
A

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

by

APPROVAL SHEET

This

is submitted in partial fulfillment of the requirements
for the degree of

Author:

Advisor:

Advisor:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:



Jennifer L. West, School of Engineering and Applied Science

To My Family, Friends, and Relatives

Abstract

Despite the colossal growth in computation power fueled by digital computers, there exist many computational problems that are still considered intractable to solve using such computing platforms. As a case in point, many problems in combinatorial optimization belong to the NP-hard (non-deterministic polynomial-time hard) computational complexity class and computing their solutions on digital computers typically requires exponentially increasing resources (computing time and memory) with increasing problem size. Consequently, solving even problems of moderate size can become unmanageable. This motivates the exploration of alternative computing models that can be more efficient in solving such problems.

Analog dynamical systems as well as computational models inspired by such platforms offer a promising alternative for solving such problems. For example, Ising machines realized using coupled oscillators have been extensively investigated for accelerating hard combinatorial optimization problems. While Ising machines help showcase the potential of the dynamical system-based approach, they are constrained in their capabilities. Specifically, an Ising 'spin' only allows two states and the traditional Ising model can only capture quadratic interactions. However, many practical combinatorial optimization problems entail more than two states as well as higher order (>2) interactions - demands that are typically accomplished using extensive computationally intensive pre-processing, resulting in an expansion in problem size.

Therefore, in this dissertation, new coupled oscillator-inspired computational models are formulated that not only allow variables with more than two values (commonly termed as

the Potts model) but also capture higher order interactions. This work advances the capabilities of the analog approach by largely circumventing the need for pre-processing required with Ising machines. Complementing this effort, the properties of such non-linear dynamical systems are analyzed from a control-theoretic viewpoint, allowing critical insights into the design and optimization of the computational properties of such systems.

Finally, to test the potential performance benefits of the models developed in this work, a 3- and 4-state Potts machine designed to solve the Max-K-Cut problem ($K=3,4$) is implemented on an FPGA. Evaluations performed on graphs with up to 10,000 nodes from the G-Set dataset reveal that the new models combined with the inherent parallelism incorporated in the FPGA architecture can provide a $\sim 390x$ speedup over the state-of-the-art GPU-based simulated bifurcation machine.

In summary, this dissertation contributes to advancing the design and computational capabilities of analog dynamical systems as an approach for accelerating a class of computationally intractable problems.

Acknowledgment

I am extremely grateful to my advisor, Dr. Nikhil Shukla, for his priceless guidance and support throughout my entire PhD. I believe that I have significantly developed professionally and personally under his mentorship. He has guided me in cultivating critical thinking, problem-solving approaches, effective communication of research findings, and assimilating diverse cultures. So, I sincerely express my gratitude to my advisor Dr. Nikhil Shukla.

I would like to express my true gratitude to my doctoral committee members, Dr. Avik Ghosh, Dr. Zongli Lin, Dr. Vijaykrishnan Narayanan, Dr. Kyusang Lee, and Dr. Tariq Iqbal for their invaluable advice on my research work during my proposal and defense.

I extend my gratitude to all of our collaborators across universities for their valuable assistance. I also appreciate the friendly support and assistance from my groupmates in the CHRL lab at UVA throughout my entire PhD journey, with a special acknowledgment to Antik Mallick.

Finally, I would like to thank my family members, friends, and relatives. I am wordless to express my gratitude to my parents, Wasek Sarkar and Amena Khaton who have always been with me with their prayers, love, emotion, inspiration, and support. I would like to thank my siblings for their love and inspiration. I would also like to thank all my friends, specifically Faysal, Sarasij, Asanka, and Shifat for their friendship, help, and support. Last but not least, I would like to thank all my relatives across countries.

Table of Contents

1. Introduction	7
1.1. Background	7
1.2. Alternative computing approaches to solving combinatorial optimization: prior work	8
1.3 History of solving combinatorial optimization with synchronized oscillators	11
1.3.a. Synchronization of coupled oscillators	11
1.3.b. Coupled oscillators under external injection: the oscillator Ising machine (OIM)	13
1.4. Motivation of the dissertation	17
1.5. Contributions	22
1.6. Dissertation organization	23
2. Coupled Oscillator-based Potts Machine	25
2.1. Theoretical framework of oscillator Potts machine	25
2.2. FPGA implementation of oscillator Potts machine	30
2.2.a. Computational model implemented on FPGA	30
2.2.b. Architecture of the Potts machine-based FPGA accelerator	31
2.2.c. Results	33
2.2.c.1. Max-K-Cut solution	33
2.2.c.2. Resource utilization and energy benchmark	37
3. Coupled Oscillators for Solving COPs with Higher Order Interaction	39
3.1. Dynamical systems to solve COPs with higher order interaction	39
3.1.a. Boolean SAT	41
3.1. b. Boolean NAE-SAT	48
3.1.c. Set splitting	51
3.1.d. Integer factorization	51
3.1.e. Graph isomorphism	56
3.1.f. Minimization of the Ising Hamiltonian and MaxCut	59
3.2. Oscillator-based dynamical systems to solve COPs on hypergraph (SAT)	62

3.3. Oscillator Ising machine with higher order interaction among Ising spins	74
3.3.a. NAE-K-SAT problem.....	76
3.3.b. Max-K-Cut on a hypergraph.....	84
3.4. Alternative formulations for higher order Ising spin interactions	90
3.4.a. Higher order interactions: even order	90
3.4.b. Higher order interactions: odd order	92
4. Dynamical Properties of Coupled Oscillator-based Computing Systems	95
4.1. Control theoretic analysis of dynamical systems	97
4.1.a. Stability analysis of fixed points of an OIM using the linearization method.....	98
4.1.b. Assessing the stability of fixed points of an OIM from energy landscape.....	105
5. Future Work.....	109
5.1. Statistical sampling with OIM.....	109
5.2. Parallelization of statistical sampling with OIM	113
6. Conclusion	115
Appendices	117
7. Publications & Patents	140
Bibliography	143

Chapter 1

1. Introduction

1.1. Background

Modern information processing has largely relied on digital computers based on the Von-Neumann architecture. Problems are solved by creating their Boolean abstractions which are then systematically processed using a suite of logic circuits (arithmetic logic unit) and memory that are realized using Silicon CMOS-based digital switches. There is a well-established mapping between the Boolean computing paradigm and the underlying hardware that supports digital computation. Moreover, the driving force behind the proliferation of digital computing has been the economics of Moore's law [1]. Traditionally, this has made computing exponentially more energy efficient (Koomey's law [2]) as well as cost efficient (quantified using Floating Point Operations Per Second (FLOPS)/\$), facilitating an exponential reduction in \$/bit. However, with the slowing of Moore's law and the cost of manufacturing evermore complex transistor architectures no longer reducing, the energy and cost efficiency have plateaued. The impact of this is felt most acutely when solving computationally intractable problems such as those in combinatorial optimization (e.g., MaxCut, Boolean Satisfiability, graph coloring) that require exponentially increasing computational resources (computation time, memory) with increasing problem size [3]. The inability for continued improvement of the hardware efficiency in conjunction with the exponentially large number of computations required for solving such problems means that even problems of moderate size become unmanageable.

Furthermore, such problems find utility across a broad set of industrial and scientific applications. For example, many tasks in operations research [4], artificial intelligence [5], communication networks [6], security [7], software engineering [8], data mining [9], computational and molecular biology [10], biocomplexity [11], molecular chemistry [12],[13], and scientific discovery [14] etc., can be expressed as combinatorial optimization problems (COPs). Consequently, this motivates the search for alternative computing approaches that could offer improved computational efficiency in solving such problems.

1.2. Alternative computing approaches to solving combinatorial optimization: prior work

Various alternative computing methods spanning the quantum mechanical and classical domains are being actively investigated for accelerating the computation of hard combinatorial optimization problems. Quantum computing aims to exploit collective quantum physical properties i.e., superposition and entanglement properties of qubits to perform computation. Though a quantum computer (QC) possesses the same computability as a classical digital computer, it has the potential to provide

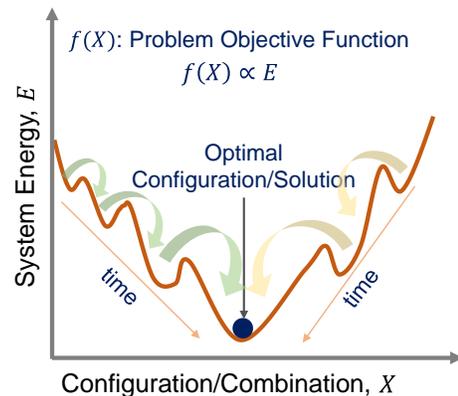


Figure 1.1. Schematic depiction of the evolution of a dynamical system; objective function of an optimization problem is mapped onto the energy of the system.

exponential speedup in solving hard problems (this speedup is known as quantum supremacy [15]). Recently, several forms of QCs have been physically implemented. For

instance, IBM recently unveiled their quantum gates-based (cloud accessible) QC [16] while D-wave has implemented quantum annealers [17]. Such quantum annealers are designed to solve optimization problems [18],[19], specifically quadratic optimization problems in QUBO (quadratic unconstrained binary optimization) format [20],[21]. Despite the promise of quantum computers, their physical implementation remains a major challenge; cryogenic cooling to ~mK temperatures [22] required for qubits entails a significant energy cost as well as creates challenges for their scalability. Additionally, quantum computers are prone to error [23] resulting in implementation challenges. Such challenges inspire us to keep the door open for other approaches to emerging computing.

Classical dynamical systems have recently been shown as a promising approach to solving intractable COPs [24],[25]. The underlying dynamics governed by energy minimization in dynamical systems can find an elegant equivalence in the minimization of multi-dimensional objective functions that define COPs as depicted in Fig. 1.1 [26]-[28]. The high-level motivation behind exploring this paradigm is that unlike digital solvers that typically rely on batch, discrete time, iterative updates (lacking temporal locality) and shared states (lacking spatial locality), continuous-time dynamical systems exhibit rich spatio-temporal properties that inherently offer a highly parallelized approach for searching the high dimensional combinatorial space. Physical manifestations of this approach include probabilistic computing [29],[30], optical computing [31],[32], in-memory computing [33]-[36] and coupled oscillator-based computing [37] among others. Additionally, Physics-inspired algorithms such as simulated annealing [38] and simulated bifurcation [39] have been widely studied to solve COPs. Their implementations on

hardware accelerators like GPUs [40],[41], FPGAs [42],[43] as well as ASIC implementations [44]-[46] have shown promising results in solving COPs.

System	Building Block	Underlying Physics	Operating Temperature
D-Wave Annealer [22],[47]	Qubit	Quantum tunneling	15 mK
Coherent Ising Machines [48]	Optical parametric oscillators	Optical coherence	300 K
Analog SAT Solver [49]	Analog inverter and integrator	Gradient descent	300 K
Oscillator Ising Machine [50],[51]	Electronic oscillators	Gradient descent	300 K
Binary Stochastic Neuron (P-Bit) [52]	Low barrier magnets	Boltzmann sampling	300 K
CMOS Annealer [53]	SRAM	Simulated annealing	300 K

Figure 1.2. Table summarizing different approaches for alternative computing platforms to solve hard combinatorial optimization problems.

Since electronic oscillators possess rich spatiotemporal phase dynamics, they have recently received significant attention. There are active research efforts that are dedicated to investigating the promise associated with electronic oscillators for designing computing systems to solve hard COPs. Some key motivations behind oscillator-based computing are that they are compatible with existing technology, compact, operable at room temperature, capable of mass production, and they demand low power. The table in Fig. 1.2 summarizes the key attributes of different emerging computing approaches.

1.3 History of solving combinatorial optimization with synchronized oscillators

1.3.a. Synchronization of coupled oscillators

Oscillators are ubiquitous in nature. Synchronization of oscillators is observed in many natural biological and ecological systems. For instance, the chirping of crickets, the flashing of fireflies, and cardiac cells in mammals [54] exhibit synchronization phenomena. Besides natural systems, synchronization of oscillators was also envisioned for designing computing systems in the mid twentieth century. In fact, von Neumann proposed an oscillator-based logic platform based on sub-harmonic injection locking. Eiichi Goto in Japan developed an oscillator-based parametron [55]. Despite these early innovations, oscillator-based digital computing did not receive much focus due to their implementation challenges and the emergence of CMOS-based digital systems.

However, in the recent years, there has been increasing interest in designing oscillator-based computing, specifically for solving hard combinatorial optimization problems (COPs). Many COPs are conveniently represented by graphs (Fig. 1.3). Such graphs can be mapped on topologically equivalent coupled oscillator networks where nodes (vertices) and edges correspond to oscillators and coupling elements, respectively (shown in Fig. 1.3). The Kuramoto model [56] is a landmark development in that field that helped analyze the synchronization behavior in oscillators. According to this model, the

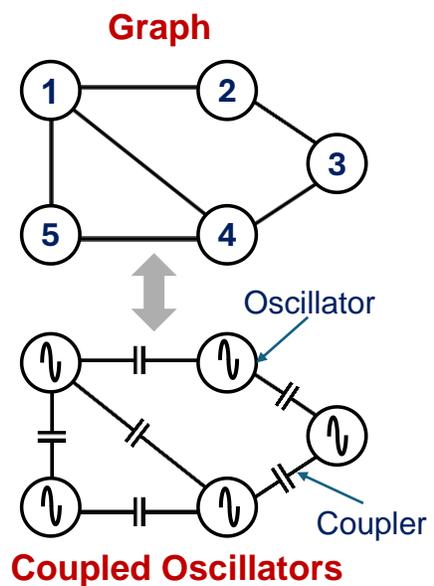


Figure 1.3. Mapping of a graph network onto a coupled oscillator.

phase dynamics of the i^{th} oscillator in a system of N -coupled sinusoidal oscillators can be written as,

$$\frac{d\phi_i}{dt} = \omega_i + k \sum_{j=1, i \neq j}^N w_{ij} \sin(\phi_j - \phi_i) \quad (1.1)$$

where, ϕ_i and ω_i are the phase and frequency of the i^{th} oscillator, respectively; k is the constant coupling strength; w_{ij} is the weight between the i^{th} and j^{th} nodes. This model can provide a theoretical limit of the coupling strength known as the critical coupling strength (k_c) below which oscillators in a system lose synchronization. Further details about the Kuramoto model can be found in the reviews by Rodrigues et al. [57], and Acebrón et al. [58].

Wu et al. [59] proposed a solution to the graph coloring problem using the Kuramoto model. The graph coloring problem is defined as the problem of assigning colors to the nodes of a graph in such a way that two connected nodes cannot have the same color. While using coupled Kuramoto oscillators, graph coloring is achieved by grouping the oscillators based on their phase sequences. Subsequently, Lee et al. [60] proposed a heuristic algorithm based on the Kuramoto model, and Wu et al. [61] proposed a method with adaptive coupling in the Kuramoto model to solve the graph coloring problem. Furthermore, several works demonstrated the solution to the graph coloring and the maximum independent set (MIS) problem with coupled oscillator hardware. MIS problem deals with finding a color group with the possible maximum number of nodes. Parihar et al. [62] experimentally demonstrated the solution to the graph coloring problem and we showcased the solution to the MIS problem [63] (with simulation) with capacitively coupled Vanadium dioxide (VO_2) relaxation oscillators. Mallick et al. demonstrated the

solution to the graph coloring problem [64], MIS problem [65] with a chip of fully reconfigurable 30 capacitively coupled Schmitt trigger oscillators. Among various approaches to oscillator-based computing, researchers have largely focused on designing oscillator based Ising machines. In the next section, we will describe the oscillator Ising machine (OIM).

