Taming a nonconvex landscape with dynamical long-range order: Memcomputing Ising benchmarks

Forrest Sheldon [1,*] Fabio L. Traversa [2,†] and Massimiliano Di Ventra [1,‡]
[1]*Department of Physics, University of California San Diego, La Jolla, California 92093, USA*
[2]*MemComputing, Inc., San Diego, California 92037, USA*

(Received 23 June 2019; published 20 November 2019)

Recent work on quantum annealing has emphasized the role of collective behavior in solving optimization problems. By enabling transitions of clusters of variables, such solvers are able to navigate their state space and locate solutions more efficiently despite having only local connections between elements. However, collective behavior is not exclusive to quantum annealers, and classical solvers that display collective dynamics should also possess an advantage in navigating a nonconvex landscape. Here we give evidence that a benchmark derived from quantum annealing studies is solvable in polynomial time using digital memcomputing machines, which utilize a collection of dynamical components with memory to represent the structure of the underlying optimization problem. To illustrate the role of memory and clarify the structure of these solvers we propose a simple model of these machines that demonstrates the emergence of long-range order. This model, when applied to finding the ground state of the Ising frustrated-loop benchmarks, undergoes a transient phase of avalanches which can span the entire lattice and demonstrates a connection between long-range behavior and their probability of success. These results establish the advantages of computational approaches based on collective dynamics of continuous dynamical systems.

DOI: 10.1103/PhysRevE.100.053311

I. INTRODUCTION

Nonconvex optimization problems draw their difficulty from the complexity of their associated landscapes [1]. These landscapes are often highly corrugated, dotted with hills, valleys, and saddles of varying heights which obscure the search for a lowest (or highest) point. The complexity of this space, combined with the "curse of dimensionality" yields an exponentially large number of potential solutions which are very difficult to prune down by any systematic method. The innate difficulty and variety displayed by optimization problems, as well as their widespread applications have made their study a continuously active field of research across science and mathematics [2,3].

The exponential growth of the state space with problem size often renders any exact algorithm for locating the optimum impractical as they require an exponential amount of time to sift through the states. As a result, practitioners must rely on *incomplete* or *approximate* methods which will often generate better solutions in a limited time but are not guaranteed to converge to the exact solution [4,5]. Despite this, incomplete methods can often converge to the global solution in times orders of magnitude faster than complete solvers [6].

Early work on approximate methods relied on analogies with the dynamics of physical systems [7] which will minimize their energy as they cool, i.e., during annealing. For example, to find the ground state of the Ising spin glass [8],

$$E = -\sum_{\langle ij \rangle} J_{ij} s_i s_j, \quad s_i \in \{-1, 1\}, \tag{1}$$

simulated annealing gradually improves an initial state $\{s_i\}_{i=1}^N$ by stochastically exploring the state space and steadily lowering an effective temperature [9]. The early success of this approach on combinatorial optimization problems has led to the proliferation of solvers based on a similar stochastic local search and their many variants [10,11]. Cross-pollination with physics has continued, spawning methods such as parallel tempering [12] and quantum simulated annealing [13] as well as the analytical characterizations of combinatorial problems [14] and random energy surfaces [15].

Annealing has again jumped to the forefront of modern research in the form of quantum annealing and the machines manufactured by DWave [16–18]. These machines contain two-state quantum mechanical elements coupled together in a graph realizing a particular energy function. During their relaxation, the quantum dynamics of the system allows for collective tunneling of elements through high, thin barriers in the energy function, which may provide some advantage in the search for the optimum.

Similar ideas in the context of cellular automata, neural networks, and neuroscience have received steady interest [19,20]. These examples substantiate the idea that *collective behavior* would offer an advantage in the convergence of a solver by allowing for a more efficient exploration of the state space. We then expect that classical solvers which incorporate

*fsheldon@physics.ucsd.edu
†ftraversa@memcpu.com
‡diventra@physics.ucsd.edu

this feature in their dynamics will have an advantage in both the quality of approximate solutions they produce and their rate of convergence.

We also note that annealers admit cluster flipping variants, which can drastically increase the rate of thermalization and thus provide faster convergence to the ground state. Incorporating proposed cluster flips into an annealer as in the Swendsen-Wang and Wolff algorithms can allow the system to overcome larger energy barriers and reduce correlation times [21,23]. Modern parallel tempering variants incorporating iso-energetic and thus rejection-free cluster moves [22,24] have proven effective at converging to the ground state of spin-glass instances. While here we have chosen to focus on introducing DLRO in continuous systems, a more detailed comparison between the properties of clusters generated by continuous and discrete solvers and their effect on convergence time is a direction of future work.

The purpose of this work is to explore the presence and advantages of collective dynamics in the context of specific deterministic dynamical systems: digital memcomputing machines (DMMs) [25–27]. In DMMs, a combinatorial optimization problem is first transformed into a physical system described by differential equations whose equilibrium points correspond to solutions of the original problem. Theoretical work [28,29] and simulations of DMMs [26,27,30,31] have indicated the presence of long-range order in their dynamics. However, as their native problem form involves several distinct dynamical elements, the complexity of the resulting solver obscures the physical principles underlying its design and function.

Here we first show that this collective behavior, in the form of *dynamical long-range order* (DLRO), allows the efficient solution of a class of benchmarks based on the Ising spin glass (1). Then, by drawing on the structure of the equations governing a DMM, we propose a simplified model that captures several features of their dynamics and illuminates the essential roles played by continuity and memory.

The paper is organized as follows. In Sec. II we briefly introduce the concept of DMMs. In Sec. III we discuss the Ising frustrated-loop instances used for benchmarking several methods and show the results of this benchmark. In Sec. IV we introduce a simplified model of DMMs that captures its main features. In Sec. V we discuss the dynamical long-range order that emerges in these dynamical systems. In Sec. VI we offer our conclusions.

II. DIGITAL MEMCOMPUTING MACHINES

In this section we present a very brief introduction to the concept of DMMs [26]. A thorough discussion of the physics behind them and the problems they have been applied to can be found in the perspective article [27].

DMMs are dynamical systems designed as circuit elements to solve circuit satisfiability (SAT) problems [25–27], so that despite operating in continuous time, initial and final states are digital, and hence the machines are scalable. A particular problem may be translated into circuit SAT format as a combination of AND, OR, and NOT gates, which are then replaced with dynamical circuit elements. The elements are composed of resistors, capacitors, voltage or current generators and resistors with memory (memristors) whose dynamics conspire to lead the circuit voltages to a state where all logical constraints are satisfied.

