/abs/1703.08768> (2017).
 66. Dransfield, M., Marek, V. W. & Truszczyński, M. Satisfiability and computing van der Waerden numbers. In *International Conference on Theory and Applications of Satisfiability Testing*, 1–13. Eds: Giunchiglia, E. & Tacchella, A. (Springer, Santa Margherita Ligure, Italy, 2003).

67. Kullmann, O. Green-Tao numbers and SAT. In *International Conference on Theory and Applications of Satisfiability Testing*, 352–362 (Springer, Edinburgh, UK, 2010).
68. Mertens, S., Mézard, M. & Zecchina, R. Threshold values of random k-SAT from the cavity method. *Random Struct. Algorithms* **28**, 340–373 (2006).
69. Kalbfleisch, J. Construction of special edge-chromatic graphs. *Can. Math. Bull.* **8**, 575–584 (1965).
70. Kunkel, C. J. & Ng, P. Ramsey Numbers: Improving the Bounds of $R(5, 5)$ (2003).
71. Kirkpatrick, S. & Selman, B. Critical-behavior in the satisfiability of random Boolean expressions. *Science* **264**, 1297–1301 (1994).
72. Mezard, M., Parisi, G. & Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science* **297**, 812–815 (2002).
73. Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. *Numerical Recipes in C. The Art of Scientific Computing*, 2nd edn (Cambridge University Press, New York, NY, 1992).
74. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **2**, 164–168 (1944).
75. Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* **11**, 431–441 (1963).

Acknowledgements

We thank to X.S. Hu, S. Datta, Z. Néda, E. Regan, X. Yin, and S. Kharel for useful comments and discussions. We also thank Brendan McKay for his corrections and comments on known Ramsey colorings for $R(5, 5)$. This work was supported in part by grants of the Romanian National Authority for Scientific Research and Innovation CNCS/CCCDI-UEFISCDI, project nrs. PN-III-P2-2.1-BG-2016-0252 (MER), PN-III-P1-1.1-TE-2016-1457 (MER, BM), COFUND-FLAGERA II-CORTICITY (MER), the GSCE-30260-2015 Grant for Supporting Excellent Research of the Babeş-Bolyai University (BM, MER). It was also supported in part by the National Science Foundation under Grants CCF-1644368 and 1640081, and by the Nanoelectronics Research Corporation, a wholly-owned subsidiary of the Semiconductor Research Corporation, through Extremely Energy Efficient Collective Electronics, an SRC-NRI Nanoelectronics Research Initiative under Research Task ID 2698.004 (Z.T., M.V., and F.M.).

Author contributions

B.M., Z.T., and M.E.R. designed the methods and algorithm. B.M., F.M., M.V., and M.E.R. implemented the simulations. Both Z.T. and M.E.R. wrote and reviewed the manuscript.

Additional information

Supplementary Information accompanies this paper at <https://doi.org/10.1038/s41467-018-07327-2>.

Competing interests: The authors declare no competing interests.

Reprints and permission information is available online at <http://npg.nature.com/reprintsandpermissions/>

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2018