1.3.b. Coupled oscillators under external injection: the oscillator Ising machine (OIM)

Theoretical framework. An Ising machine can minimize the Ising Hamiltonian defined by the following equation:

$$H = - \sum_{i,j=1,i < j}^N J_{ij} s_i s_j \quad (1.2)$$

Where, s_i represents Ising spin ($s_i \in \{-1, +1\}$), J_{ij} represents the interaction coefficient between i^{th} and j^{th} spins. The Zeeman term is neglected here. The MaxCut problem of a graph, which is a well-known COP, can be directly represented by the Ising Hamiltonian shown in Eq. (1.2). The MaxCut problem is defined as the problem of dividing the nodes of a graph into two sets so that the sum of the weights of the common edges maximizes. The objective function of the MaxCut problem can be represented by an Ising Hamiltonian where spin -1 represents one node set and $+1$ represents the other set (Fig. 1.4), $J_{ij} = -w_{ij}$ (w_{ij} : weight of the edge between the i^{th} and j^{th} nodes). Thus, the minimization of the Ising Hamiltonian maximizes the cut in the corresponding graph. Wang et al. [66] first demonstrated an oscillator Ising machine (OIM). The theory of OIM is derived from the generalized Adler's equation (shown below) [67],[68] which describes the phase dynamics of an oscillator under a perturbation.

$$\frac{d\phi_{osc}}{dt} = \omega_{osc} - \omega_{pb} + \omega_{osc} \mu(\phi_{osc}(t) - \phi_{pb}(t)) \quad (1.3)$$

where, ϕ_{osc} , ϕ_{pb} , ω_{osc} , ω_{pb} are oscillator phase, perturbation phase, oscillator frequency, and perturbation frequency, respectively. μ is the perturbation projection vector (PPV) that quantifies the phase response of the oscillator in presence of a perturbation. Now, for the i^{th} oscillator in a system of N-coupled oscillators, Eq. (1.3) can be written as,

$$\frac{d\phi_i}{dt} = \omega_i - \omega_{nat} + \omega_i \sum_{j=1, j \neq i}^N \mu_{ij}(\phi_i(t) - \phi_j(t)) \quad (1.4)$$

where, ω_{nat} is the natural frequency of the oscillators. For a sinusoidal oscillator, PPV is also sinusoidal. Hence, considering $\omega_i = \omega_{nat}$, for sinusoidal oscillators, Eq. (1.4) can be modified to,

$$\frac{d\phi_i}{dt} = -k \sum_{j=1, j \neq i}^N J_{ij} \sin(\phi_i(t) - \phi_j(t)) \quad (1.5)$$

where k is the constant coupling strength. Eq. (1.5) is nothing, but the Kuramoto model shown in Eq. (1.1). In the presence of a second harmonic injection signal, Eq. (1.5) can be written as,

$$\frac{d\phi_i}{dt} = -k \sum_{j=1, j \neq i}^N J_{ij} \sin(\phi_i(t) - \phi_j(t)) - k_s \sin(2\phi_i(t)) \quad (1.6)$$

where k_s is the injection strength. Eq. (1.6) shows the dynamics of an OIM. Now, an energy function E is defined in a way so that, $-\alpha(\nabla E)_i = \frac{d\phi_i}{dt}$, $\alpha > 0$ (here, $\alpha = \frac{1}{2}$). It ensures that $\frac{dE}{dt} \leq 0$ which means the energy function is nonincreasing and the system

exhibits gradient descent in the energy domain. The following function is such an energy function,

$$E = -k \sum_{i,j=1,i \neq j}^N J_{ij} \cos(\phi_i(t) - \phi_j(t)) - k_s \sum_{i=1}^N \cos(2\phi_i(t)) \quad (1.7)$$

Here, the second harmonic injection with sufficient strength discretizes the oscillator phases to $\{0, \pi\}$. Hence, $\cos(\phi_i(t) - \phi_j(t))$ maps $s_i s_j$ and $-\sum_{i,j=1,i \neq j}^N J_{ij} \cos(\phi_i(t) - \phi_j(t))$ maps the Ising Hamiltonian H shown in Eq. (1.2). Thus, the

minimization of E directly maps the minimization of H (since k and k_s are nonnegative). For Eq. (1.7) it can be derived that $\frac{dE}{dt} = -2 \sum_{i=1}^N \left(\frac{d\phi_i}{dt}\right)^2$ i.e., $\frac{dE}{dt} \leq 0$. Hence

the system evolves towards lower energy unless it gets stuck in a local minimum. Wang et al. [67] referred to this as a Lyapunov function. Thus, OIM performs a *gradient descent* on an energy function that corresponds to the Ising Hamiltonian.

Figure 1.4 shows the computation of an Ising problem (i.e., minimization of Ising Hamiltonian) using an oscillator based Ising machine [69]. The network to be analyzed is

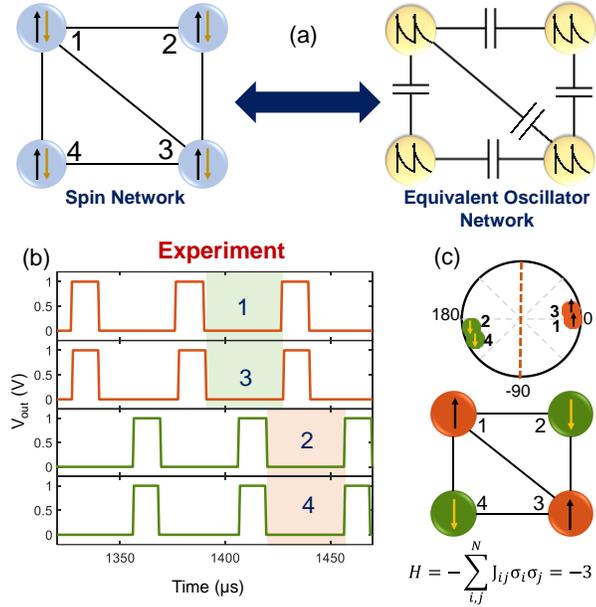


Figure 1.4. (a) A representative network of spins and their interactions along with the corresponding topologically equivalent coupled oscillator network. (b) Experimentally measured time-domain output of the oscillators in the network under second harmonic injection. (c) Corresponding phase plot of the oscillators showing a phase bipartition with each set corresponding to a spin state (± 1). The oscillators (spin) aim to achieve a configuration that minimizes the Ising Hamiltonian H .

mapped to the oscillator hardware by creating a topologically equivalent oscillator circuit using the following relationship: spin \equiv oscillator, and spin interaction \equiv coupling capacitor (only binary interaction considered here), as shown in the example in Fig. 1.4(a). The corresponding waveform (Fig. 1.4b) of the coupled oscillators under the influence of the externally applied second harmonic signal, and the resulting phase plot (Fig. 1.4c) clearly shows a phase bipartition corresponding to the two spin states. The resulting spin assignment that gives rise to the minimum value of H ($= -3$, here) is $(1, 3) \uparrow$; $(2, 4) \downarrow$.

OIM hardware. Many OIM hardware have been proposed with varied electronic oscillators which include both CMOS oscillators and emerging device-based oscillators. While emerging technology-based oscillators possess the promise of compact design and power efficiency, large-scale designs are still dependent on CMOS technology. Wang et al. [70] demonstrated a discrete OIM, implemented with 240 LC oscillators on PCBs where the coupling network has a chimera architecture. A weighted OIM was demonstrated by Chou et al. [71] with 4 discrete LC oscillators. Our previous work [51] presented the first all-to-all connected OIM chip with 30 capacitively coupled Schmitt trigger-based oscillators. It was then scaled to 600 oscillators, coupled in a tiled architecture [50]. A hybrid approach was proposed for obtaining high-quality MaxCut solution using the 600-oscillator OIM, since solution quality degrades with the increase in problem / hardware size. Research efforts were also employed to explore the design of OIM using emerging technology-based devices. For instance, Dutta et al. experimentally demonstrated an OIM with 8 discrete coupled VO₂ nano oscillators [72]. Additionally, a ring oscillator based probabilistic OIM was proposed [73],[74]. Apart from these electronic oscillators, several other types of oscillators have been investigated to design Ising

machines. Spin Hall nano-oscillator [75], spin-torque nano-oscillators (STNOs) [76] (using simulation), and Kerr nonlinear parametric oscillator [77] are some of the candidates that can be put in this list.

Though many research efforts have investigated OIM, specifically the design of OIM hardware, theoretical analysis to understand the dynamical properties of such systems remain sparsely explored. Such analysis can help optimize the design of OIM for better performance. Therefore, this dissertation introduces a control theoretic analysis of OIM.

1.4. Motivation of the dissertation

Challenges with Ising machines. An Ising machine can directly map COPs whose inherent mathematical formulations consist of only pairwise interaction of 2-state variables since the Ising Hamiltonian $H = \sum_{i \neq j}^N J_{ij} s_i s_j$ comprises of pairwise interactions $(s_i s_j)$ of 2-state Ising spins $(s_i \in \{-1, +1\})$. However, many COPs like the Max-K-Cut ($K > 2$) problem need multi-state variable representation (shown in Fig. 1.5a). Furthermore, many COPs such as the hypergraph MaxCut are inherently associated with hypergraphs where a hyperedge can connect more than two nodes (shown in Fig. 1.5b). Their natural objective functions need higher order interactions among variables / spins, e.g., $s_i s_j s_k$. Thus, we can classify COPs in the context of dynamical systems. Fig. 1.5c summarizes this classification in a table. It can be observed that Ising machines only can directly map the binary quadratic (pairwise interaction-based) COPs.

An Ising machine still can solve COPs whose inherent representations need multi-state variables, higher order interaction, or both. For instance, the Max-K-Cut problem

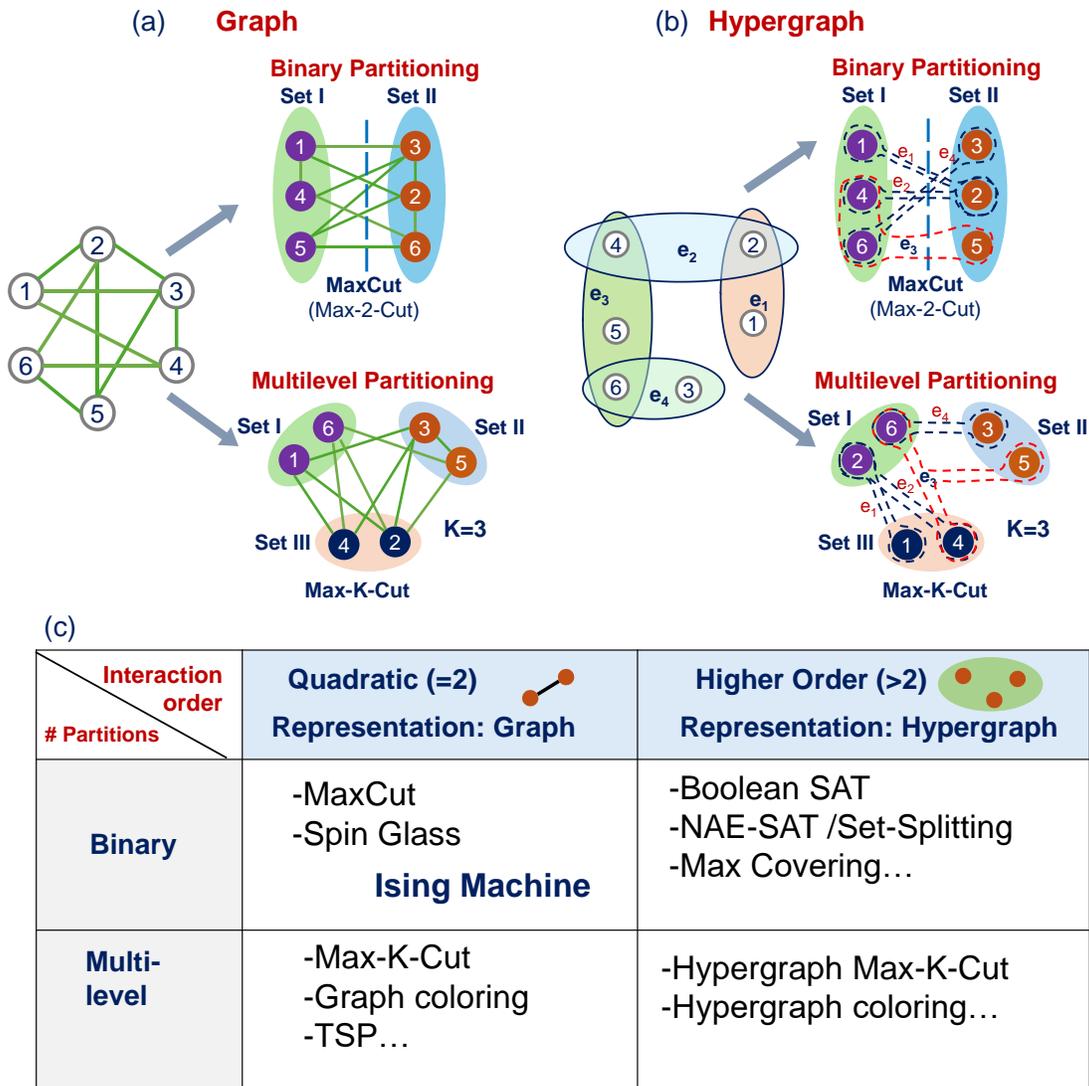


Figure 1.5. Classification of COPs in the context of dynamical system-based computing. Binary and multi-level partitioning in (a) graphs, (b) hypergraphs. (c) Table summarizing the COP classification. Ising machines can inherently map the binary quadratic COPs.

can be solved using Ising machines by converting the input problem using the following method:

Max-K-Cut to Ising problem. Let's consider an N -node graph $G(V, E)$ where V is the set of vertices and E is the set of edges. Now, if we want to solve a problem that aims to divide the node set into K sets so that the number of edges connecting more than one set maximizes, we can consider the following Ising Hamiltonian (details are shown in [78]),

$$H_{K-cut} = C_1 \sum_{i=1}^N \left(1 - \sum_{r=1}^K \frac{1 + s_{i,r}}{2} \right)^2 + C_2 \sum_{(i,j) \in E} \sum_{r=1}^K \frac{(1 + s_{i,r})(1 + s_{j,r})}{4} \quad (1.8)$$

Here, C_1 and C_2 are constants, $s_{i,r} = +1$ denotes that the i^{th} node is placed at r^{th} set and $s_{i,r} = -1$ denotes it is not placed in the r^{th} set. The first term in H_{K-cut} ensures that a node is placed only in one set. The second term ensures the maximization of cuts, i.e., the

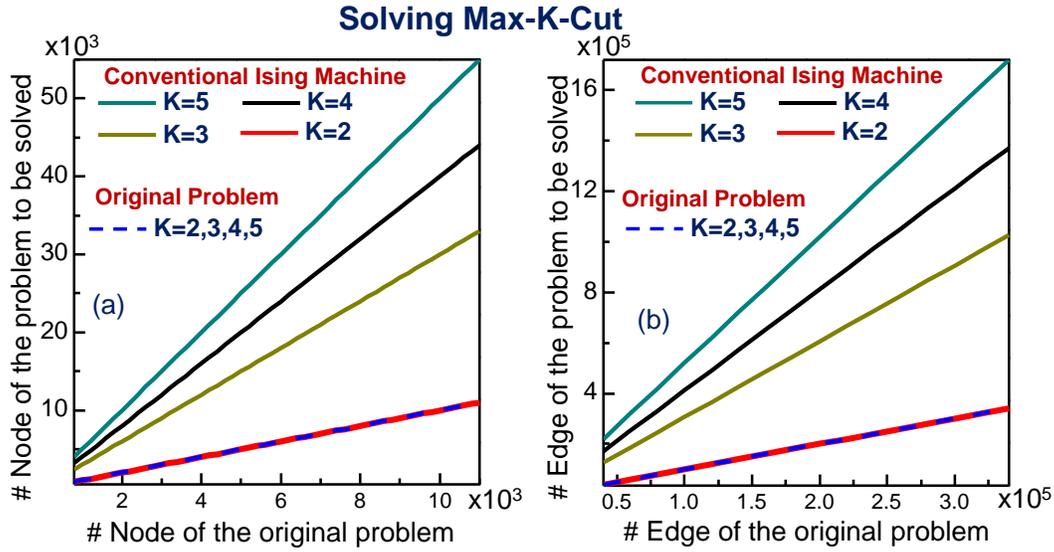


Figure 1.6. The variation in (a) the number of nodes and (b) the number of edges of the graph required to solve the Max-K-Cut problem, as a function of the corresponding quantities in the original input graph.

number of edges connecting more than one set. Eq. (1.8) can be solved using an Ising machine, i.e., it is now an Ising problem. However, the problem size has now increased to KN . Figure 1.6 shows the size of a converted Ising problem as a function of the original problem size (nodes and edges) while solving Max-K-Cut using the Ising machine. It can be observed that the problem size increases significantly (K times) if we want to solve the Max-K-Cut problem with an Ising machine.

Higher order COPs to Ising problems. We can illustrate the conversion of a higher order COP to the Ising problem with the Boolean satisfiability (SAT) problem. Let's consider the following CNF (conjunctive normal form) formula,

$$y = (x_1 \vee x_2) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \quad (1.9)$$

The formula in Eq. (1.9) comprises two conjunctive (logical and) Boolean terms and each of them contains a disjunction (logical OR) of some variables enclosed by parenthesis. Such terms are known as clauses. To solve the decision version of the SAT problem, we need to determine if there exists an assignment of the Boolean variables in Eq. (1.9) that will satisfy the formula y . The optimization version of the SAT, known as the MaxSAT (maximum satisfiability) problem can be solved by finding an assignment of the corresponding variables so that it satisfies the possible maximum number of clauses (in Eq. (1.9), both clauses). For instance, an assignment of $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0$, will satisfy both clauses in the above formula (y). Using binary variables ($x_i \in \{0,1\}$), this problem can be formulated by the following objective function (details are shown in [79]):

$$f = 1 - x_1 - x_2 + x_1x_2 + x_2x_3x_4 \quad (1.10)$$

Further, Eq. (1.10) can be translated to an Ising Hamiltonian shown in Eq. (1.11) by using

$$x_i = \frac{s_i + 1}{2}.$$

$$H_{SAT} = \frac{3 - 2s_1 - s_2 + s_3 + s_4 + 2s_1s_2 + s_2s_3 + s_2s_4 + s_3s_4 + s_2s_3s_4}{8} \quad (1.11)$$

Eq. (1.11) is to be minimized to maximize the number of satisfied clauses in y . Unlike the Ising Hamiltonian formulation ($\sum_{i \neq j}^N J_{ij} s_i s_j$), Eq. (1.11) is non-quadratic since it comprises a third order interaction term ($s_2s_3s_4$). The degree of such an equation will increase with the increase of clause size, i.e., the number of literals (variable in normal or inverted form) in a clause [79].