These elements are governed by ordinary differential equations and constraints imposed by Kirchoff's laws, and thus they may be efficiently simulated. Numerical studies have shown that DMMs are effective at solving a wide range of combinatorial optimization problems [27,30–32]. For instance, DMMs have proven effective at solving integer-linear programming problems (a DMM solver compared favorably to standard algorithms when applied to instances in the MIPLIB 2010 library and was able to establish the feasibility of an instance for which this was previously unknown [32]), and maximum satisfiability (MAXSAT) problems derived from XORSAT where DMMs displayed linear scaling in their approach to a satisfiable threshold while all other solvers tested scaled exponentially [30].

III. FRUSTRATED-LOOP INSTANCE BENCHMARKING

Recent progress on fabricating quantum mechanical hardware has generated renewed interest in problems of the form (1), or as it is known in the optimization community, quadratic unconstrained binary optimization (QUBO), which can be mapped into a MAXSAT problem [33]. When the couplings J_{ij} are assigned randomly, this is known as the Ising spin glass and is a simplified model for the behavior of glassy systems [8]. Studies of the thermodynamic properties of these systems have shown that, below a certain temperature, the phase space can become separated into clusters of states and may admit an exponential number of metastable states. This proliferation of metastable states obscures the search for the global minimum and gives a physical interpretation for the computational hardness of the optimization problem.

The problem of benchmarking MAXSAT solvers is generally hindered by the fact that the problems are NP-hard, and, for an arbitrary instance, even confirming a solution may require exponential time [34]. For this reason, planted solution instances are commonly employed in which instances are generated such that they have a known solution [35].

Benchmarking studies on quantum annealing have introduced the class of frustrated-loop Hamiltonians [16,36] in which the total Hamiltonian is written as the sum of Hamiltonians of a set of loops containing a single frustrated bond (see schematic in Fig. 1),

$$H = \sum_i H_{FL,i}. \tag{2}$$

The loops are formed such that the planted solution minimizes all of the Hamiltonians $H_{FL,i}$ simultaneously and so minimizes their sum.

In order to generate these instances, we first construct an underlying lattice which we take to be hypercubic in D dimensions with periodic boundary conditions. Each loop is generated by beginning at a randomly selected site and performing a random walk until it crosses itself. The length, l, of the loop formed is generally required to be above some limit, otherwise it is rejected. For example, the instances solved on DWave employ a loop length limit of $l \geqslant 8$ [16]. It is also noted that discarding the length limit seems to lead to very

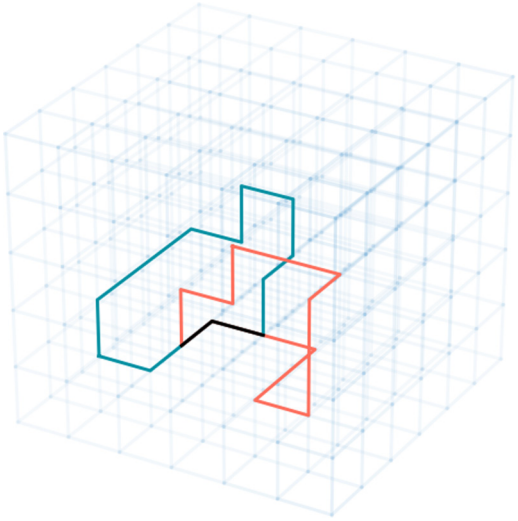

FIG. 1. A schematic representation of instance creation. Separate frustrated loops (blue and red curves) are generated by random walking around the lattice until the walk crosses itself. Each loop has its own Hamiltonian consisting of $J_{ij} = 1$ for all bonds except one with $J_{kl} = -1$ such that the ground state of the loop will have one unsatisfied bond. When the loops are combined, overlapping bonds (shown in black) have a coupling J_{ij} which is the sum of the contributions from each separate loop.

difficult instances, although an explanation for this feature is not given. In our investigations of the instances, we found that discarding the loop length limit leads to instances of widely varying difficulty, and that both the variance and mean of the solution time (measured with simulated annealing) decreased as the length limit increased. In order to avoid the complications of a widely varying difficulty, while generating the most difficult available instances, we then chose a length limit of $l \geqslant 6$ for our generated instances.

In order to generate a loop, we consider planting the ferromagnetic solution $s_i = 1$. After generating an instance, any other solution may be hidden by means of a gauge transformation. All interactions in the loop are chosen to be ferromagnetic, $J_{ij} = 1$, except one which is selected at random to be antiferromagnetic $J_{ij} = -1$. The solution to the loop Hamiltonian $H_{FL,i} = -\sum_{\langle ij\rangle \in l_i} J_{ij} s_i s_j$ is thus an assignment with one unsatisfied interaction.

The number of loops, M, generated must be proportional to the number of sites $N = L^D$ and may be characterized by a *density* α such that $M = \alpha N$. These instances are known to demonstrate a *hardness peak* in α such that the most difficult instances are generated when there are neither too few loops, in which case they do not overlap and each may be solved separately, nor too many, in which case the antiferromagnetic interactions tend to be canceled by the more numerous ferromagnetic interactions [16,37]. The value of α at the peak also tends to align with the amount of frustration in the instance, as measured by the number of unsatisfied interactions in the ground state.

In order to generate difficult instances, in $D = 2$ dimensions we used a simulated annealing solver to test instances across a range of α, finding that the most difficult instances lay at $\alpha \approx 0.2$, consistent with the results on the pseudoplanar chimera graphs in Ref. [16]. For $D = 3$ dimensions, the optimal value of α was estimated using the amount of frustration in the instances as suggested in Ref. [16] and found to lie at $\alpha \approx 0.3$.

Benchmarking was carried out using instances generated on a three-dimensional hypercubic lattice with periodic boundaries. The implementation of the dynamical equations of DMMs as in Ref. [26] was appropriately modified to handle the Ising frustrated loop instances expressed as a maximum satisfiability problem in conjunctive normal form [38] (see the Supplemental Material [39] for a discussion of this transformation). These were then simulated using a commercial sequential MATLAB solver dubbed Falcon provided by MemComputing, Inc. In addition, we have implemented two standard annealing algorithms in Python [simulated annealing (SA) and parallel tempering (PT)], as well as used a well-known commercial mixed-integer programming solver, IBM CPlex [40]. Since Falcon was implemented in interpreted MATLAB and the focus was on scaling rather than runtime, we used only the simplest implementation of each solver but performed substantial tuning. Details of the implementation and tuning on the instance class for SA and PT, as well as the configuration for IBM CPlex, can be found in the Supplemental Material [39].