Now, we will see how to convert the SAT Hamiltonian shown in Eq. (1.11). The process of converting a higher order interaction term to a second order interaction term is known as quadratization. To quadratize the term $s_2s_3s_4$, we can introduce an ancillary (or auxiliary) variable s_{a1} so that $s_{a1} = s_3s_4$. Hence, the Hamiltonian becomes,

$$H_{SAT} = \frac{3 - 2s_1 - s_2 + s_3 + s_4 + 2s_1s_2 + s_2s_3 + s_2s_4 + s_{a1} + s_2s_{a1}}{8} \quad (1.12)$$

Eq. (1.12) is a quadratic function i.e., it is an Ising problem now. However, to ensure that $s_{a1} = s_3s_4$, we need to add a constraint as follows,

$$H_{SAT} = \frac{1}{8} [3 - 2s_1 - s_2 + s_3 + s_4 + 2s_1s_2 + s_2s_3 + s_2s_4 + s_{a1} + s_2s_{a1} + P(4 + s_3 + s_4 - s_{a1} - 2s_{b1} + s_3s_4 - s_3s_{a1} - s_4s_{a1} - 2s_3s_{b1} - 2s_4s_{b1} + 2s_{a1}s_{b1})] \quad (1.13)$$

Here, P is a large penalty. The term $(4 + s_3 + s_4 - s_{a1} - 2s_{b1} + s_3s_4 - s_3s_{a1} - s_4s_{a1} - 2s_3s_{b1} - 2s_4s_{b1} + 2s_{a1}s_{b1})$ ensures (along with a large penalty) $s_{a1} = s_3s_4$ [80]. To reduce the degree of a higher order term by 1, we need to add 2 ancillary / auxiliary Ising spin-based variables. Hence, to quadratize m higher order terms, we need up to $\sum_{\substack{i=1, \\ d_i > 2}}^m 2(d_i - 2)$ auxiliary variables where, d_i is the degree of the i^{th} term. Thus, quadratization of the objective function of a higher order COP necessitates the introduction of additional variables (auxiliary variables) and increases the problem size significantly.

The main motivation behind the tremendous efforts being employed to design an efficient and scaled Ising machine is that any NP problem can be solved using an Ising machine by converting them into Ising problems and then solving that problem. However, such conversion to Ising problems adds significant computation overhead, expands the

problem size multiple times, and reduces the possibility of obtaining valid solutions due to the imposition of additional constraints [78]. Consequently, solving such COPs using Ising machines needs more computing resources (hardware size and time), and solution quality declines since expanded problems must be solved by the Ising machine. These issues motivate the search for designing domain-specific dynamical systems that will be able to directly map and solve a wide range of COPs irrespective of their inherent nature.

Therefore, the goal of this dissertation is to formulate oscillator-based dynamical systems for solving hard combinatorial optimization problems directly (i.e., without the need for conversion to the Ising problem) and to incorporate theoretical methods for analyzing the dynamical properties of coupled oscillator-based computing systems that regulates their computational performance.

1.5. Contributions

The key contributions of this dissertation can be summarized with the following points:

- 1. A novel coupled oscillator-based Potts machine has been formulated with a phase sensitive coupling scheme. The formulated Potts machine-based computational models have been utilized to directly map and solve several COPs that need multi-state variable representations. One of the oscillator Potts machine-based computational models has been implemented on an FPGA accelerator to solve the Max-K-Cut problem. The FPGA implementation has showcased up to 390x speedup over a state-of-the-art Ising machine, in solving the Max-K-Cut solution for up to 10,000 node benchmarking graphs, while maintaining similar solution quality.*

2. *Dynamical systems have been formulated to map and solve multiple higher order COPs including the Boolean satisfiability (SAT) problem.*
3. *A first of its kind oscillator-based higher order Ising machine has been formulated to directly map and solve higher order COPs whose objective functions are represented with Ising spins.*
4. *Control theoretic methods have been employed to analyze the dynamics of oscillator-based computing systems. Such analysis can be utilized to optimize the system parameters of these computing systems for improving computational performance.*

1.6. Dissertation organization

The dissertation is organized as follows:

Chapter 1 has discussed the background and prior related works of this dissertation. The synchronization of coupled oscillators and their direct use in solving combinatorial optimization are briefly discussed in Chapter 1. Here, the theoretical framework for the oscillator Ising machine along with the previous work on its hardware implementations is discussed. Subsequently, the limitations of Ising machines in solving COPs are discussed to frame the motivation behind this research.

The first section of Chapter 2 discusses the theoretical formulation of oscillator-based Potts machines with phase sensitive coupling function. The coupling function and its modification to map several COPs is described briefly. In the second section of Chapter 2, an FPGA implementation of the oscillator Potts machine-based Max-K-Cut solving

algorithm is described. Subsequently, benchmarking of the results obtained from this FPGA implementation is also presented in Chapter 2.

Chapter 3 describes the formulation of several dynamical systems to solve several higher order COPs including the Boolean SAT and its derivative- the NAE-SAT problem. Subsequently, Chapter 3 delves into the formulation of an extended oscillator Ising machine which can directly map higher order interaction among Ising spins and thus solve higher order COPs. Subsequently, a method to incorporate higher order interaction in oscillator Potts machines is discussed in Chapter 3. Additionally, an alternative formulation to express higher order Ising spin interactions with second order interactions are shown in Chapter 3.

Chapter 4 discusses the stability analysis of the fixed points of oscillator-based dynamical systems (here OIM) and its possible utilization in optimizing system parameters for enhanced computational performance. Additionally, a method to analyze the stability of fixed points of oscillator Ising machines directly from the energy landscape is described at the end of Chapter 4.

Chapter 5 describes the potential application of oscillator-based dynamical systems to perform statistical sampling such as Gibbs sampling. Thus, this Chapter discusses the prospective future directions of coupled oscillator-based computing.

Finally, the dissertation is summarized in the conclusion section to highlight the key findings and future directions.

Chapter 2

2. Coupled Oscillator-based Potts Machine

This chapter focuses on the formulation and evaluation (with FPGA) of novel oscillator-based computational models that can directly map COPs with multi-state variables.

The Potts model [81] also known as the clock model is a general form of the Ising model as it deals with K-state variables / spins ($K \geq 2$), while Ising spins have 2-state. K-state Potts spins can be represented by K uniformly spaced angles on a circle $(0, 1 \cdot \frac{2\pi}{K}, 2 \cdot \frac{2\pi}{K}, \dots, (K - 1) \cdot \frac{2\pi}{K})$. Many COPs such as the Max-K-Cut problem, graph coloring problem, and Hamiltonian cycle problem (HCP) inherently need K-state variable representation and hence can be represented with the Potts model. A dynamical system that is designed to minimize Hamiltonians associated with the Potts model is known as the Potts machine. In this effort, oscillator Potts machines are formulated to directly solve multiple COPs. The oscillator Potts machine utilizes a K^{th} harmonic injection as well as a phase sensitive coupling function. This phase sensitive coupling function can be modified to map a wide range of COPs. The model will be briefly described below:

2.1. Theoretical framework of oscillator Potts machine

We will first discuss the formulation of a coupled oscillator Potts machine that can minimize a K-state Potts Hamiltonian associated with the Max-K-Cut problem. The Max-K-Cut problem is defined as the problem of dividing the vertices of a graph into K sets so that the weight sum of the edges connecting more than one set maximizes. Potts spin

can have K values associated with oscillator phase $\phi_i = k \frac{2\pi}{K}$, $k \in \{0,1,2, \dots, (K-1)\}$.

Hence, a Potts spin can be represented by the complex quantity $s_i = e^{i\phi_i}$. Here, i denotes the imaginary unit and i denotes index. we can write the Potts Hamiltonian as,

$$H_{K-cut} = - \sum_{i,j,i < j}^N J_{ij} \cdot \text{Re} \left(e^{if_{K-cut}(\Delta\phi_{ij})} s_i s_j^* \right) \quad (2.1)$$

where, $\Delta\phi_{ij} = \phi_i - \phi_j$, Re denotes real values; J_{ij} is the interaction coefficient between spin i and j . For a graph, $J_{ij} = -w_{ij}$, w_{ij} : edge weight. The phase sensitive coupling function $f_{K-cut}(\cdot)$ is defined as,

$$f_{K-cut}(\Delta\phi_{ij}) = \lim_{\sigma \rightarrow 0} \sum_{k=1}^{K-1} \left(\left((2k-1)\pi - \frac{2k\pi}{K} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2k\pi}{K})^2}{2\sigma^2} \right)} + \left(\frac{2k\pi}{K} - (2k-1)\pi \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2k\pi}{K})^2}{2\sigma^2} \right)} \right) \quad (2.2)$$

We will denote the general form of this function with $f_K(\cdot)$. Figure 2.1 shows spin assignments and $f_{K-cut}(\cdot)$ for different values of K .

Now, using Eq. (2.2), Eq. (2.1) can be written as,

$$H_{K-cut} = - \sum_{i,j,i < j}^N J_{ij} \cos \left(\Delta\phi_{ij} + f_{K-cut}(\Delta\phi_{ij}) \right) \quad (2.3)$$

# of Partitions (K)	Spin Assignment	$f_{K-cut}(\Delta\phi_{ij})$
2 (Max-Cut)	$1e^{i\pi}(\equiv -1)$ $1e^{i(2\pi)}(\equiv 1)$	
3	$1e^{i(\frac{2\pi}{3})}(\equiv -0.5 + i\frac{\sqrt{3}}{2})$ $1e^{i(\frac{4\pi}{3})}(\equiv -0.5 - i\frac{\sqrt{3}}{2})$ $1e^{i(\frac{6\pi}{3})}(\equiv 1)$	
4	$1e^{i(\frac{2\pi}{4})}(\equiv +i)$ $1e^{i(\frac{4\pi}{4})}(\equiv -1)$ $1e^{i(\frac{6\pi}{4})}(\equiv -i)$ $1e^{i(\frac{8\pi}{4})}(\equiv +1)$	
5	$1e^{i(\frac{2\pi}{5})}(\equiv 0.3 + 0.95i)$ $1e^{i(\frac{4\pi}{5})}(\equiv -0.8 + 0.58i)$ $1e^{i(\frac{6\pi}{5})}(\equiv -0.8 - 0.58i)$ $1e^{i(\frac{8\pi}{5})}(\equiv 0.3 - 0.95i)$ $1e^{i(\frac{10\pi}{5})}(\equiv 1)$	

Figure 2.1. Spin assignments and $f_{K-cut}(\cdot)$ for various K values.

Like OIM, oscillators map nodes and coupling elements map edges in this oscillator Potts machine. Furthermore, each Potts spin value represents a node set / partition. In Eq. (2.3), $-J_{ij}\cos(\Delta\phi_{ij} + f_{K-cut}(\Delta\phi_{ij})) = -1$, only if two spins / nodes are separated by $m\frac{2\pi}{K}$ where $m = 1, 2, \dots, (K - 1)$. So, the Hamiltonian will get a maximum reduction if two connected nodes are placed in two different partitions. Thus, minimization of H_{K-cut} will maximize the sum of the edge weights that connect two different partitions, i.e., the

ground state of H_{K-cut} maps the optimal Max-K-Cut. Now, the corresponding system dynamics and the energy function can be respectively written as,

$$\frac{d\phi_i(t)}{dt} = -C \sum_{j=1, j \neq i}^N J_{ij} \sin(\Delta\phi_{ij} + f_{K-cut}(\Delta\phi_{ij})) - \sum_{i=1}^N C_s \sin(K\phi_i(t)) \quad (2.4)$$

$$E(\phi(t)) = -\frac{KC}{2} \sum_{i,j, j \neq i}^N J_{ij} \cos(\Delta\phi_{ij} + f_{K-cut}(\Delta\theta_{ij})) - \sum_{i=1}^N C_s \cos(K\phi_i(t)) \quad (2.5)$$

While deriving Eq. (2.5), we used the relationship $-\frac{1}{K}(\nabla E)_i = \frac{d\phi_i}{dt}$ to ensure a gradient descent in the energy domain, i.e., $\frac{dE(\phi(t))}{dt} \leq 0$. Details can be found [82]. Fig. 2.2 shows

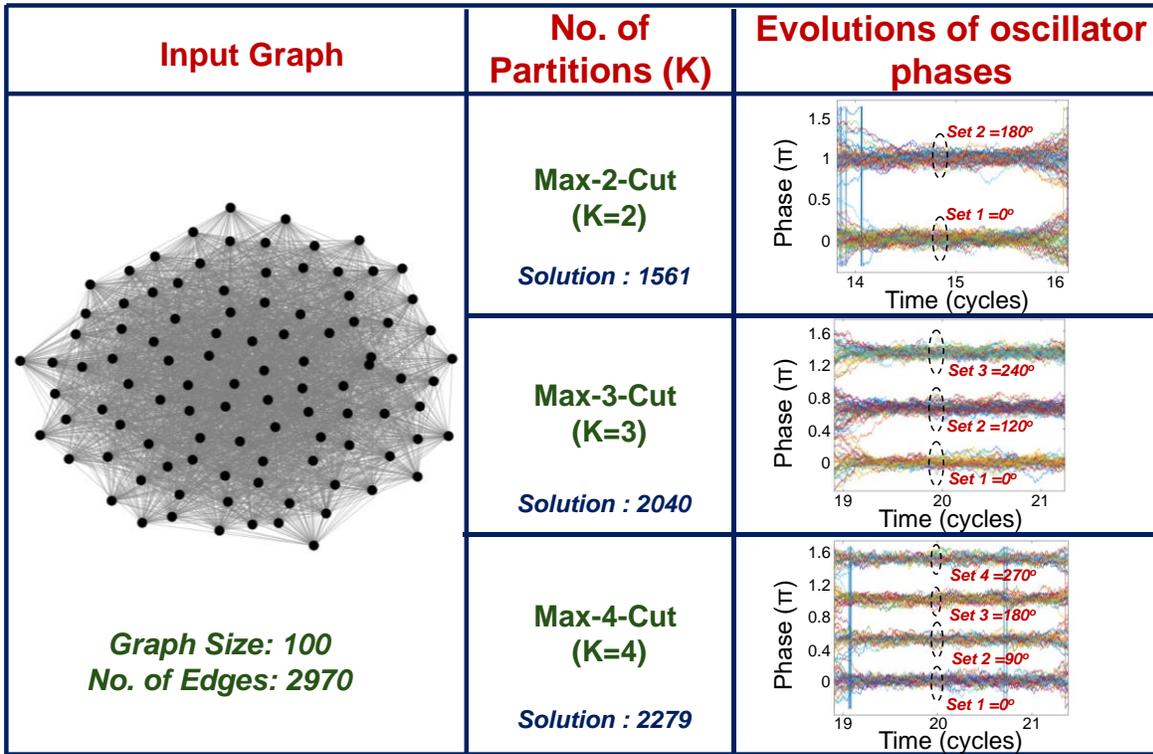


Figure 2.2. Illustration of the computation of Max-K-Cut (K=2,3,4) for a representative 100-node graph by simulating the oscillator Potts machine.

the Max-K-Cut solution of a representative graph, obtained by simulating the oscillator Potts machine.

To map other COPs, Eq. (2.4) can be modified by tailoring the $f_K(\cdot)$ function. Table 2.1 lists the $f_K(\cdot)$ function that can be used for solving multiple other COPs.

COP	$f_K(\cdot)$
Max-K-Cut, Graph Coloring, Maximum Independent Set (MIS)	$f_{K-cut}(\Delta\phi_{ij}) = \lim_{\sigma \rightarrow 0} \sum_{k=1}^{K-1} \left(\left((2k-1)\pi - \frac{2k\pi}{K} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2k\pi}{K})^2}{2\sigma^2} \right)} + \left(\frac{2k\pi}{K} - (2k-1)\pi \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2k\pi}{K})^2}{2\sigma^2} \right)} \right)$
Traveling Salesman Problem (TSP)	$f_{TSP}(\Delta\phi_{ij}) = \lim_{\sigma \rightarrow 0} - \sum_{\gamma=1, N-1} \left(\frac{2\gamma\pi}{N} \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2\gamma\pi}{N})^2}{2\sigma^2} \right)} + \left(-\frac{2\gamma\pi}{N} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2\gamma\pi}{N})^2}{2\sigma^2} \right)} \right) + \lim_{\sigma \rightarrow 0} \sum_{k=2, k \neq N-1}^N \left(\left(\pi - \frac{2k\pi}{N} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2k\pi}{N})^2}{2\sigma^2} \right)} + \left(\frac{2k\pi}{N} - \pi \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2k\pi}{N})^2}{2\sigma^2} \right)} \right)$
Hamiltonian Cycle Problem (HCP)	$f_{HC}(\Delta\phi_{ij}) = \lim_{\sigma \rightarrow 0} - \sum_{\gamma=1, N-1} \left(\left(\pi - \frac{2\gamma\pi}{N} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2\gamma\pi}{N})^2}{2\sigma^2} \right)} + \left(\frac{2\gamma\pi}{N} - \pi \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2\gamma\pi}{N})^2}{2\sigma^2} \right)} \right) + \lim_{\sigma \rightarrow 0} \sum_{k=2, k \neq N-1}^N \left(\left(\frac{\pi}{2} - \frac{2\pi k}{N} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} - \frac{2\pi k}{N})^2}{2\sigma^2} \right)} + \left(\frac{2\pi k}{N} - \frac{\pi}{2} \right) \cdot e^{-\left(\frac{(\Delta\phi_{ij} + \frac{2\pi k}{N})^2}{2\sigma^2} \right)} \right)$

Table 2.1. $f_K(\cdot)$ function for a few COPs.

Details of these formulations and results obtained from these models can be found in [82].

2.2. FPGA implementation of oscillator Potts machine

The oscillator Potts machine-based computational model to solve the Max-K-Cut problem is implemented as an algorithm on an FPGA accelerator [83]. In such an implementation, FPGA numerically solves the corresponding dynamics. We implement it on an AWS F1 instance. It leverages design techniques such as sparse matrix random access parallelization and uses an efficient dataflow architecture to accelerate the solution to the Max-K-Cut problem. By leveraging the inherent parallelism in the computational models and the FPGA implementation, we demonstrate the solutions to the Max-K-Cut problem (K=2,3,4) on graphs up to 10,000 nodes with speedups ranging from 17x - 390x over a state-of-the-art Ising machine-based accelerator, while maintaining similar solution quality.