All solvers were run on frustrated-loop instances in three dimensions, ranging in size from $L = 6$ (total number of spins $N = 216$) to $L = 40$ ($N = 64\,000$). The sizes used for tuning were included for the annealers (SA and PT) while CPlex and Falcon were run on sizes $L \geqslant 10$. Comparisons between solvers are hampered by ambiguity in the efficiency of the implementation and differences in processor speed when run on different machines. In order to skirt the second of these ambiguities we have displayed the solutions in terms of estimated floating point operations (flops) calculated by multiplying solution time by the peak floating point operations rate of a single core. However, we emphasize that when comparing different implementations only differences in *scaling* are relevant.

As is clearly visible from Fig. 2, the memcomputing solver converged to the exact ground state with superior scaling to all solvers tested, allowing us to achieve sizes much larger than possible with other solvers within the time limitation of $\approx 10^6$ sec. The total number of flops appears to scale approximately as $N^{1.5}$ at large sizes, while all other solvers appear to scale exponentially as $\exp(bN^c)$, with b and c solver-specific constants reported in the Supplemental Material [39] and shown in Fig. 2. Details of the fitting procedure as well as the figure displaying the recorded time to solution may be found in the Supplemental Material [39].

IV. A SIMPLIFIED DMM MODEL

As mentioned in Sec. II, a DMM is constructed in correspondence to the logical circuit it will solve. For example, the subset-sum problem studied in Ref. [26] utilizes a circuit with the same structure as one used to add a subset from a group of numbers. Each traditional logic gate is replaced by a *self-organizing logic gate* consisting of a set of interconnected input and output terminals, each of which is dressed with a number of memristors (resistors with memory), resistors,

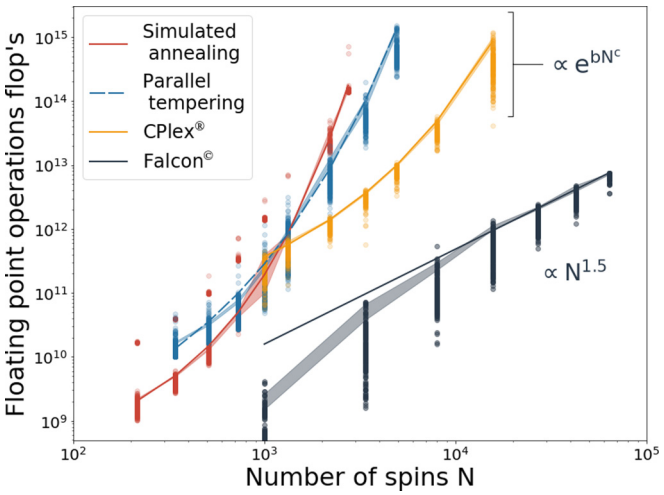

FIG. 2. Scaling of floating point operations necessary for different solvers to reach the ground state of the 3D frustrated-loop instances as a function of the total number of spins N. Solution times have been converted to estimated number of floating point operations. The sequential memcomputing solver implemented in MATLAB is dubbed Falcon and was run on an Intel Xeon 6148. Varying numbers of instances were run at each size and solver depending on required computation time (see the Supplemental Material for details [39]). Comparisons with simulated annealing (SA), parallel tempering (PT), and IBM CPlex run on an Intel Xeon E5430 are also shown. All calculations were performed on a single core. The solid lines are the best fits of the 95th quantile time to solution for all four solvers. The exponential fits have the following parameters: for IBM CPlex, $b = 0.12$ and $c = 0.46$, for SA, $b = 0.069$ and $c = 0.67$, and for PT $b = 0.32$ and $c = 0.46$.

capacitors, and voltage or current generators forming a *dynamic correction module* (DCM) [26]. When voltages are applied to the boundaries of the circuit, the dynamics of these elements are configured to satisfy the constraints enforced by each gate, and lead the circuit to a state where no logical contradictions are present.

However, the restrictions imposed by the native hardware formulation of DMMs necessarily complicate their design and obscure important features of their implementation. Moreover the dependence on a large number of physical parameters in these models makes understanding their dependence on these parameters difficult. As such, an effective model that discards the constraints of hardware and that is more tractable for analysis is desirable. In the following we formulate a simple model that replicates several features of the full implementation of digital memcomputing machines, and we use this to probe the existence of DLRO in the search for a solution.

For our purposes, finding the ground state of the Ising system provides the advantage that it can be expressed in terms of very simple homogeneous constraints leading to a concise set of equations. In addition, its real-space lattice representation allows for a clearer demonstration of DLRO since the real-space distance of the lattice corresponds to the distance in the constraint graph.

We consider the contribution of constraint C to the dynamics of site i (see Fig. 3) [26]. The dynamics of the circuit are constructed such that the voltage generators impose the

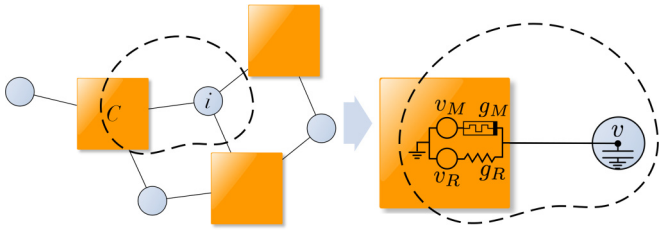

FIG. 3. An arbitrary constraint (C) satisfaction problem expressed as a factor graph can be translated into an electrical circuit with memory by considering the effect of each constraint on the site i. v_R is a voltage generator, and g_R is the conductance of a standard resistor. v_M is a voltage generator, and g_M is the conductance of a resistor with memory.

logical constraint on the voltage v_i at site i. The memristor conductance g_M, sensing a current flowing across it due to an unsatisfied constraint, will alter its value to accelerate the convergence of v_i to the logically consistent solution. Generally, this is accomplished by increasing the memristor conductance, thus allowing more current to flow into or out of the site. As memristors are polar objects, complex constraints may require several memristors and generators to accomplish this, accounting for the number of memristors in DCMs [26].