2.2.a. Computational model implemented on FPGA

The computational model (i.e., the dynamics), shown in Eq. (2.4), is solved using a stochastic differential equation (SDE) solver developed on an FPGA platform. Here, SDE is used so that noise can be introduced into the system. Such noise helps escape local minima that result in suboptimal solutions. The dynamics that are solved, can be written as,

$$\frac{d\phi_i(t)}{dt} = -C \sum_{j=1, j \neq i}^N J_{ij} \tanh\left(k \sin\left(\Delta\phi_{ij} + f_{K-Cut}(\Delta\phi_{ij})\right)\right) - \sum_{i=1}^N C_s \sin(K\phi_i(t)) \quad (2.6)$$

$$+ A_n dw_t$$

where, ϕ represents oscillator phases, A_n is noise amplitude, dw_t is a Weiner process [84] which adds stochasticity to the system. We have also used $\tanh(\cdot)$ in the dynamics

so that it augments the phase dynamics as demonstrated in other works [85], [86]. A simple trapezoidal integration technique [87] is used to numerically solve the differential equation (Eq. (2.6)). After solving the dynamics, the evolution of oscillator phases (ϕ) is

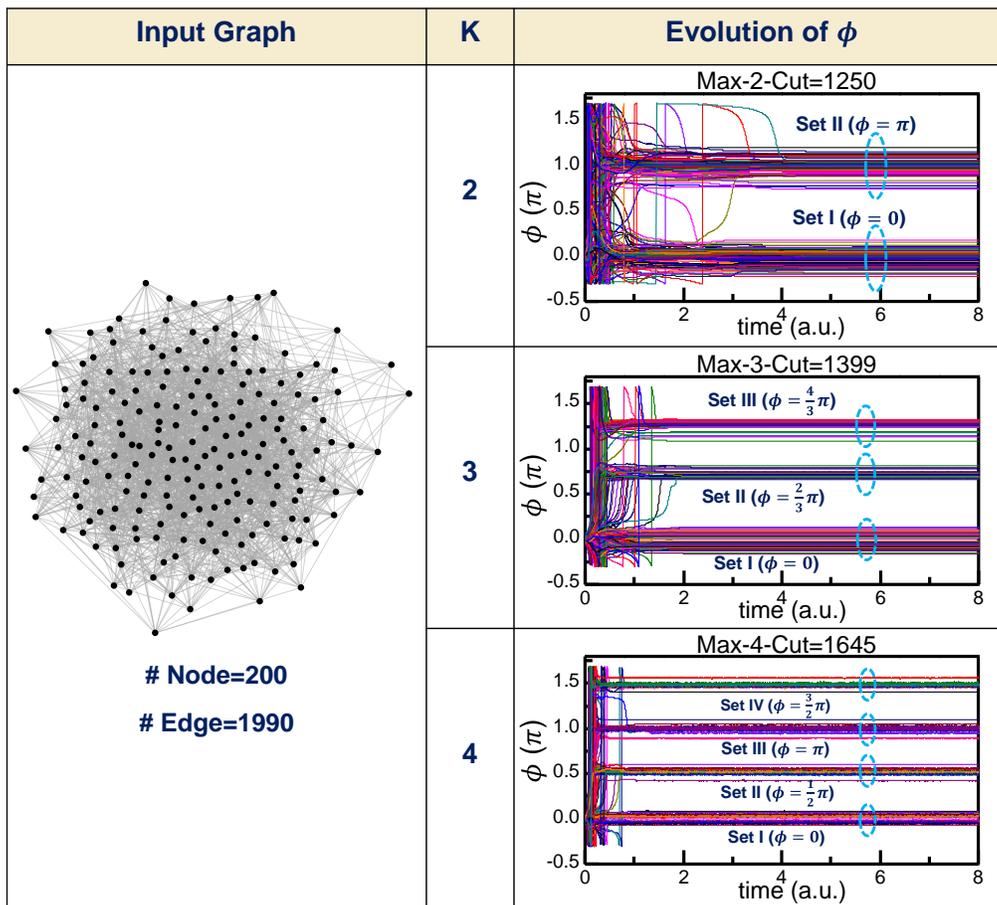


Figure 2.3. Illustrative example showing the Max-K-Cut solution obtained using the oscillator Potts machine-based FPGA accelerator for a 200-node graph. The solutions are calculated for $K=2$, $K=3$, and $K=4$.

obtained. Steady-state oscillator phases create K partitions encoding the K sets of the Max- K -Cut solution. Figure 2.3 illustrates the Max- K -Cut solutions of a 200-node representative graph, for $K=2,3$, and 4 , obtained using the FPGA-based Potts machine.

2.2.b. Architecture of the Potts machine-based FPGA accelerator

While designing the FPGA accelerator, the following objectives are prioritized: 1) leveraging a maximum level of parallelism; 2) consuming a reasonable amount of

hardware resources; 3) ensuring flexibility so that it supports the general case of the Max-K-Cut for any graph of arbitrary size without FPGA reconfiguration. The key components of the FPGA system design are described below:

Figure 2.4 shows the high-level block diagram of the FPGA accelerator. The Potts machine algorithm is implemented on the FPGA using two major kernels: (a) Graph Initializer; and (b) Kuramoto Kernel. The graph adjacency matrix in CSR (compressed row storage) format, is stored in the host DRAM. Subsequently, they are transferred to

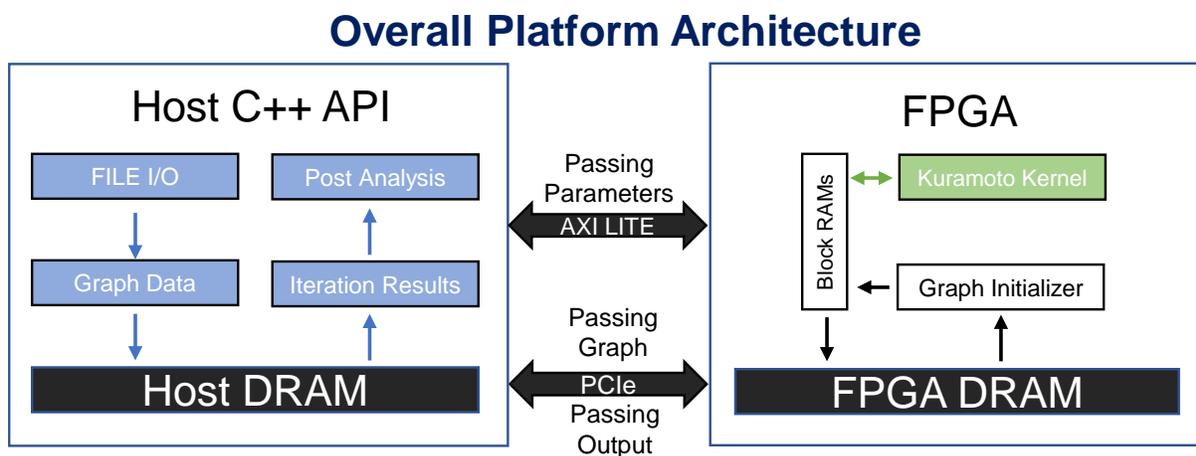


Figure 2.4. Block diagram depicting the architecture of the proposed FPGA-based Max-K-Cut solver.

the FPGA DRAM through a PCIe. A set of control registers can communicate with the FPGA through the AXI-lite kernel interface that allows dynamical modification of the essential parameters such as the graph size, K (in Max-K-Cut), number of iterations without the need for reconfiguring the FPGA platform each time. At first, the graph adjacency is transferred to the FPGA, and then the Graph Initializer kernel transfers it to the high-bandwidth on-chip Block RAMs (BRAMs). BRAMs allow parallel access. Hence, the off-chip DRAM is only read once. After initialization, the Kuramoto Kernel performs the computation. Table 2.2 lists key parameters used in the design.

Parameters	Definition	Parameter values used in the evaluation
N	Size of the graph	Input problem dependent (up to 10,000)
K (Max-K-Cut)	Number of partitions required from the Cut	Input problem dependent (2, 3, 4)
X	Size of each segment in Block RAM	80
n	Number of bits used to represent the fractional part in standard fixed-point format	19 bits (Fractional part)

Table 2.2. Parameters used in the FPGA implementation.

Details of the design can be found in [83]. The key features include a sparse matrix parallel access technique that parallelizes the computation of ϕ values in each iteration. LUTs are used for the sinusoidal function which significantly reduces the use of hardware resources and computation time. LUTs are also used for implementing the $f_{K-cut}(\cdot)$ function. If the K value is changed, the only operational change that is required is to change the LUT segment for the $f_{K-cut}(\cdot)$ function. Additionally, to generate noise, random numbers are generated using a standard Box-Muller Random Generator [88].

2.2.c. Results

2.2.c.1. Max-K-Cut solution

We evaluate the performance of our implementation using instances from the G-Set benchmark database. The database contains hard non-planar random graphs with a broad size range allowing us to solve graphs with sizes ranging from 800 to 10,000 nodes (specified in Fig. 2.5a). The oscillator dynamics are evaluated for 4000 iterations (epochs). During evaluation, we set the FPGA frequency to 100 MHz. We first compare our results (mean computation time) for the MaxCut case with two other GPU-based implementations, namely the MARS (Mean-field Annealing from a Random State)

algorithm [89] and PBBM (Population Based Boltzmann Machine) algorithm [90] that have also evaluated problems from the G-Set database. Furthermore, the FPGA-based implementations for solving MaxCut demonstrated in [91] and [92] have only focused on simpler planar and toroidal graphs. Additionally, they do not address the general case of the Max-K-Cut problem for $K > 2$ cases. Hence, we have not included these results in the benchmarking. From Fig. 2.5, it can be observed that our approach provides 18x and 2x mean speedup, respectively, compared to the MARS and the PBBM approach, while

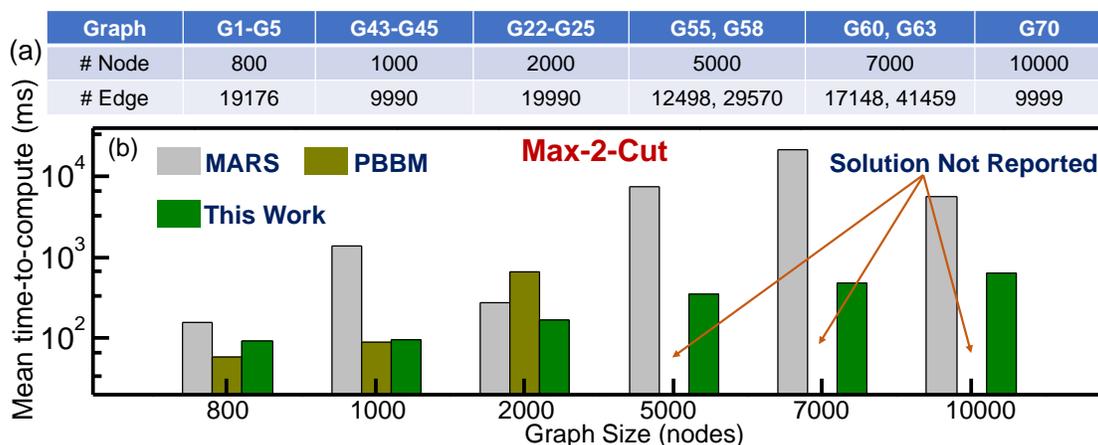


Figure 2.5. Comparison of our approach for the baseline MaxCut ($K=2$) with prior works. (a) Graph instances from the G-Set database used for benchmarking. (b) Comparison of the mean computation time for the GPU-based MARS algorithm and the GPU-based PBBM algorithm. Our approach exhibits solution quality comparable to that of the MARS approach. PBBM did not report average solution quality.

providing similar solution quality ($>98.5\%$); here, solution quality is defined as the ratio of the obtained solution to the best-known cut [93]. We note that none of the GPU, FPGA, and ASIC-based annealing approaches reported direct implementation (i.e., without preprocessing and auxiliary variables) of the broader Max-K-Cut ($K \geq 2$) problem. Current methods have to rely on transforming the Max-K-Cut problem to a binary optimization form (QUBO: quadratic unconstrained binary optimization) entailing additional nodes (auxiliary variables) so that it can be mapped to an Ising machine. We now compare our direct implementation (using the new models) with the Ising machine implementation for

the Max-K-Cut (after the problem transformation). We use the GPU-based simulated bifurcation (Ising) machine (SBM; from Toshiba and available on AWS [41]) for this comparison. While we experimentally evaluate the Max-3-Cut and Max-4-Cut solutions on the SBM, we use the SBM-based results reported for the MaxCut (K=2) [93].



Figure 2.6. Cumulative time-to-compute for solving the (a) Max-2-Cut, (b) Max-3-Cut, (c) Max-4-Cut on the G-set graph instances and their comparison with state-of-the-art SBM approach. In all cases, we maintain a high mean accuracy exceeding >95% of the best-known solution.

Fig. 2.6a presents a comparison of the cumulative computation time between our approach and the state-of-the-art GPU-based SBM approach [93] for solving the

archetypal Max-2-Cut problem over the G-set graphs (Fig. 2.5a). While for smaller graphs, computation time from our approach is comparable with the SBM approach, it provides a 20x speedup over the SBM for graphs exceeding 5000 nodes. We obtain >98.5% average solution quality for all the Max-2-Cut instances tested, while SBM showcased >99.9% solution quality. Next, we evaluate the computation time for the Max3-Cut and Max-4-Cut problems. The comparison of computation times for Max-3-Cut and Max-4-Cut are presented in Fig. 2.6b and Fig. 2.6c, respectively. It can be observed that our method achieves a remarkable ~510x and ~270x mean speedup compared to the SBM in solving the Max-3-Cut and Max-4-Cut problems, respectively (~390x average speedup for the Max-3-Cut and Max-4-Cut combinedly), while maintaining similar solution quality. In our approach, the time-to-compute primarily consists of the Kuramoto Kernel computation time, which accounts for most of the overall time-to-solution. We calculate the K-Cut values in the host system using the Kernel results obtained from the FPGA. Similarly, the SBM computation time consists of only the Ising problem computing time; the pre-processing (conversion to Ising problem) and post-processing (finding K-Cut solution from Ising solution) are performed in the host and their computation time is not added in the overall computation time presented here. Also, we are unable to solve the Max-3-Cut and Max-4-Cut for some of the larger graphs (here, 5000 to 10000 node graphs) using SBM since after converting them to Ising problems, their size exceeds 10,000 nodes that is the maximum limit for the SBM available on AWS.

We also analyze how time-to-compute scales with problem size and the value of K. Figure 2.7 presents the average time-to-compute as a function of graph size. It can be observed that as the number of nodes increases, the computation time scales linearly,

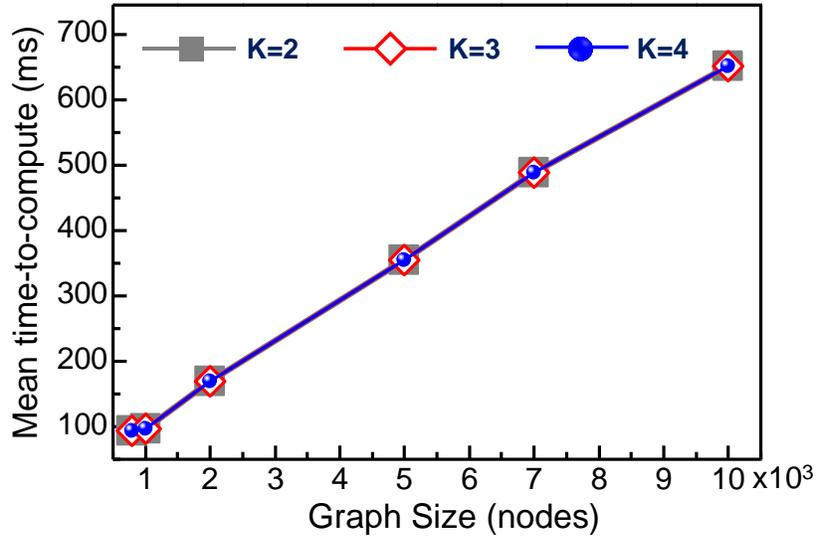


Figure 2.7. Mean time-to-compute as a function of graph size.

which can be attributed to the parallel access of each graph rows by the FPGA platform (i.e., row-wise parallelization). Most importantly, the computation time does not change with the value of K, unlike prior designs and implementations. It can be attributed to the computational model that facilitates the solution of the Max-K-Cut without increasing the problem size. Furthermore, if the K value changes, the only operational change that the FPGA system needs to make is switching the LUT for the $f_{K-Cut}(\cdot)$ function. Hence, the computation time remains the same regardless of the value of K for a particular graph.

2.2.c.2. Resource utilization and energy benchmark

Resource	Xilinx VU9P FPGA	Resource	Xilinx VU9P FPGA
Kernel Frequency	100 MHz	DSP	2.7K (38%)
Block RAMs	1.3K (30%)	Flip Flops	470K (19%)
Ultra RAMs	0.5K (50%)	Logic Slices	518K (43%)

Table 2.3. Overall resource utilization.

Table 2.3 presents a detailed overview of the resource utilization in the FPGA implementation. Table 2.4 compares the energy consumption between our FPGA implementation (collected using the AWS ‘FPGA image describe’ command) and the

SBM implementation used for benchmarking in this work. While the SBM energy data for the Max-2-Cut has been reported, the energy numbers for K=3, 4 are projected since the SBM energy data (on AWS) is unavailable. It can be observed that our approach not only offers better computational capability but also enables over 8x improvement in the energy consumption / iteration.

Approach	Platform	Problem Solved	Energy (mJ) / Iteration (2000 node)		
			Max-2-Cut	Max-3-Cut	Max-4-Cut
SBM [43]	GPU	MaxCut	3.44 (reported)	10.32 (projected)	13.76 (projected)
This Work	FPGA	Max-K-Cut & MaxCut	0.4225 (measured)	0.4225 (measured)	0.4225 (measured)

Table 2.4. Comparison with other approaches.

Chapter 3

3. Coupled Oscillators for Solving COPs with Higher Order Interaction

This chapter focuses on the formulation of novel oscillator-based computational models that can directly map COPs with higher order interaction.

3.1. Dynamical systems to solve COPs with higher order interaction

Ising machines can directly map COPs that can be represented by quadratic objective functions ($H = -\sum_{i,j}^N J_{ij}\sigma_i\sigma_j$). However, there is a larger class of problems such as Boolean satisfiability and integer factorization among others wherein the objective functions have a degree greater than two. Such problems entail the use of hypergraphs for their representation and analysis (Fig. 1.5). A hypergraph can be considered as a generalization of graphical data structures wherein an edge (known as a hyperedge) can connect any number of vertices; this is in contrast to a graph where an edge can join a maximum of two vertices. Analog models for solving combinatorial problems in hypergraphs have been relatively less explored [94]-[96]. We note that such problems can, in theory, be reduced to problems that have objective functions with quadratic degrees [97], [98]. However, this typically involves the introduction of additional ancillary variables (additional nodes/variables) which can effectively increase the size of the (quadratic degree) combinatorial problem that must then be solved [99], [78]. Therefore, in this research, the goal is to formulate analog computational models for solving such problems without introducing ancillary / auxiliary variables [86], [100].

The developed approach builds on the foundational work performed by Ercsey-

Ravasz et al. [94], wherein the authors proposed an approach for solving the Boolean Satisfiability (SAT) problem using continuous (analog) variables. The SAT problem is defined as the challenge of evaluating a Boolean assignment (1 or 0) that will satisfy a Boolean formula expressed in the conjunctive normal form (CNF); $Y = C_1 \wedge C_2 \wedge \dots \wedge C_M$. A SAT problem with N variables and M clauses can be represented by a hypergraph of N nodes and M hyperedges where the hyperedge 'connects' all the nodes in a clause.

The decision version of the problem evaluates if such an assignment exists. We formulate a new dynamical system to solve the SAT problem. Subsequently, building on the method developed by Ercsey-Ravasz et al. [94], computational models are formulated for: (a) the NAE (Not-All-Equal) SAT problem, which is an NP-complete variant of the SAT problem. Besides finding an assignment for the Boolean variables such that every clause is satisfied, the NAE-SAT problem also requires that at least one literal in every clause is false. Further, the computational model for the NAE-SAT problem can be extended to the Set Splitting problem, which evaluates if there exists a partition that splits a finite set into two parts such that all the subsets of the finite set are split by the partition. The Set Splitting problem is a special case of the NAE-SAT problem wherein all the variables in the normal form (positive NAE-SAT); (b) Integer factorization problem, considered here as the problem of dividing a number into two integer factors; we note that directly representing the above problems entails the use of hypergraphs; (c) The Graph Isomorphism problem, which evaluates if two graphs with the same number of edges and vertices (non-trivial case) have the same edge connectivity. (d) Finally, we show that the proposed approach can be used to minimize the Ising Hamiltonian (quadratic optimization problem), and in fact, provides an alternate dynamical system formulation to the well-

known oscillator-based dynamical system proposed earlier [85]. Subsequently, using this formulation, we show its application in solving the archetypal Maximum Cut (MaxCut) problem, defined as the challenge of dividing the nodes of a graph into two sets such that the number of shared edges (among the two sets) is maximized.

3.1.a. Boolean SAT

We first consider the Boolean SAT problem where we represent each variable x_i in the Boolean expression by $\gamma_i \equiv \frac{1+\cos(\alpha_i)}{2}$, where α_i is an analog variable. The $\cos(\cdot)$ function sets the bounds of γ_i to $[0,1]$, and ensures that the Boolean variable and its analog counterpart have the same value at the maxima and the minima of the analog variable function. We note that while the above formulation resembles a (level-shifted) oscillator described by the general form $\gamma_i \equiv \frac{1+\cos(\omega t+\alpha_i)}{2}$, the two are not exactly equivalent since the ‘ ωt ’ term (oscillating term) is not considered here; with the ‘ ωt ’ term, the dynamics of the system do not directly map to the objective function of the SAT problem (instead they can be mapped to the dynamics of the NAE-SAT (not-all-equal SAT) problem as shown in [100]). Nevertheless, we will refer to α as a ‘phase’ for simplicity. For each clause C_m , we define $K_m(\alpha) = \prod_{i=1}^N \left(1 - \left(\frac{1+c_{mi}\cos(\alpha_i)}{2}\right)\right)$, $c_{mi} = 1(-1)$, if x_i in the m^{th} clause appears in the normal (negated) form, respectively; $c_{mi} = 0$, if x_i is absent from the m^{th} clause. $K_m(\alpha)$ can be considered as an analog equivalent of $1 - C_m$, and exhibits the property that $K_m = 0$ if and only if the clause is satisfied ($C_m = 1$), i.e., at least one variable is TRUE. We define a continuous time dynamical system given by $\frac{d\alpha}{dt} = F = (-\nabla_{\alpha} V)$, which has an energy function given by:

$$V = \sum_{m=1}^M A(K_m(\alpha))^2 \quad (3.1)$$

where $A (> 0)$ is a constant. It can be observed from Eq. (3.1) that V is minimized by maximizing the number of satisfied clauses. Further, $V = 0$ is the global minima of the function and is attained when all the clauses are satisfied i.e., $K_m = 0$ for $m = 1, 2, \dots, M$.

To show that the energy function defined in Eq. (3.1) decreases with time i.e., $\frac{dV}{dt} \leq 0$.

$$\frac{dV}{dt} = \left(2A \sum_{m=1}^M K_m(\alpha) \cdot \left[\frac{c_{mi}K_m(\alpha)}{1 - c_{mi} \cos(\alpha_i)} \right] \sin(\alpha_i) \right) \cdot \left(\frac{d\alpha_i}{dt} \right) \quad (3.2)$$

Further,

$$\frac{dV}{d\alpha_i} = 2A \sum_{m=1}^M K_m(\alpha) \left[\frac{c_{mi}K_m(\alpha)}{1 - c_{mi} \cos(\alpha_i)} \right] \sin(\alpha_i) \quad (3.3)$$

It can be observed from Eq. (3.2) and (3.3) that the term in the first bracket on the righthand side of Eq. (3.2) is equal to $\frac{dV}{d\alpha_i}$. Further, $\frac{dV}{d\alpha_i} = -\frac{d\alpha_i}{dt}$. Substituting these terms into Eq. (3.2), $\frac{dV}{dt}$ can be expressed as

$$\frac{dV}{dt} = -\left(\frac{d\alpha_i}{dt} \right)^2 \leq 0 \quad (3.4)$$

Eq. (3.4) shows that $\frac{dV}{dt} \leq 0$, that implies that the system always evolves to minimize V (energy), or in other words, maximize the number of satisfied clauses. The corresponding dynamics can be computed as:

$$\frac{d\alpha_i}{dt} = (-\nabla_{\alpha} V)_i = \sin(\alpha_i) \cdot \left(- \sum_{m=1}^M \left[2AK_m(\alpha) \left[\frac{c_{mi}K_m(\alpha)}{1 - c_{mi} \cos(\alpha_i)} \right] \right] \right) \quad (3.5)$$

Figures 3.1(a),(b) show a representative Boolean SAT problem with 6 variables and 10

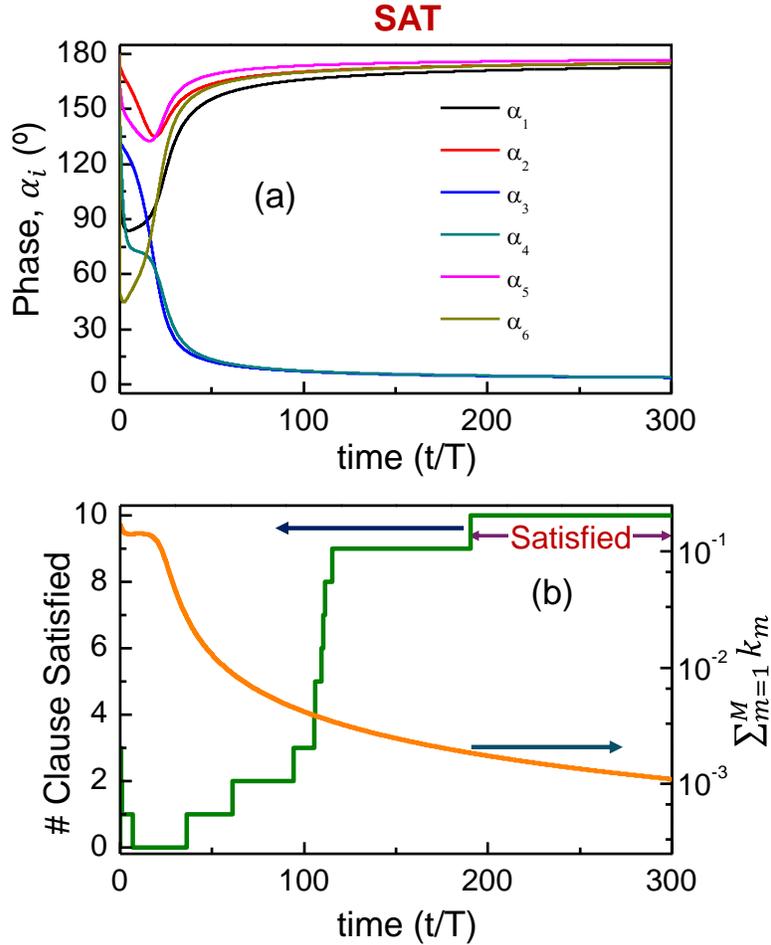


Figure 3.1. (a) Evolution of the phases (α_i), (b) the number of satisfied clauses and $\sum_{m=1}^M K_m$, respectively, as a function of time for a Boolean SAT problem. Since numerical methods are used for solving the dynamics, a threshold value of $K_m < 4 \times 10^{-4}$ was used for a clause to be considered as TRUE (satisfied). The Boolean expression considered in this illustrative example consists of 6 variables and 10 clauses ($Y = (\bar{x}_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee x_6) \wedge (\bar{x}_2 \vee x_5) \wedge (\bar{x}_2 \vee x_6) \wedge (x_1 \vee \bar{x}_6) \wedge (x_2 \vee \bar{x}_5 \vee x_6) \wedge (\bar{x}_1 \vee \bar{x}_4) \wedge (\bar{x}_4 \vee \bar{x}_6) \wedge (\bar{x}_5 \vee x_6)$).

clauses solved using the above computational model. It can be observed that the system minimizes $K_m(\alpha)$ which subsequently maximizes the number of clauses satisfied. We once again acknowledge that this formulation derives strong inspiration from the elegant analog dynamics formulated by Ercsey-Ravasz et al. [94]. We formulate an alternative dynamical system to solve the SAT problem which is described below:

Alternative dynamical system to solve SAT. To formulate an alternative dynamical system [100], we represent every variable x_i in the Boolean expression using an analog variable ϕ_i (similar to oscillator phase), where $x_i = \frac{1+\cos(t+\phi_i)}{2}$, which can be considered as a level-shifted oscillator with an angular frequency ω , that is assumed to be $\omega = 1$ in this theoretical analysis. The relationship between x_i and α_i ($x_i = \frac{1+\cos(t+\phi_i)}{2}$) is defined such that the maximum (or minimum) value of the analog variable equals the Boolean assignment for $x_i \in \{0,1\}$, respectively. For each clause C_m , we define $K_{m,osc}(t, \alpha) = \prod_{i=1}^N \left(1 - \left(\frac{1+c_{mi}\cos(t+\phi_i)}{2}\right)\right)$, where $c_{mi} = 1(-1)$, if the i^{th} variable appears in the m^{th} clause in the normal (negated) form; $c_{mi} = 0$, if the variable is absent from the m^{th} clause; $\phi = [\phi_1 \phi_2 \dots \phi_N]$; N is the number of variables in the SAT problem. It can be observed that $K_{m,osc}(t, \phi) = 0$, if and only if the clause is satisfied. We define the dynamical system: $(-\nabla_{\phi} V)_i = 1 + \frac{d\phi_i}{dt}$. The energy function for the system is defined as:

$$V = \sum_{m=1}^M A \left(K_{m,osc}(t, \phi) \right)^2 \quad (3.6)$$

Here, M is the total number of clauses in the problem. $V = 0$ when all the clauses are satisfied, and consequently, corresponds to the solution of the SAT problem (if the problem is satisfiable). To evaluate the temporal evolution of the system energy, we calculate $\frac{dV}{dt}$, which is given by:

$$\frac{dV}{dt} = \sum_{i=1}^N \left(\frac{\partial V}{\partial \phi_i} \right) \left(\frac{d\phi_i}{dt} \right) + \frac{\partial V}{\partial t} \quad (3.7)$$

Using Eq. (3.6) and the definition of $K_{m,osc}(t, \phi)$, we can calculate $\frac{\partial V}{\partial t}$ as,

$$\begin{aligned}
\frac{\partial V}{\partial t} &= \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{\partial (K_{m,osc}(t, \phi))}{\partial t} \right) \\
&= \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \left(\sum_{i=1}^N \frac{c_{mi}K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \right) \right) \quad (3.8) \\
&= \sum_{i=1}^N \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{c_{mi}K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \right)
\end{aligned}$$

Further, $\frac{\partial V}{\partial \phi_i}$ can be calculated as,

$$\begin{aligned}
\frac{\partial V}{\partial \phi_i} &= \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{\partial (K_{m,osc}(t, \phi))}{\partial \phi_i} \right) \\
&= \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{c_{mi}K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \right) \quad (3.9)
\end{aligned}$$

Substituting Eq. (3.9) into Eq. (3.8), $\frac{\partial V}{\partial t}$ can be expressed as,

$$\frac{\partial V}{\partial t} = \sum_{i=1}^N \frac{\partial V}{\partial \phi_i} \quad (3.10)$$

By substituting the expression for $\frac{\partial V}{\partial t}$ from Eq. (3.10) into Eq. (3.7), $\frac{dV}{dt}$ can be calculated

as,

$$\frac{dV}{dt} = \sum_{i=1}^N \left(\frac{\partial V}{\partial \phi_i} \right) \left(\frac{d\phi_i}{dt} \right) + \frac{\partial V}{\partial t} = \sum_{i=1}^N \left(\frac{\partial V}{\partial \phi_i} \right) \left(\frac{d\phi_i}{dt} \right) + \sum_{i=1}^N \frac{\partial V}{\partial \phi_i} = \sum_{i=1}^N \left(\frac{\partial V}{\partial \phi_i} \right) \left(1 + \frac{d\phi_i}{dt} \right) \quad (3.11a)$$

Further, utilizing the system dynamics $(-\nabla_{\phi} V)_i = 1 + \frac{d\phi_i}{dt}$ (defined above), Eq. (3.11a)

can be expressed as,

$$\frac{dV}{dt} = \sum_{i=1}^N \left(\frac{\partial V}{\partial \phi_i} \right) \left(1 + \frac{d\phi_i}{dt} \right) = - \sum_{i=1}^N \left(1 + \frac{d\phi_i}{dt} \right) \left(1 + \frac{d\phi_i}{dt} \right) = - \sum_{i=1}^N \left(1 + \frac{d\phi_i}{dt} \right)^2 \quad (3.11b)$$

It can be observed from Eq. 3.11(b) that V is a decreasing function with time since $\frac{dV}{dt} \leq 0$.

Consequently, this implies that the corresponding system dynamics will evolve to reduce system energy (V).