A few simplifying assumptions give the general form for the contribution of constraint C to site i as [26]

$$\dot{v}_i = \Delta g_M x \Delta V_M + g_R \Delta V_R, \tag{3}$$

$$\dot{x} = h(\Delta V_M, x), \quad x \in [0, 1], \tag{4}$$

for the voltage v_i representing the variable i and the memory state variable of the memristor x. We can regard the first and second terms on the rhs of Eq. (3) as representing the *total* memristive and resistive contributions from the DCM, respectively. These are weighted by the conductances Δg_M and g_R, respectively, into which we have absorbed a capacitive timescale. We regard the memory state variable x and function h in Eq. (4) as an effective representation of the state and evolution of all memristors in the DCM, giving us considerable freedom in choosing the form of h.

These equations bear a close resemblance to those of Lagrange programming neural networks (LPNNs) proposed in Refs. [41,42] and the dynamical systems proposed in Ref. [43]. In these works a Lagrangian, $\mathcal{L}$, for a constraint satisfaction problem on variables $\{s_i\}$ is formed from a set of constraint functions $C_m(\{s_i\})$ which vary from 0 when the constraint is satisfied to 1 when unsatisfied and a set of weights for each constraint x_m, $\mathcal{L} = \sum_m x_m C_m(\{s_i\})$. In the case of LPNNs, the equations of motion of the system are then derived as

$$\dot{s}_i = -\nabla_{s_i}\mathcal{L} = -\sum_m x_m \nabla_{s_i} C_m, \tag{5}$$

$$\dot{x}_m = \nabla_{x_m}\mathcal{L} = C_m, \tag{6}$$

which in our formulation [Eqs. (3) and (4)] would correspond to an *unbounded*, voltage-controlled set of memristors with equal weight. In Ref. [43] the equations for the multipliers are altered to $\dot{x}_m = x_m C_m$, which has the effect of making the system hyperbolic, and is analogous to choosing *unbounded*

current-controlled memristors in Eq. (4). The dynamics of both systems are such that the variables s_i of the optimization problem act to minimize the energy, while the weights x_m act to increase it, forming a sort of competitive dynamics which seek out saddle points in the Lagrangian. The weights may be reexpressed as an integral memory term in the s_i equations and so may be interpreted asxbrk "memory terms."

The continuous constraint weighting that these Lagrangian methods perform bears a close resemblance to DMMs. However, the existence of DLRO in these systems has never been explored. In the following we propose a system that combines the bounded motion of memristive variables and attempts to explicitly include a form of long-range order in their dynamics. The model produces a transient phase of avalanches similar to those seen in DMMs [27–29,44] and allows us to continuously control the extent to which both the memory and DLRO are present. Using this we show that the transitions in these avalanches are capable of flipping an extensive number of variables in the system and that both the correlation length and success probability are modulated by the range of these memory variables. Most importantly, the behavior this system shows is a consequence of continuity and does not have discrete analogues.

The inspiration for a simplified model of DMMs draws on the notion of *rigidity* in a condensed matter system. For example, the breaking of translational symmetry also coincides with presence of an elastic potential energy contribution leading to the collective motion of a solid phase [45]. The presence of a continuous symmetry in the equations and its effective breakdown can give rise to behavior analogous to zero modes in statistical physics or field theory [46]. As a consequence, along some directions of the phase space the system can respond in a correlated, or "rigid," manner in which large clusters of variables will transition together [28,29].

For example, in a lattice of continuous "spins" obeying [here $\sigma'(\cdot)$ is the derivative of $\sigma(\cdot)$, giving a rect function which limits the range of v_i to $[-1, 1]$],

$$\dot{v}_i = -\sum_j |J_{ij}|[\sigma(v_i) - \mathrm{sgn}(J_{ij})\sigma(v_j)]\sigma'(v_i), \tag{7}$$

$$\sigma(x) = \begin{cases} 1, & x > 1 \\ x, & -1 \leqslant x \leqslant 1 \\ -1 & -1 < x \end{cases}, \tag{8}$$

the system will exponentially relax to a state in which every variable v_i takes the value $\mathrm{sgn}(J_{ij})v_j$ for all of its neighbors v_j. If the underlying lattice is ferromagnetic ($J_{ij} = 1$), then taking any spin to its limiting value $v_i = \pm 1$ via an external field will cause the entire lattice to transition with it in a manner analogous to long-range order. In contrast, for a discrete system in the ferromagnetic state $s_i = 1$, flipping a single spin will not cause the rest of the lattice to transition as it will not change the sign of any local fields. The ability of a local perturbation to flip large clusters of spins might benefit a solver attempting to satisfy a constraint while maintaining the satisfaction of its neighbors.

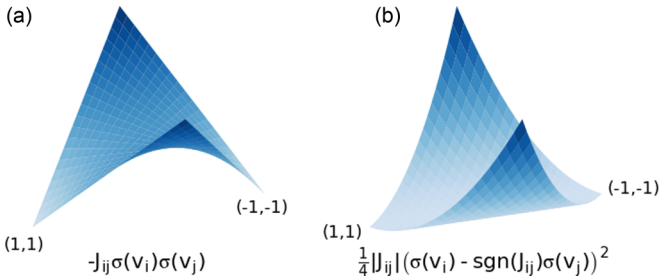

FIG. 4. During the dynamics, memristive variables x_{ij} act to interpolate between gradient dynamics on (a) the original interaction, when a constraint is unsatisfied ($x_{ij} \to 1$), and (b) the "rigid" interaction when the constraint is satisfied $x_{ij} \to 0$. This removes the energy barrier between the two states that satisfy the interaction, allowing transitions of large clusters of variables. The satisfied configurations are denoted for the ferromagnetic case, $v_i = v_j$, and below each surface we give the effective energy function on which the voltages follow gradient dynamics.

This may be viewed as gradient motion on the potential,

$$E = \frac{1}{2}\sum_{\langle ij\rangle} |J_{ij}|[\sigma(v_i) - \mathrm{sgn}(J_{ij})\sigma(v_j)]^2 \tag{9}$$

$$= \frac{1}{2}\sigma(\vec{v})L_J\sigma(\vec{v}), \tag{10}$$

where L_J is a weighted graph Laplacian. For a satisfiable instance, this operator will always have a zero eigenvalue corresponding to the ground state and which connects the two satisfied states. However, the presence of unsatisfiable or frustrated constraints renders this impossible in the above model: in this case all spins will relax to $v_i = 0$, and pulling a single spin to ± 1 will not propagate through the lattice.