In order to formulate the system dynamics $\frac{d\phi_i}{dt}$, we express $\frac{dV}{dt}$ as,

$$\frac{dV}{dt} = \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{d(K_{m,osc}(t, \phi))}{dt} \right) \quad (3.12a)$$

$$\frac{dV}{dt} = \sum_{m=1}^M \left(2AK_{m,osc} \sum_{i=1}^N \left(c_{mi} \frac{K_{m,osc}}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \left(1 + \frac{d\phi_i}{dt} \right) \right) \right) \quad (3.12b)$$

$$\frac{dV}{dt} = \sum_{i=1}^N \left(\sum_{m=1}^M \left(2AK_{m,osc} c_{mi} \frac{K_{m,osc}}{1 - c_{mi} \cos(t + \phi_i)} \right) \right) \sin(t + \phi_i) \left(1 + \frac{d\phi_i}{dt} \right) \quad (3.12c)$$

Equating (3.11b) and (3.12c), we get

$$-\left(1 + \frac{d\phi_i}{dt} \right) = \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{c_{mi} K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \right) \quad (3.13a)$$

Eq. (3.13a) can be rewritten as,

$$\frac{d\phi_i}{dt} = - \left(\sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{c_{mi} K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \phi_i)} \sin(t + \phi_i) \right) + 1 \right) \quad (3.13b)$$

Eq. (3.13b) describes the phase dynamics of the system which computes the SAT solutions. The first term on the RHS in Eq. (3.13b) represents the dissipative component

of the system dynamics. The right-hand side in Eq. (3.13b) is 2π periodic in time. At

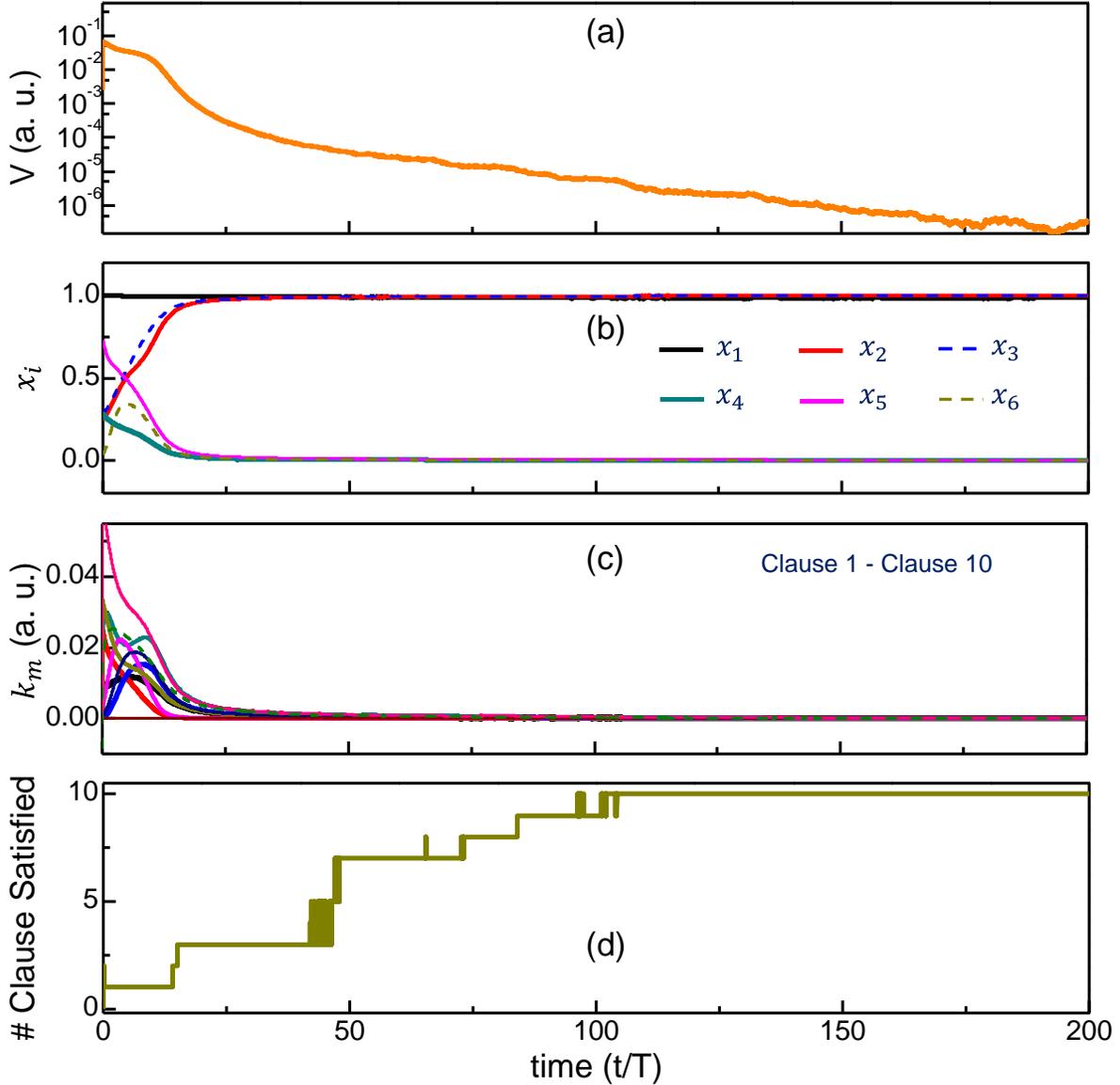


Figure 3.2. Evolution of (a) V , (b) x_i , (c) k_m , and (d) No. of clauses satisfied with time for an illustrative 3-SAT problem with 6-variables and 10-clauses that is computed using the alternative dynamical system to solve the SAT. Here, $\omega = 2\pi$ is used such that $T = 1$. The simulation is performed using a stochastic differential equation framework.

steady state, $V = 0$; $\frac{d\phi_i}{dt} = -1$, which implies that $\phi_i = -t + c_i$, with c_i being a constant offset in the time varying phase that assumes a value in $\{0, \pi\}$ in a way that minimizes the total system energy, and solves the SAT problem. A node i (defined by $\frac{1+\cos(t+\phi_i)}{2}$) will

eventually settle to 1 (when $c_i = 0$) or 0 (when $c_i = \pi$). Thus, the system is designed such that the out-of-phase feedback essentially ‘cancels’ out the oscillations when the system achieves the ground state energy. This corresponds to all the clauses being satisfied (if the problem is satisfiable). Figure 3.2 illustrates the evolution of the system dynamics and the corresponding solution for a representative SAT problem. Details of the simulation framework will be discussed at the end of Section 3.2.

3.1. b. Boolean NAE-SAT

The NAE-SAT problem is an NP-complete variant of the SAT problem with the added constraint that every clause must contain a literal that is true and false. To evaluate the NAE-SAT problem for a Boolean expression $Y = C_1 \wedge C_2 \wedge \dots \wedge C_M$, each clause $C_i = (x_1 \vee \bar{x}_2 \vee x_3 \vee \dots \vee x_N)$ in the original expression can be modified to $C_{NAE,i} = (x_1 \vee \bar{x}_2 \vee x_3 \vee \dots \vee x_N) \cdot (\overline{x_1 \wedge \bar{x}_2 \wedge x_3 \wedge \dots \wedge x_N}) \equiv C_i \cdot S_i$, where S_i is the negation of the conjunction of all the literals in that clause. While C_i imposes the condition that at least one literal must be true, S_i imposes the added constraint that at least one literal must be false in order that $C_{NAE,i} = 1$ (TRUE); an example of this is shown below (Table 3.1):

x_1	x_2	x_3	x_4	C_i (SAT)	S_i	$C_{NAE-SAT} = C_i \cdot S_i$ (NAE-SAT)
0	0	0	0	1	1	1
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	0	1	1	0	1	0
0	1	0	0	1	1	1
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0 (Additional constraint imposed by NAE-SAT)
1	1	0	1	1	1	1

1	1	1	0	1	1	1
1	1	1	1	1	1	1

Table 3.1. Illustrative example showing the reformulation of the clause $C_i = (x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4)$ for NAE-SAT.

Thus, the NAE-SAT problem can be expressed as evaluating if the expression $Y_{NAE} = C_{NAE,1} \wedge C_{NAE,2} \wedge \dots \wedge C_{NAE,M}$ can be made TRUE. To define the computational model for this problem, we again define an energy function with an analog variable α (similar to that of the SAT problem):

$$V = \sum_{m=1}^M A \left(K_{m,NAE}(\alpha) \right)^2 \quad (3.14)$$

albeit with a different analog formulation for each clause. $K_{m,NAE}(\alpha)$ is now defined as:

$$K_{m,NAE}(\alpha) = \left[\prod_{i=1}^N \left(1 - \left(\frac{1 + c_{mi} \cos(\alpha_i)}{2} \right) \right) \right] + \left[\prod_{i=1}^N \left(\frac{1 + c_{mi} \cos(\alpha_i)}{2} \right) \right] \quad (3.15a)$$

$$K_{m,NAE}(\alpha) = K_m^1(\alpha) + K_m^2(\alpha) \quad (3.15b)$$

Here, $K_m^1(\alpha)$ is similar to the $K_m(\alpha)$ defined for the SAT problem, and essentially captures the constraint that the contribution of that clause to the energy function is zero when the clause is satisfied. $K_m^2(\alpha)$ is formulated to define the additional constraint for the NAE-SAT problem entailing that all the literals cannot be equal to each other. Together, the formulation of $K_{m,NAE}(\alpha)$ for the NAE SAT clause ensures that it's contribution to the energy function is zero only when the clause is satisfied i.e., at least one literal is true, and all the literals are not equal to each other. The latter condition essentially ensures that at least one literal must be false. The corresponding system dynamics can be defined by:

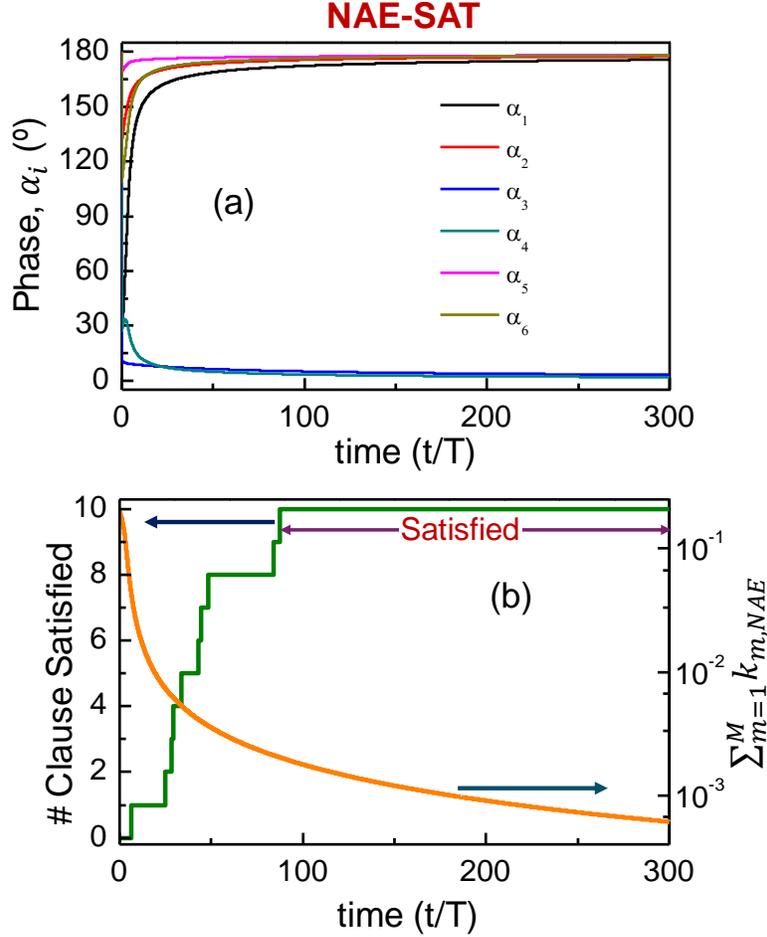


Figure 3.3. (a) Evolution of the phases (α_i), (b) the number of satisfied clauses and $\sum_{m=1}^M K_{m,NAE}$, respectively, as a function of time for a Boolean NAE-SAT problem. Since numerical methods are used for solving the dynamics, a threshold value of $K_{m,NAE} < 4 \times 10^{-4}$ was used for a clause to be considered as TRUE (satisfied). The Boolean expression considered in this illustrative example consists of 6 variables and 10 clauses ($Y = (\bar{x}_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee x_6) \wedge (\bar{x}_2 \vee x_5) \wedge (\bar{x}_2 \vee x_6) \wedge (x_1 \vee \bar{x}_6) \wedge (x_2 \vee \bar{x}_5 \vee x_6) \wedge (\bar{x}_1 \vee \bar{x}_4) \wedge (\bar{x}_4 \vee \bar{x}_6) \wedge (\bar{x}_5 \vee x_6)$).

$$\frac{d\alpha_i}{dt} = (-\nabla_{\alpha} V(\alpha))_i = - \sum_{m=1}^M 2A K_{m,NAE}(\alpha) \left(\frac{dK_{m,NAE}(\alpha)}{d\alpha_i} \right) \quad (3.16a)$$

where

$$\frac{dK_m(\alpha_i)}{d\alpha_i} = \frac{-c_{mi} K_m^1(\alpha)}{1 - c_{mi} \cos(\alpha_i)} \cdot (-\sin(\alpha_i)) + \frac{c_{mi} K_m^2(\alpha)}{1 + c_{mi} \cos(\alpha_i)} \cdot (-\sin(\alpha_i)) \quad (3.16b)$$

$$= \left[\frac{-c_{mi}K_m^1(\alpha)}{1 - c_{mi} \cos(\alpha_i)} + \frac{c_{mi}K_m^2(\alpha)}{1 + c_{mi} \cos(\alpha_i)} \right] \cdot (-\sin(\alpha_i)) \quad (3.16c)$$

$$\frac{d\alpha_i}{dt} = \sin(\alpha_i) \left(- \sum_{m=1}^M 2A K_m(\alpha) \left[\frac{c_{mi}K_m^1(\alpha)}{1 - c_{mi} \cos(t + \alpha_i)} - \frac{c_{mi}K_m^2(\alpha)}{1 + c_{mi} \cos(t + \alpha_i)} \right] \right) \quad (3.16d)$$

Figures 3.3(a),(b) show a representative example of an NAE-SAT expression (with 6 variables and 10 clauses) solved using the above computing model.

3.1.c. Set splitting

Given a finite set S , where S_1, S_2, \dots, S_N are the subsets, the objective of the Set Splitting problem is to evaluate if there exists a partition that divides all the subsets into two parts. This problem is equivalent to computing the solution of the positive NAE-SAT i.e., with only normal variables. To establish the relationship between the Set Splitting problem and the NAE-SAT problem, each element in the set can be represented by a variable x_i ; $x_i = 1(0)$, if x_i lies in Set I (II) (or vice-versa). We note that only variables in the normal form are needed. Subsequently, each subset S_i of the finite set can be mapped to $C_{NAE,i}$. It can be observed that only if the set is split (i.e., some nodes of S_i lie in Set I and II each) by the partition, $C_{NAE,i}$ evaluates to 1; if the nodes of a subset S_i lie entirely in Set I or II, $C_{NAE,i} = 0$. A partition that splits all the subsets exists when all C_{NAE} are satisfied, i.e., $V = 0$.

3.1.d. Integer factorization

The integer factorization problem is an NP complete problem that entails finding the integer factors of a number. Here, we consider the challenge of dividing a number F into two factors X and Y such that $XY = F$, or in other words, $XY - F = 0$. Expressing the

factors X and Y in binary form, this relationship can be used to formulate an energy function:

$$V = A \left(\left[\sum_{i=1}^N 2^{i-1} \left(\frac{1 + \tanh(k \cos(\alpha_i))}{2} \right) \right] \left[\sum_{j=N+1}^{2N} 2^{j-N-1} \left(\frac{1 + \tanh(k \cos(\alpha_j))}{2} \right) \right] - F \right)^2 \quad (3.17)$$

where each binary bit in X and Y is represented by $\left(\frac{1 + \tanh(k \cos(\alpha_{i,j}))}{2} \right)$; F is the integer number to be factorized ($F = \sum_{i=1}^N 2^{i-1} F_i$), α_i and α_j are used to represent the bits in X and Y , respectively, and k essentially decides the ‘steepness’ of the $\tanh(\cdot)$ function. This formulation of the energy function is inspired from that adopted by Borders et al. [101] and it can be observed that the energy function is expressed as a ‘product of sums’, instead of the ‘sum of products’ used in the formulation for the SAT and the NAE-SAT problems. The corresponding system dynamics are given by:

$$\begin{aligned} \frac{d\alpha_i}{dt} &= (-\nabla_{\alpha} V(\alpha))_i \\ &= \sin(\alpha_i) 2A \left(\left[\sum_{j=1}^N 2^{j-1} \left(\frac{1 + \tanh(k \cos(\alpha_j))}{2} \right) \right] \left[\sum_{m=N+1}^{2N} 2^{m-N-1} \left(\frac{1 + \tanh(k \cos(\alpha_m))}{2} \right) \right] - F \right) \\ &\quad \left(\sum_{n=N+1}^{2N} 2^{n-N-1} \left(\frac{1 + \tanh(k \cos(\alpha_n))}{2} \right) \right) 2^{i-2} k \operatorname{sech}^2(k \cos(\alpha_i)) \end{aligned} \quad (3.18a)$$

$$\begin{aligned} \frac{d\alpha_j}{dt} &= (-\nabla_{\alpha} V(\alpha))_j \\ &= \sin(\alpha_j) 2A \left(\left[\sum_{i=1}^N 2^{i-1} \left(\frac{1 + \tanh(k \cos(\alpha_i))}{2} \right) \right] \left[\sum_{m=N+1}^{2N} 2^{m-N-1} \left(\frac{1 + \tanh(k \cos(\alpha_m))}{2} \right) \right] - F \right) \\ &\quad \left(\sum_{n=1}^N 2^{n-1} \left(\frac{1 + \tanh(k \cos(\alpha_n))}{2} \right) \right) 2^{j-N-2} k \operatorname{sech}^2(k \cos(\alpha_j)) \end{aligned} \quad (3.18b)$$

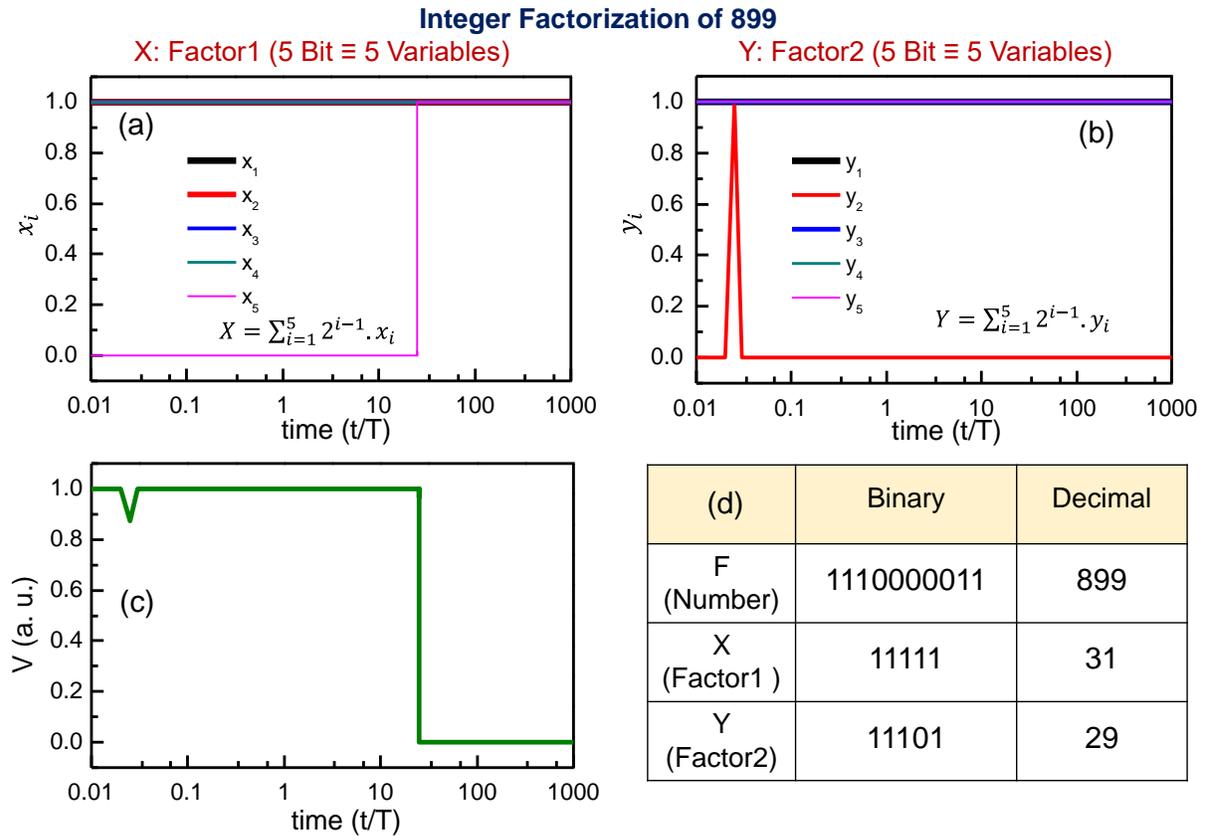


Figure 3.4. Integer factorization of 899. Temporal evolution of: (a) (b) the variables corresponding to bits in the integer factors X and Y , respectively; (c) Energy (V). (d) Integer factors X and Y computed by the system expressed in binary and decimal form.

Figure 3.4 presents an illustrative example showing the integer factorization of 899 performed using the above model. We note that the $\tanh(\cdot)$ function used in the analog formulation of the bits of the factors X and Y helps to effectively ‘binarize’ the output of the $\cos(\cdot)$ function. This is because the energy function (without the $\tanh(\cdot)$ function) may not always converge to *integer* factors of Z , i.e., $V = 0$, may also be achieved when $\cos(\alpha_{i,j}) \neq 1$ or -1 , resulting in non-integer factors. The $\tanh(\cdot)$ function helps drive the phases towards 0 ($\cos(\alpha_{i,j}) = 1$) or π ($\cos(\alpha_{i,j}) = -1$). This can be understood by considering the $\text{sech}^2(\cdot)$ function (arising from the $\tanh(\cdot)$ term in the energy function) in

the resulting dynamical system (Eq. (3.18b)) – the $\text{sech}^2(\cdot)$ function achieves a maximum (=1) when the (resulting) input to the function is zero (i.e., $\cos(\alpha_{i,j}) = 0$; $\alpha_{i,j} = \pm \frac{\pi}{2}$, and the corresponding ‘bit’ achieves a value of 0.5), and decays asymptotically towards zero as the input deviates from zero (i.e., $\text{sech}^2(\cdot)$ reduces as $\alpha_{i,j} \rightarrow 0$ ($\cos(\alpha_{i,j}) \rightarrow 1$) and $\alpha_{i,j} \rightarrow \pi$ ($\cos(\alpha_{i,j}) \rightarrow -1$). This implies that the function selectively reduces the perturbation as phases settle towards $\alpha_{i,j} = 0$ and π . This impact of using the $\tanh(\cdot)$ function is illustrated below:

Integer Factorization without $\tanh(\cdot)$ in the dynamics:

Here, we evaluate the system dynamics for computing the integer factors of 899 without considering the $\tanh(\cdot)$ function in the description of the system energy (Eq. (3.17)). The resulting dynamics (without the $\tanh(\cdot)$ function) are:

$$\begin{aligned} \frac{d\alpha_i}{dt} = (-\nabla_{\alpha} V(\alpha))_i = & \\ \sin(\alpha_i) \left(2A \left(\left(\sum_{j=1}^N 2^{j-1} \left(\frac{1 + \cos(\alpha_j)}{2} \right) \right) \left(\sum_{m=N+1}^{2N} 2^{m-N-1} \left(\frac{1 + \cos(\alpha_m)}{2} \right) \right) \right) \right. & \\ \left. - F \left(\sum_{n=N+1}^{2N} 2^{n-N-1} \left(\frac{1 + \cos(\alpha_n)}{2} \right) \right) \cdot 2^{i-2} \right) & \end{aligned} \quad (3.19)$$

$$\frac{d\alpha_j}{dt} = (-\nabla_{\alpha} V(\alpha))_j =$$

$$\sin(\alpha_j) \left(2A \left(\left(\sum_{i=1}^N 2^{i-1} \left(\frac{1 + \cos(\alpha_i)}{2} \right) \right) \left(\sum_{m=N+1}^{2N} 2^{m-N-1} \left(\frac{1 + \cos(\alpha_m)}{2} \right) \right) \right. \right.$$

$$\left. \left. - F \right) \left(\sum_{n=1}^N 2^{n-1} \left(\frac{1 + \cos(\alpha_n)}{2} \right) \right) \cdot 2^{j-N-2} \right) \quad (3.20)$$

Figure 3.5 shows the resulting dynamics of the system when computing the integer factors of the number 899. It can be observed that without the $\tanh(\cdot)$ function, the system can get stabilized when $\cos(\alpha) \neq \pm 1$, resulting in non-integer solutions.

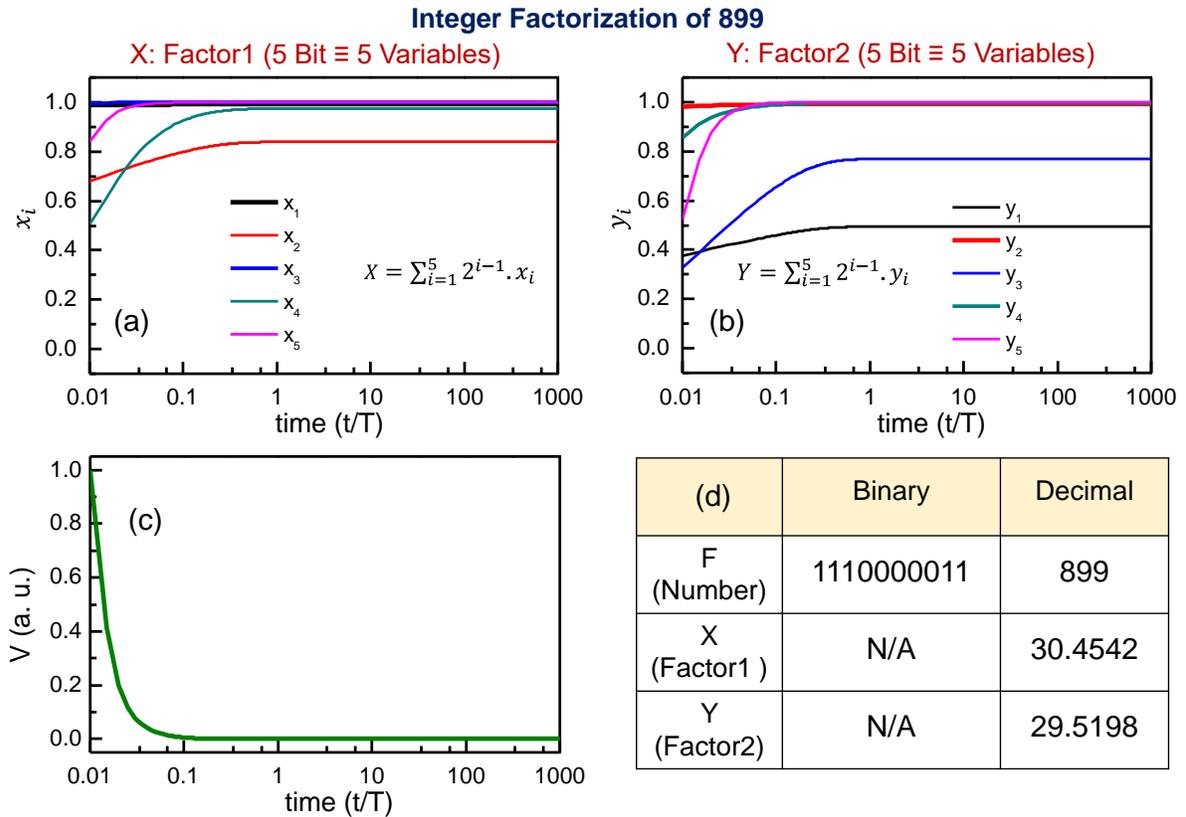


Figure 3.5. Integer factorization of 899 without using $\tanh(\cdot)$ in the energy formulation. Temporal evolution of: (a) (b) the variables representing bits in the integer factors X and Y, respectively; (c) Energy V. (d) Factors X and Y computed by the system. It can be observed that the system settles to non-integer values.

3.1.e. Graph isomorphism

This problem is defined as the challenge of evaluating if two graphs are equivalent. The non-trivial case entails evaluating if two graphs with equal number of vertices and edges have the same edge connectivity, i.e., adjacency matrices. Mathematically this problem can be expressed as: Given two graphs with adjacency matrices defined by A and B , is there a permutation matrix P such that $AP = PB$? [102] To formulate the computational model for this problem, we represent each element in P as $P_{ij} = \left(\frac{1+\tanh(k \cos(\alpha_{ij}))}{2}\right)$, and formulate the energy function as:

$$V = \sum_{m=1}^N \sum_{n=1}^N A(K_{mn}(\alpha))^2 \quad (3.21)$$

Here, $N \times N$ is the size of the matrices A and B . K_{mn} is defined as:

$$K_{mn} = \frac{1}{N} \left(\sum_{r=1}^N a_{mr} \left(\frac{1 + \tanh(k \cos(\alpha_{rn}))}{2} \right) - \sum_{s=1}^N \left(\frac{1 + \tanh(k \cos(\alpha_{ms}))}{2} \right) b_{sn} \right) \quad (3.22)$$

and represents the element-wise difference between the products of AP and PB i.e., $AP - PB$.

Derivation of K_{mn} used in the dynamics of graph isomorphism

Here, we now derive the expression for K_{mn} defined in Eq. (3.22).

$$K_{mn} = \frac{1}{N} ([A][P] - [P][B]) \quad (3.23)$$

The i^{th} elements in $[A]$, $[B]$ and $[P]$ are addressed as a_{ij} , b_{ij} and $p_{ij} = \left(\frac{1+\tanh(k \cos(\alpha_{ij}))}{2}\right)$, respectively. Thus, the mn^{th} element in $[X] = [A][P]$ and $[Y] = [P][B]$ can be calculated as:

$$x_{mn} = \sum_{r=1}^N a_{mr} \cdot p_{rn} = \sum_{r=1}^N a_{mr} \left(\frac{1 + \tanh(k \cos(\alpha_{rn}))}{2} \right) \quad (3.24)$$

and

$$y_{mn} = \sum_{s=1}^N p_{ms} \cdot b_{sn} = \sum_{s=1}^N \left(\frac{1 + \tanh(k \cos(\alpha_{ms}))}{2} \right) b_{sn} \quad (3.25)$$

Subsequently, the mn^{th} element of $[X] - [Y]$ can be computed as,

$$\begin{aligned} ([X] - [Y])_{mn} &= \sum_{r=1}^N a_{mr} \left(\frac{1 + \tanh(k \cos(\alpha_{rn}))}{2} \right) \\ &\quad - \sum_{s=1}^N \left(\frac{1 + \tanh(k \cos(\alpha_{ms}))}{2} \right) b_{sn} \end{aligned} \quad (3.26)$$

$([X] - [Y])_{mn}$ is normalized to N to calculate K_{mn} used in the energy function,

$$K_{mn} = \frac{1}{N} \left(\sum_{r=1}^N a_{mr} \left(\frac{1 + \tanh(k \cos(\alpha_{rn}))}{2} \right) - \sum_{s=1}^N \left(\frac{1 + \tanh(k \cos(\alpha_{ms}))}{2} \right) b_{sn} \right) \quad (3.27)$$

$K_{mn} = 0$ when the two terms are equal, and $V = 0$ when all the terms (elementwise) are matched. We note here that the energy function has a quadratic degree. Nevertheless, the problem is considered since the formulation is well aligned with the dynamical system proposed here. The corresponding dynamics of the system can be defined by:

$$\frac{d\alpha_{ij}}{dt} = (-\nabla_{\alpha} V(\alpha))_{ij} = - \sum_{m=1}^N \sum_{n=1}^N 2AK_{mn}(\alpha) \left(\frac{dK_{mn}(\alpha)}{d\alpha_{ij}} \right) \quad (3.28a)$$

where

$$\frac{dK_{mn}(\alpha)}{d\alpha_{ij}} = -\frac{1}{2N} \sin(\alpha_{ij}) \cdot k \operatorname{sech}^2(k \cos(\alpha_{ij})) \cdot [(a_{mi})_{n=j} - (b_{jn})_{m=i}] \quad (3.28b)$$

$$\frac{d\alpha_{ij}}{dt} = \sin(\alpha_{ij}) \left[\frac{A}{N} k \operatorname{sech}^2(k \cos(\alpha_{ij})) \cdot \left(\sum_{m=1}^N a_{mi} k_{mj} - \sum_{n=1}^N b_{jn} k_{in} \right) \right] \quad (3.28c)$$

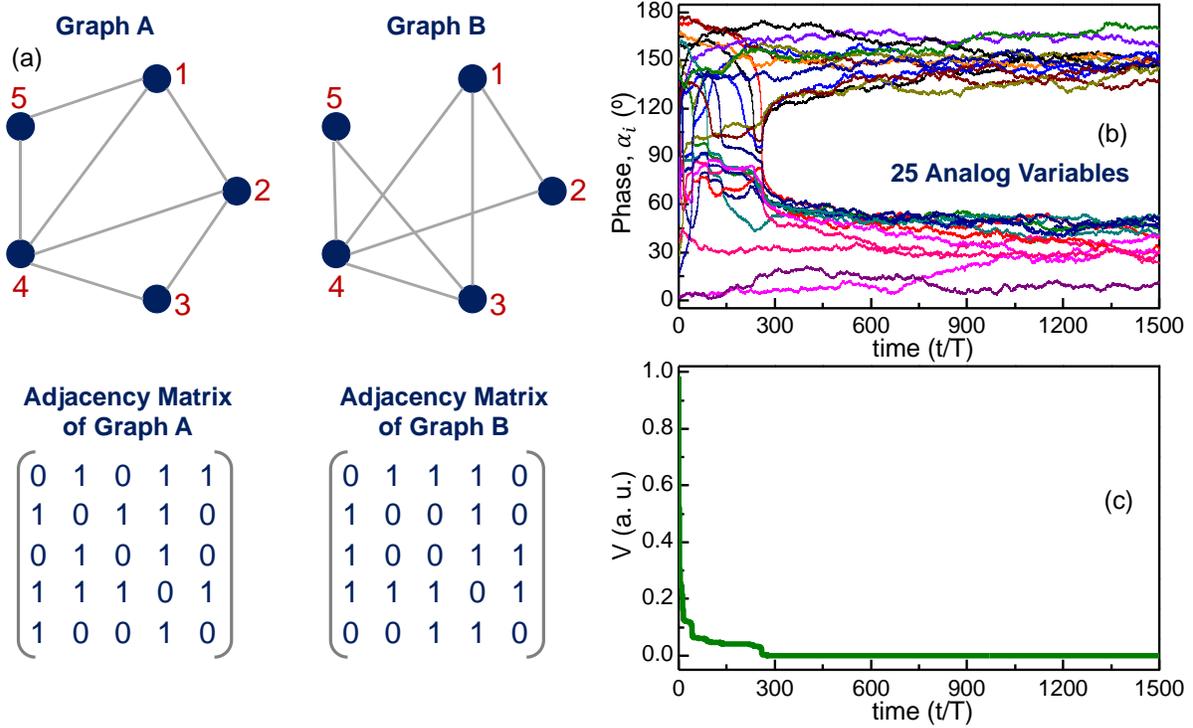


Figure 3.6. (a) Two representative graphs along with their respective adjacency matrices; (b)(c) Evolution of the phases and the total energy as a function of time, respectively. It can be observed that the energy (V) reduces to 0 indicating that the graphs are isomorphic.

Figure 3.6 shows an illustrative example (considering two graphs of 5 nodes) for evaluating the isomorphism between two graphs using the model proposed above.