We can combine the weighted gradient dynamics of the LPNN-type systems (5) with terms that induce rigidity (7) by utilizing the bounded motion of memristive variables to smoothly transition between these two interactions,

$$\dot{v}_i = \sum_j J_{ij}x_{ij}\sigma(v_j) - (1 - x_{ij})\frac{|J_{ij}|}{2} \times [\sigma(v_i) - \mathrm{sgn}(J_{ij})\sigma(v_j)], \quad v_i \in [-1, 1], \tag{11}$$

$$\dot{x}_{ij} = \beta x_{ij}(1 - x_{ij})\{|J_{ij}|[1 - \mathrm{sgn}(J_{ij})v_iv_j] - \gamma\}. \tag{12}$$

When $x_{ij} \approx 1$ the voltages follow the fields imposed by the neighboring voltages as in an LPNN with $\mathcal{L} = \sum_{\langle ij\rangle} x_{ij}|J_{ij}|[1 - \mathrm{sgn}(J_{ij})v_iv_j]$, causing them to take the integral values $v_i = \pm 1$, which agree with the sign of the overall local field. Constraints which are satisfied by this configuration will then see their contribution to the field reduced as $x_{ij} \to 0$. These constraints then move under the equation of motion $\dot{v}_i = -\frac{|J_{ij}|}{2}[\sigma(v_i) - \mathrm{sgn}(J_{ij})\sigma(v_j)]$ which allows v_i and v_j to transition collectively between the two satisfied states of the constraint. The two interaction terms interpolated between are displayed in Fig. 4. The voltages v_i are limited to the interval $[-1, 1]$. From a state of the dynamical system, the spins of the original Ising model (1) are assigned as $s_i = \mathrm{sgn}(v_i)$ such that the spins of the Ising model undergo the orthant dynamics of the underlying continuous voltages.

The memory state follows the simplest equation for a *bounded*, volatile memristor subject to an effective voltage $|J_{ij}|[1-\mathrm{sgn}(J_{ij})v_i v_j]$. This voltage is the energy with which the constraint is violated, and the constant γ sets a threshold below which x_{ij} will begin to decay. The constant β indicates that the memristive timescale is generally different from the voltage timescale (set by the *RC* constant at the node), which will play an important role in our analysis of the system. While the memory variables do not directly interact, their coupling through the voltages leads to an effective interaction. For the memristive network model shown in Refs. [47,48] it was shown that the network topology leads to pairwise interactions obeying a Lyapunov function such that despite the lack of an explicit interaction term in the dynamical equations, interactions between memristive variables are present through coupling with the voltages.

To clarify the dynamics of this model and the function of the rigidity terms, we consider a two-spin system coupled with $J_{12}=1$, where the first spin is subject to a small local field $1>h>0$. Initializing the system with $x_{12}\approx 1$, the voltages will initially obey

$$\dot{v}_1 = \sigma(v_2)+h, \tag{13}$$

$$\dot{v}_2 = \sigma(v_1). \tag{14}$$

If set near $v_1=v_2=-1$ the voltages will quickly relax to this state, satisfying the J_{12} constraint and causing x_{12} to decrease. For a small field h, the voltages will remain at $-1,-1$ until $x_{12}\approx 0$ at which point they will obey

$$\dot{v}_1 = -[\sigma(v_1)-\sigma(v_2)]+h, \tag{15}$$

$$\dot{v}_2 = -[\sigma(v_2)-\sigma(v_1)]. \tag{16}$$

The local field h will now cause v_1 to drift in the positive direction, and the interaction will cause v_2 to drift along with it, causing both spins to transition collectively to $v_1=v_2=1$.

Over the course of the dynamics, the "rigidity terms" above allow voltages to form clusters with satisfied constraints that are capable of transitioning together under the influence of neighboring unsatisfied constraints. This has a dramatic effect on the dynamics, and inclusion of these "rigidity terms" to the gradient-like first terms in Eq. (11) acts to ensure that these transitions maintain the satisfaction of these clusters. The form of the interactions that are transitioned between is displayed in Fig. 4 such that after relaxing to one of the two states which satisfy the constraint, the decay of $x_{ij}\to 0$ removes the barrier between the two states allowing the simultaneous transition of the two variables between them. Any unsatisfiable spin system may be associated with one or several satisfiable instances formed by removing any unsatisfied bonds in the ground state. The dynamics of the system attempt to discover the underlying satisfiable instance as sublattices where $x_{ij}\to 0$.

We simulate the system described by Eqs. (11) and (12) from random initial voltages and $x_{ij}(0)=0.99$, integrating the equations of motion until the energy (1) (calculated from the signs of the voltages) has reached the planted ground state or some maximum time has elapsed. This is typically chosen quite long, such that the system solves an instance with a probability $p\approx 0.95$ for a given initial condition. For a more detailed discussion of the numerical implementation, see the Supplemental Material [39]. A typical run, showing the voltages, memristances, and energy of the system is shown in Fig. 5 on a two-dimensional instance, $L=15$, where we also show that in the absence of constraint weighting via the memory variables ($\dot{x}_{ij}=0$) the system is unable to reach the ground state [the red curve in Fig. 5(c)]. In this case the system undergoes gradient dynamics and converges to a local minimum of $\mathcal{H}=-\sum_{\langle ij\rangle}J_{ij}v_i v_j,\ v_i\in[-1,1]$. The action of the memory variables may be interpreted as slowly modifying this landscape to destabilize these local minima and push the system into an avalanche. That these avalanches display DRLO, is a feature of the added "rigidity terms" in Eq. (11).

V. DYNAMICAL LONG-RANGE ORDER

The discussion of DLRO in continuous dynamical systems is complicated by the continuity of the dynamics, making it difficult to clearly infer causal relationships between changes in variables. However, we can take advantage of the timescales above to separate the dynamics into causally related events. As shown in Fig. 5(a) when we slow the memristor timescale β relative to that of the voltages (e.g., by choosing $\beta=1/400$), after the initial transient the dynamics progress through a series of rapid transitions interpretable as *avalanches* (or instantons in the field-theory language [28,29]). After an initial relaxation in which the gradient dynamics of the voltages rapidly seek out critical points in the energy landscape, the slow evolution of the memristive dynamics transforms the stability of these points [27–29], leading to a subsequent relaxation. As more constraints become satisfied and transition to a rigid interaction, larger clusters of voltages begin transitioning together [see Fig. 5(c)] in a manner analogous to the dynamics of physical systems in the vicinity of a phase transition [27,49]. We note that in contrast to gradient dynamics the energy is *not* guaranteed to decrease monotonically, but the transitions induced by the memristive variables allow the system to reach lower values than are achievable through gradient dynamics alone.

We now argue that the existence of DLRO in the dynamics of the memcomputing solver is strongly connected to its ability to solve an optimization instance. As a measure of DLRO we compute correlation functions over the largest avalanche that occurs in a simulation. In the limit that the timescales become separated (i.e., the slow driving limit) the points at which each avalanche occurs tend towards well-defined times as seen in Fig. 5(a). For small β these events may be detected as sharp spikes in the voltage derivatives. (See Supplemental Material [39] for a detailed discussion of the method used to extract the structure of the avalanches.) We are interested in the voltages or spins which change sign in the avalanche and thus will affect the energy of the system. We define the avalanche configuration as $\Delta_i=1$ for all spins which change sign during an avalanche, and $\Delta_i=0$ otherwise. A few typical examples of these avalanches and their sizes occurring during dynamics are plotted in Fig. 5(c).

Using the avalanche configurations we are able to compute correlation functions for these events and investigate their

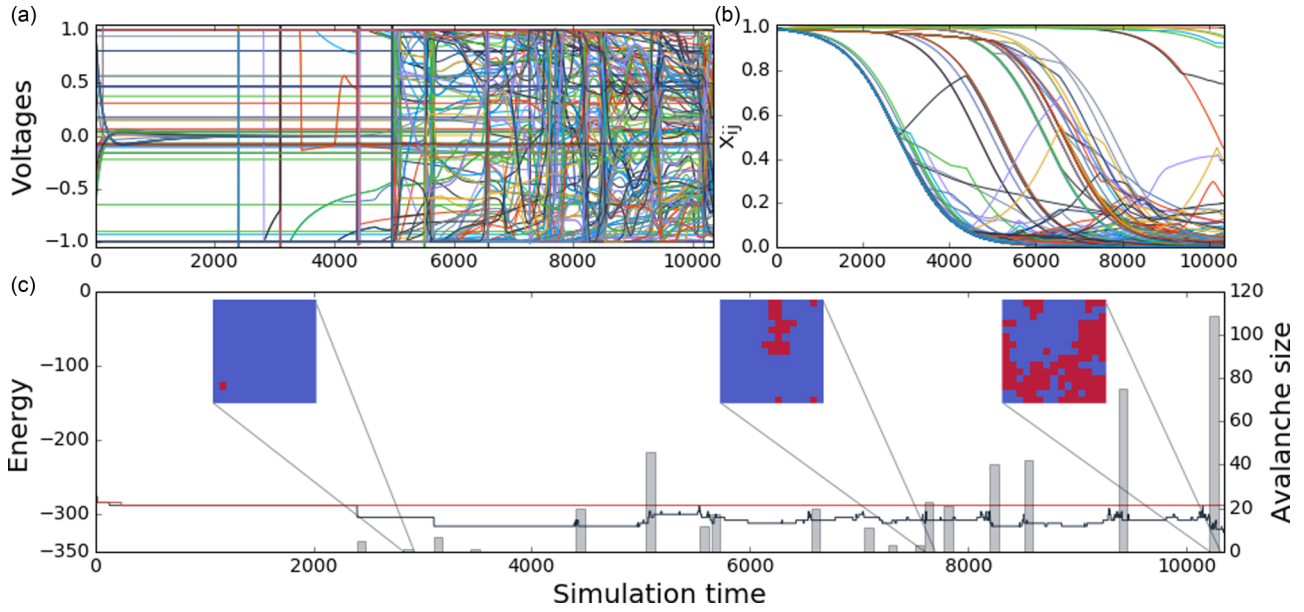

FIG. 5. When the model of Eqs. (11) and (12) is simulated for a two-dimensional instance ($L = 15$) under a separation of timescales ($\beta = \frac{1}{400}$), the voltage trajectories (a) evolve under a series of sharply defined avalanches due to the slow motion of the memristors (b) modifying the clause weights. In panel (c) we have plotted the energy (left axis) without the influence of memristors (red, $\beta = 0$) and with them (black, $\beta = \frac{1}{400}$, $\gamma = 0.65$) showing that the motion of the memory variables allows the system to reach a far lower energy, and ultimately the ground state. The sizes of the avalanches (c, right axis) are plotted as gray bars, showing that their size grows over the course of the simulation until a large avalanche brings the system to its ground state. The avalanches are depicted in the inset in red with the rightmost inset corresponding to the largest avalanche in the run.

decay across the lattice. For each run (defined as generating a unique instance and initial conditions) the system is simulated until it reaches the ground state or a maximum time is reached. If the instance is solved within this interval, the largest avalanche is selected and its configuration and first flipping spin are stored. By averaging across a sample of configurations generated on different instances and initial conditions, suitably shifted so that the initial flipping spins coincide, the probability that a voltage a distance r from the initial spin changes sign, $\langle\Delta\rangle(r)$, may then be calculated. In order to achieve large distances with reasonable simulation times, we calculated these correlations in both two-dimensional ($L = 15, 19, 23$) as well as three-dimensional ($L = 7, 8$) systems. For the parameters tested, success probabilities ranged from 92.8% to 98.6% depending on size and dimension.

As shown in Fig. 6, the largest avalanches possess correlations that take finite values all the way to the furthest corner of the lattice, manifesting a form of DLRO. Dimensionally, this requires that the size of the largest avalanche scales as $\sim L^D$ for a system of dimension D, and is thus extensive. We also note that, as the system size increases, the correlations appear to saturate to a dimension (and instance class) -dependent value.

We can control the presence of DLRO by adjusting the relative magnitudes of terms in the equations of motion. First, we set a limit to the minimum size of x_{ij} such that the barrier between the two satisfied states does not completely vanish. This is accomplished by changing the memristive equation $\dot{x}_{ij}$ to

$$\dot{x}_{ij} = \beta(x_{ij} - x_{\min})(1 - x_{ij})\{|J_{ij}|[1 - \mathrm{sgn}(J_{ij})v_i v_j] - \gamma\}. \tag{17}$$

This is simulated for $L = 16$, $D = 2$, $\beta = 1/518$ and $x_{\min} = 0, 0.05, 0.1, 0.2, 0.3$. At each value of $x_{\min}$ correlations and success probabilities were calculated by averaging over 1000, 1000, 800, 600, and 500 runs, respectively, with the results displayed in Fig. 7. Runs at larger values of $x_{\min}$ are more computationally expensive due to the low success probability. Further details on the simulations are included in the Supplemental Material [39].

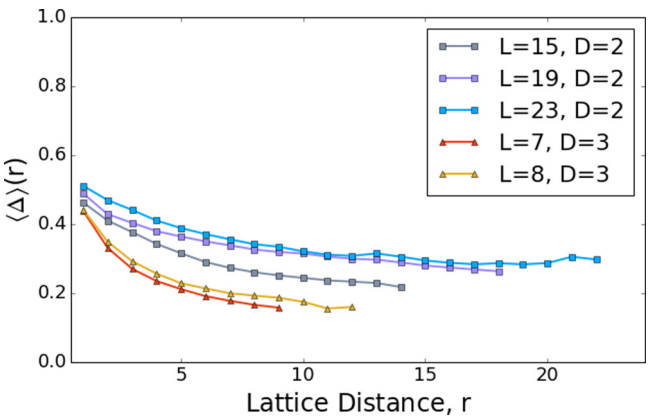

FIG. 6. Spatial correlations, $\langle\Delta\rangle(r)$, among voltages or spins calculated from the orthant dynamics in the slow driving limit of model (11) and (12) for the largest avalanches in two and three dimensions and for different lattice sizes. The correlations take a finite value all the way to the lattice edge, indicating that the largest avalanches are extensive. As the system size increases the values appear to saturate to a dimension-dependent value for this instance class.

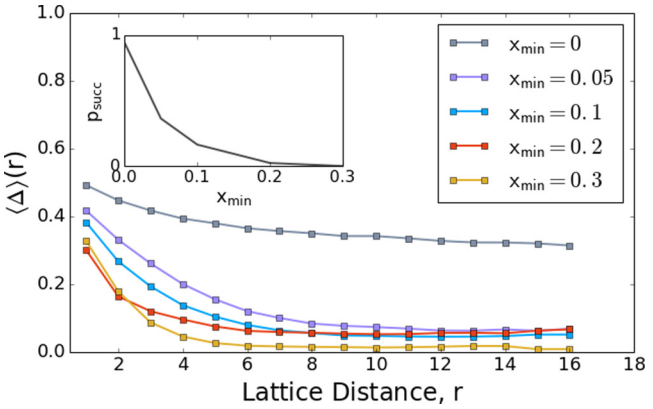

FIG. 7. Correlations and success probability (inset) as the minimum memristive value x_{min} is varied. When $x_{min} = 0$, no barrier remains between the two satisfied states of the interaction, and the avalanche configurations are most long-ranged. As x_{min} increases, both the correlations and the success probability decay rapidly such that by $x_{min} = 0.3$ no instances are solved.

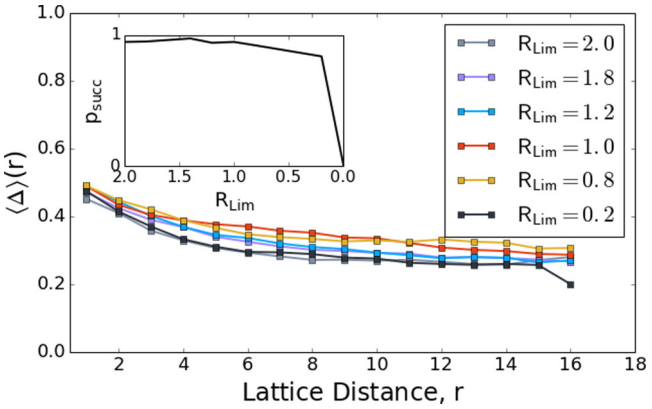

FIG. 8. Correlations and success probability (inset) as the magnitude of the rigidity terms are varied. In contrast to varying x_{min} (as in Fig. 7), which introduces a barrier between the two satisfied states, adjusting the magnitude of R_{Lim} maintains the ability of interacting variables to freely transition between their satisfied states. We observe a wide range in which R_{Lim} has a negligible effect on both the magnitude of avalanche correlations and the success probability.

We observe that both the correlations and the success probability decay rapidly as x_{min} is moved away from zero such that by $x_{min} = 0.3$ the success probability has vanished. In a continuous dynamical system, the lack of energy barriers as $x_{min} \to 0$ is directly connected to the emergence of "zero modes" along which the system can show collective behavior. We also note that as gradient dynamics correspond to $x_{min} = 1$, none of the instances tested would be solved by gradient dynamics alone suggesting the existence of many spurious critical points in this instance class.

We can also vary the magnitude of the added "rigidity terms" independently, modifying our equation of motion of the voltages [Eq. (11)] to

$$\dot{v}_i = \sum_{\langle ij \rangle} J_{ij} x_{ij} v_j - R_{\mathrm{Lim}}(1 - x_{ij}) \frac{|J_{ij}|}{2} \times [v_i - \mathrm{sgn}(J_{ij}) v_j], \quad v_i \in [-1, 1]. \tag{18}$$

These were simulated on the same instance class for R_{Lim} values from 0 to 2, averaging across 300 runs at each value, with the results shown in Fig. 8. In this case the ability of variables to transition collectively is maintained and in contrast to the previous results, we observe a wide range in which varying the size of R_{Lim} has almost no effect on both the correlations and the success probability, and that the success probability vanishes as the size of the rigidity terms is brought to 0. We interpret this as evidence that during the dynamics, voltages whose interaction has been satisfied remain close to the minimum of the interaction term [see Fig. 4(b)] and as such are not sensitive to the overall magnitude of this term. Correlations are not plotted for $R_{Lim} = 0$ as these do not undergo qualitatively similar dynamics and the avalanche detection scheme described in the Supplemental Material [39] fails.

VI. CONCLUSIONS

In this paper, using the frustrated-loop instances based on the Ising spin glass as a well-known benchmark, we have shown that a solver exploiting dynamical long-range order can navigate a nonconvex landscape more efficiently than traditional methods based on annealing despite being composed of only local connections. First, using a full implementation of DMMs we have shown results on 3D frustrated loop instances which indicate this approach is extremely effective in converging to the ground-state solution. In particular, DMMs demonstrated polynomial scaling in reaching the ground state on the tested instances, while all other solvers we have employed scaled exponentially, or possibly polynomially but with a substantially larger degree.

As the effectiveness of DMMs has been attributed to the presence of dynamical long-range order (DLRO) in the dynamics, we have constructed a simple model based on the structure of DMMs in which long-range behavior was introduced heuristically, allowing us to probe the connection between the presence of DLRO and the success probability of the solver. By calculating correlations (or flipping probabilities) on the largest avalanche in the dynamics, we have demonstrated the existence of a form of long-range order which allows a transition to flip spins spanning the entire lattice. By varying the parameters of the model, we have further shown that the magnitude of these correlations and the success probability are controlled by the presence of zero modes along which variables can respond in a correlated manner. The utilization of this effect is enabled by the use of *continuous* dynamical systems to solve natively discrete problems. The results presented here further reinforce the advantages of employing collective dynamics to compute hard problems efficiently.

ACKNOWLEDGMENTS

F.S. and M.D. acknowledge partial support from the Center for Memory and Recording Research at UCSD. The Falcon solver used in the reported simulations has been provided by MemComputing, Inc. [50]. The authors would be delighted to provide, upon request, all instances of the spin-glass problems used in this work.

[1] C. Moore and S. Mertens, *The Nature of Computation* (Oxford University Press, New York, 2011).

[2] C. H. Papadimitriou and K. Stieglitz, *Combinatorial Optimization* (Dover Publications, Mineola, 1998).

[3] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization* (John Wiley & Sons, Hoboken, 2013).

[4] H. A. Kautz, A. Sabharwal, and B. Selman, in *Handbook of Satisfiability*, edited by A. Biere, M. Heule, H. van Maaren, and T. Walsh, Frontiers in Artificial Intelligence and Applications Vol. 185 (IOS Press, Fairfax, 2009), pp. 185–203.

[5] C. P. Gomes, H. Kautz, A. Sabharwal, and B. Selman, in *Handbook of Knowledge Representation*, edited by F. Van Harmelen, V. Lifschitz, and B. Porter, Foundations of Artificial Intelligence Vol. 1 (Elsevier, Oxford, 2008), pp. 89–134.

[6] http://www.maxsat.udl.cat/16/index.html.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[8] K. H. Fischer and J. A. Hertz, *Spin Glasses*, Cambridge Studies in Magnetism Vol. 1 (Cambridge University Press, New York, 1993).

[9] S. Cocco, R. Monasson, A. Montanari, and G. Semerjian, in *Computational Complexity and Statistical Physics*, edited by A. Percus, G. Istrate, and C. Moore (Oxford University Press, New York, 2006), pp. 63–106.

[10] B. Selman, H. J. Levesque, D. G. Mitchell *et al.*, in *Proceedings of the Tenth National Conference on Artificial Intelligence*, edited by P. Rosenbloom and P. Szolovits (AAAI Press, Cambridge, 1992), pp. 440–446.

[11] U. Schöning, in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, 1999), pp. 410–414.

[12] W. Wang, J. Machta, and H. G. Katzgraber, Phys. Rev. E **92**, 013303 (2015).

[13] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, Science **295**, 2427 (2002).

[14] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).

[15] A. J. Bray and D. S. Dean, Phys. Rev. Lett. **98**, 150201 (2007).

[16] I. Hen, J. Job, T. Albash, T. F. Rønnow, M. Troyer, and D. A. Lidar, Phys. Rev. A **92**, 042325 (2015).

[17] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, Phys. Rev. X **6**, 031015 (2016).

[18] H. G. Katzgraber, F. Hamze, Z. Zhu, A. J. Ochoa, and H. Munoz-Bauza, Phys. Rev. X **5**, 031026 (2015).

[19] C. G. Langton, Physica D **42**, 12 (1990).

[20] D. R. Chialvo, Nat. Phys. **6**, 744 (2010).

[21] R. H. Swendsen and J. S. Wang, Phys. Rev. Lett. **57**, 2607 (1986).

[22] J. Houdayer, Eur. Phys. J. B **22**, 479 (2001).

[23] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[24] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, Phys. Rev. Lett. **115**, 077201 (2015).

[25] F. L. Traversa and M. Di Ventra, IEEE Trans. Neural Networks Learn. Syst. **26**, 2702 (2015).

[26] F. L. Traversa and M. Di Ventra, Chaos **27**, 023107 (2017).

[27] M. Di Ventra and F. L. Traversa, J. Appl. Phys. **123**, 180901 (2018).

[28] M. Di Ventra, F. L. Traversa, and I. V. Ovchinnikov, Ann. Phys. (Berlin) **529**, 1700123 (2017).

[29] M. Di Ventra and I. V. Ovchinnikov, Ann. Phys. **409**, 167935 (2019).

[30] F. L. Traversa, P. Cicotti, F. Sheldon, and M. Di Ventra, Complexity **2018**, 7982851 (2018).

[31] H. Manukian, F. L. Traversa, and M. Di Ventra, Neural Netw. **110**, 1 (2019).

[32] F. L. Traversa and M. Di Ventra, arXiv:1808.09999.

[33] S. Arora and B. Barak, *Computational Complexity: A Modern Approach* (Cambridge University Press, Cambridge, 2009).

[34] F. Barahona, J. Phys. A: Math. Gen. **15**, 3241 (1982).

[35] H. Jia, C. Moore, and B. Selman, in *International Conference on Theory and Applications of Satisfiability Testing*, edited by H. H. Hoos and D. G. Mitchell, Lecture Notes in Computer Science Vol. 3542 (Springer, Berlin, 2004), pp. 199–210.

[36] A. D. King, T. Lanting, and R. Harris, arXiv:1502.02098.

[37] T. Albash and D. A. Lidar, Phys. Rev. X **8**, 031016 (2018).

[38] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York, 1990).

[39] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.100.053311 for details concerning instance creation, simulations, and tuning.

[40] IBM ILOG CPlex Optimizer, https://www.ibm.com/analytics/cplex-optimizer.

[41] S. Zhang and A. Constantinides, IEEE Trans. Circuits Syst. II **39**, 441 (1992).

[42] M. Nagamatu and T. Yanaru, Neurocomputing **13**, 119 (1996).

[43] M. Ercsey-Ravasz and Z. Toroczkai, Nat. Phys. **7**, 966 (2011).

[44] S. R. Bearden, F. Sheldon, and M. Di Ventra, Europhys. Lett. **127**, 30005 (2019).

[45] P. W. Anderson, *Basic Notions of Condensed Matter Physics* (Benjamin-Cummings, Menlo Park, 1984).

[46] M. E. Peskin and D. V. Schroeder, *An Introduction to Quantum Field Theory* (CRC Press, New York, 2018).

[47] F. Caravelli, F. L. Traversa, and M. Di Ventra, Phys. Rev. E **95**, 022140 (2017).

[48] F. Caravelli and P. Barucca, Europhys. Lett. **122**, 40008 (2018).

[49] F. C. Sheldon and M. Di Ventra, Phys. Rev. E **95**, 012305 (2017).

[50] http://memcpu.com.