3.1.f. Minimization of the Ising Hamiltonian and MaxCut

Next, we also illustrate how the above approach can be applied to minimizing the Ising Hamiltonian, and subsequently, show its application in solving the Maximum Cut problem- the minima of the Ising Hamiltonian $-\sum_{i,j;i<j}^N J_{ij}\sigma_i\sigma_j$ (Zeeman term neglected here) corresponds to the MaxCut of the equivalent graph when an edge between the nodes i and j is represented by $J_{ij} = -1$. Thus, both the problems also have objective functions with quadratic degree. We formulate the energy function for the above problem as:

$$V = A \sum_{i,j;i \neq j}^N J_{ij} (\cos(\alpha_i) - \cos(\alpha_j))^2 \quad (3.29)$$

where $J_{ij} = -1(0)$, if an edge is present (absent) between the nodes i and j , respectively.

The energy function in Eq. (3.29) can be expressed as:

$$V = A \sum_{i,j;i \neq j}^N J_{ij} (\cos(\alpha_i))^2 + A \sum_{i,j;i \neq j}^N J_{ij} (\cos(\alpha_j))^2 - 2A \sum_{i,j;i \neq j}^N J_{ij} \cos(\alpha_i) \cos(\alpha_j) \quad (3.30a)$$

Further,

$$\sum_{i,j;i \neq j}^N J_{ij} (\cos(\alpha_i))^2 = - \sum_{i=1}^N \Delta_i (\cos(\alpha_i))^2 \quad (3.30b)$$

where Δ_i is the degree of the i^{th} node in the graph. Therefore, Eq. (3.30a) can be expressed as:

$$V = -2A \sum_{i=1}^N \Delta_i (\cos(\alpha_i))^2 - 2A \sum_{i,j;i \neq j}^N J_{ij} \cos(\alpha_i) \cos(\alpha_j) \quad (3.30c)$$

Generalizing Eq. (3.30c), we have

$$V = - \sum_{i=1}^N C_i (\cos(\alpha_i))^2 - C \sum_{i,j;i \neq j}^N J_{ij} \cos(\alpha_i) \cos(\alpha_j) \quad (3.30d)$$

where C_i and C are positive constants. It can be observed from Eq. (3.30d) that V attains a minimum when $(\alpha_i, \alpha_j) = (0, \pi)$ or $(\pi, 0)$. At these specific phase points, Eq. (3.30d) can be simplified as:

$$V = - \sum_{i=1}^N C_i - C \sum_{i,j;i < j}^N J_{ij} \cos(\alpha_i) \cos(\alpha_j) \quad (3.30e)$$

The first term on the right-hand side is essentially a constant for a given graph. Further, by considering each oscillator $\cos(\alpha_i)$ as a spin σ_i , Eq. (3.30e) can be recast as:

$$V = -C \sum_{i,j;i < j}^N J_{ij} \sigma_i \sigma_j - C_s \quad (3.30f)$$

where, C and C_s are positive constants. Eq. (3.30f) is equivalent to the Ising Hamiltonian (the Zeeman term has been neglected here) with a constant offset.

Using Eq. (3.30d), the corresponding system dynamics can be defined as:

$$\frac{d\alpha_i}{dt} = (-\nabla_{\alpha} V(\alpha))_i = -2C_i \cos(\alpha_i) \sin(\alpha_i) - C \sum_{j=1; j \neq i}^N J_{ij} \sin(\alpha_i) \cos(\alpha_j) \quad (3.31a)$$

Exploiting the trigonometric relationships: $2 \cos(\alpha_i) \sin(\alpha_i) = \sin(2\alpha_i)$, and $2 \sin(\alpha_i) \cos(\alpha_j) = \sin(\alpha_i + \alpha_j) + \sin(\alpha_i - \alpha_j)$, Eq. (3.31a) can be expressed as:

$$\frac{d\alpha_i}{dt} = -C_i \sin(2\alpha_i) - Q \sum_{j=1; j \neq i}^N J_{ij} (\sin(\alpha_i + \alpha_j) + \sin(\alpha_i - \alpha_j)) \quad (3.31b)$$

where $Q = \frac{c}{2}$.

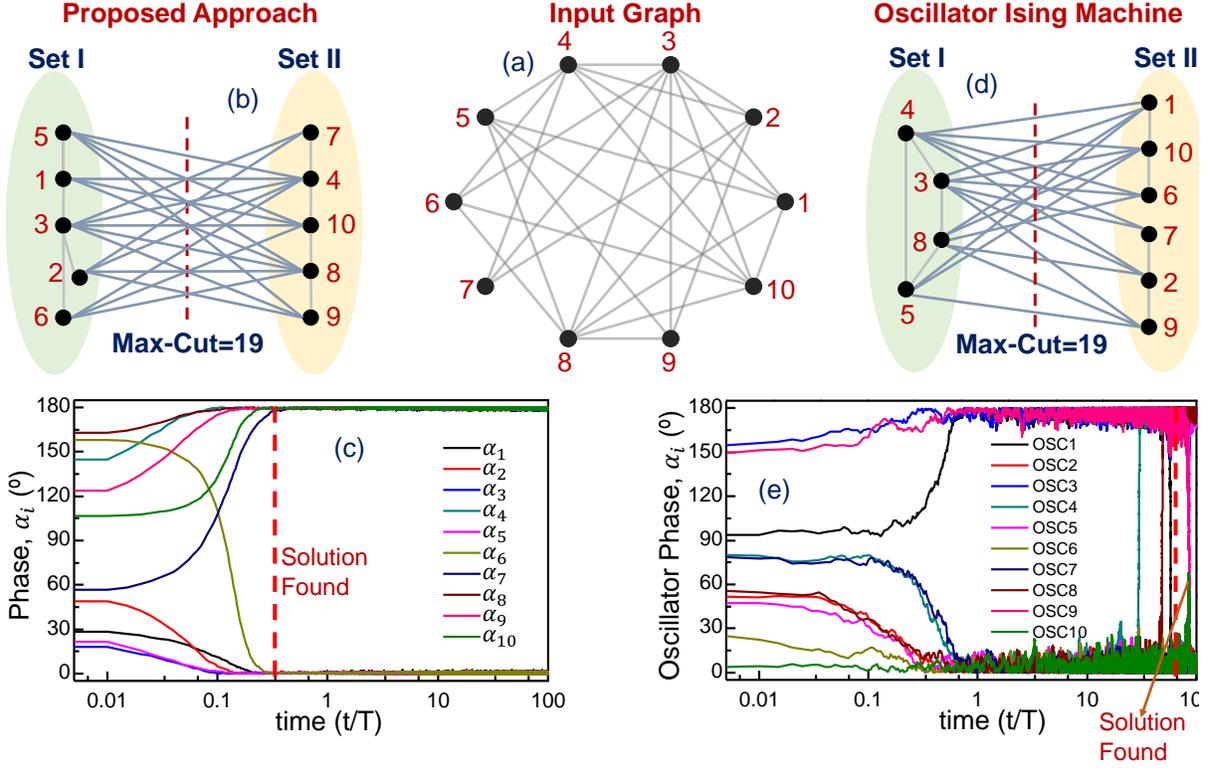


Figure 3.7. Computing MaxCut. (a) Illustrative graph considered. Phase evolution, and the resulting MaxCut solution computed using (b) (c) the proposed model, and (d) (e) the model developed in [85], respectively. Optimal solutions are achieved using both the models.

Eq. (3.31b) reveals the temporal dynamics of the system. In fact, as a computational model, Eq. (3.31) presents an alternative dynamical system to the oscillator-based dynamical system formulation proposed earlier [85]- the ground state energy is still equivalent to the global minima of Ising Hamiltonian for both the systems, but they will evolve with a different set of dynamics. Figure 3.7 shows the MaxCut computed on an illustrative 10-node graph using the proposed approach compared with the oscillator-based model developed earlier. Optimal solutions are observed in both cases.

3.2. Oscillator-based dynamical systems to solve COPs on hypergraph (SAT)

In this effort, we design and analyze an oscillator-inspired dynamical system and show that its dynamics can be directly used to compute solutions to the Max-NAE-3-SAT problem. As described in the previous section, the Not-all-Equal (NAE)-SAT problem is an NP-complete variant of the SAT problem which imposes the additional constraint that every clause must contain a literal that is true and another literal that is false; the Max-NAE-SAT problem is the optimization version of the problem where the objective is to maximize the number of clauses that meet this constraint. Ercsey-Ravasz. et al. [94] proposed an analog computational model for solving the SAT problem which was formulated using non-oscillating (analog) variables; further, our previous work also proposed computational models for many combinatorial problems (e.g., NAE-SAT, integer factorization among others) with non-oscillating analog variable [86]. While we draw many important insights from these works, our effort here is fundamentally different in that our dynamical systems use oscillating (analog) variables, and consequently, exhibit a different set of dynamics.

For this implementation, we formulate the system dynamics as: $(-\nabla_{\phi} E)_i = \frac{d\phi_i}{dt}$, where E is the potential energy function of the system. In contrast to the prior approach, here, we will first define the system dynamics, and subsequently, aim to show that there exists a Lyapunov (energy) function which can directly be mapped to the solution to the Max-NAE-3-SAT problem. We consider a system whose dynamics are defined by:

$$\begin{aligned}
\frac{d\phi_i}{dt} &= \sin(t + \phi_i) \left(- \sum_{m=1}^M \left(2AK_{m,osc}(t, \phi) \frac{c_{mi}K_{m,osc}(t, \phi)}{1 - c_{mi} \cos(t + \alpha_i)} \right) \right) \\
&\quad - \sin(2t + 2\phi_i) A_s \cos(2t) \\
&\equiv \chi(t + \phi_i(t)) B_i(t) + \chi(2t + 2\phi_i(t)) B^{(2)}(t)
\end{aligned} \tag{3.32}$$

Eq. (3.32) can be interpreted as a (sinusoidal) oscillator under perturbation ($B_i(t)$), and second harmonic signal injection $B^{(2)}(t) \equiv A_s \cos(2t)$ which helps binarize the phases to $(0, \pi)$ [68],[103], as illustrated further on. $\chi(t + \phi_i)$ and $\chi(2t + 2\phi_i)$ are the first and the second harmonics of the perturbation projection vectors (PPVs) of the oscillator, respectively. A and A_s are positive constants. It can be observed that the dynamics described in Eq. (3.32) are a modified version of the dynamics derived in Eq. (3.13b). However, it must be emphasized here that we do not use the potential energy function V since it does not decrease monotonically. Instead, using the dynamics described above, we will formulate a new energy function E whose ground state maps to the solution to the Max-NAE-3-SAT problem.

To define E , we first reformulate Eq. (3.32) in terms of the relative phase difference. Substituting the definition of $K_{m,osc}(t, \phi)$, Eq. (3.32) can be rewritten as,

$$\begin{aligned}
\frac{d\phi_i}{dt} &= -A \sin(t + \phi_i) \sum_{m=1}^M \left(c_{mi} \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \left(\frac{1 - c_{mi} \cos(t + \phi_i)}{2} \right) \right) \\
&\quad - \sin(2t + 2\phi_i) A_s \cos(2t)
\end{aligned} \tag{3.33}$$

Expanding Eq. (3.33), we have

$$\begin{aligned}
\frac{d\phi_i}{dt} = & -\frac{A}{2} \left(\sum_{m=1}^M \left(c_{mi} \sin(t + \phi_i) \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \right) \right. \\
& - \sum_{m=1}^M \left(\frac{1}{2} c_{mi}^2 \sin(2(t + \phi_i)) \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \right) \Big) \quad (3.34) \\
& - \sin(2t + 2\phi_i) A_s \cos(2t)
\end{aligned}$$

Further, using trigonometric identities to express all the product terms in

$\left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2$ as the sum of $\cos(\cdot)$ terms, we rewrite the expression as,

$$\sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \cos \left(\left(\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \right) t + \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j \right)$$

Using the approach described by Wang et al. [85], a differential equation such as Eq. (3.34) can be formulated as a Multi-time Partial Differential Equation (MPDE), wherein the fundamental oscillation is assumed to happen in fast time t_1 while the phases evolve in slow time t_2 . Subsequently, Eq. (3.34) can then be approximated as,

$$\begin{aligned}
\frac{d\phi_i}{dt} = & -A \sum_{m=1}^M \left(\sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi} Q_1 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i}^1 \sin \left(\phi_i \right. \right. \\
& \left. \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \Big|_{Q_1} \right) \right)
\end{aligned}$$

$$\begin{aligned}
& + A \sum_{m=1}^M \left(\sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi}^2 Q_2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^2 \sin \left(2\phi_i \right. \right. \\
& \quad \left. \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right|_{Q_2} \right) - A_{s1} \sin(2\phi_i)
\end{aligned} \tag{3.35}$$

Here, $Q_1 = 1$ when $\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 1$ else $Q_1 = 0$; $Q_2 = 1$ when $\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 2$ else $Q_2 = 0$. Additional details regarding the derivation of Eq. (3.35) can be found in Appendix I. Remarkably, there is a Lyapunov function $E(\phi(t))$ which can be defined for these dynamics as,

$$\begin{aligned}
E(\phi(t)) &= \sum_{i=1}^N \left[-A \sum_{m=1}^M \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi} Q_1 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^1 \cos \left(\phi_i(t) \right. \right. \\
& \quad \left. \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right|_{\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 1} \right) \Big] \\
& + \sum_{i=1}^N \left[\frac{A}{2} \sum_{m=1}^M \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi}^2 Q_2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^2 \cos \left(2\phi_i(t) \right. \right. \\
& \quad \left. \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right|_{\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 2} \right) \Big] - \frac{A_{s1}}{2} \cos(2\phi_i(t))
\end{aligned} \tag{3.36}$$

Unlike V (defined for the SAT solver (in 3.1)), $E(\phi(t))$ is defined in terms of relative phase difference (and not in terms of the absolute phase). To show that $E(\phi(t))$ is a decreasing function in time i.e., $\frac{dE(\phi(t))}{dt} \leq 0$, we express $\frac{dE(\phi(t))}{dt} = \frac{dE(\phi(t))}{d\phi_i(t)} \frac{d\phi_i(t)}{dt}$, where $\frac{dE(\phi(t))}{d\phi_i(t)}$ can be calculated as,

$$\begin{aligned}
\frac{\partial E(\phi(t))}{\partial \phi_i(t)} = & A \sum_{m=1}^M \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi} Q_1 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^1 \sin \left(\phi_i - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right) \Bigg|_{Q_1} \\
& - \frac{2A}{2} \sum_{m=1}^M \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi}^2 Q_2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^2 \sin \left(2\phi_i \right. \\
& \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right) \Bigg|_{Q_2} + A_{s1} \sin(2\phi_i) \equiv -\frac{d\phi_i(t)}{dt}
\end{aligned} \tag{3.37}$$

Thus,

$$\frac{\partial E(\phi(t))}{\partial \phi_i(t)} = -\frac{d\phi_i(t)}{dt} \tag{3.38}$$

It can be observed that Eq. (3.38) represents the system dynamics described earlier.

Subsequently,

$$\frac{dE(\phi(t))}{dt} = \sum_{i=1}^N \left[\left(\frac{\partial E(\phi(t))}{\partial \phi_i(t)} \right) \left(\frac{d\phi_i(t)}{dt} \right) \right] \tag{3.39}$$

$$= -\sum_{i=1}^N \left[\left(\frac{d\phi_i(t)}{dt} \right)^2 \right] \leq 0 \tag{3.40}$$

Eq. (3.40) reveals that $E(\phi(t))$ is decreasing in time.

While Eq. (3.40) represents a general form, we will specifically define the energy E for the case when each clause contains exactly 3 literals, and subsequently, show that its ground state can be used to find the solution of the NAE-3-SAT problem. When a clause contains 3 literals (corresponding to variables i, j, k), E can be expressed as,

$$\begin{aligned}
E(\phi) = \sum_{i=1}^N \left(\pi A \cdot 2^{-2N+1} \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(2c_{mi}c_{mj} \left(1 + \frac{1}{2}c_{mk}^2 \right) \cos(\phi_i - \phi_j) \right. \right. \\
+ 2c_{mi}c_{mk} \left(1 + \frac{1}{2}c_{mj}^2 \right) \cos(\phi_i - \phi_k) \\
+ \frac{1}{2}c_{mi}c_{mj}c_{mk}^2 \cos(\phi_i + \phi_j - 2\phi_k) \\
+ \frac{1}{2}c_{mi}c_{mk}c_{mj}^2 \cos(\phi_i + \phi_k - 2\phi_j) \\
+ \frac{1}{8}c_{mi}^2c_{mk}^2 \left(1 + \frac{1}{2}c_{mj}^2 \right) \cos(2\phi_i - 2\phi_k) \\
+ \frac{1}{2}c_{mi}^2c_{mj}c_{mk} \cos(2\phi_i - \phi_j - \phi_k) \\
\left. \left. + \frac{1}{8}c_{mi}^2c_{mj}^2 \left(1 + \frac{1}{2}c_{mk}^2 \right) \cos(2\phi_i - 2\phi_j) \right) \right) - \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i)
\end{aligned} \tag{3.41}$$

The details of this derivation are shown in Appendix II. The output variables are defined by the oscillator phases ϕ which settle to $\{0, \pi\}$ owing to the second harmonic injection. We note that if a clause contains literals corresponding to only one or two distinct variables, $i \neq j \neq k$ constraint will not be imposed for that specific clause in Eq. (3.41). The specific nature of the arguments of the $\cos(\cdot)$ functions shown in Eq. (3.41) arise from the characteristics of the cross-correlation operation performed in Eq. (3.41). The corresponding dynamics associated with Eq. (3.41) can be defined as,

$$\begin{aligned}
\frac{d\phi_i}{dt} = & \pi A 2^{-2N+1} \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(2c_{mi}c_{mj} \left(1 + \frac{1}{2}c_{mk}^2 \right) \sin(\phi_i - \phi_j) \right. \\
& + 2c_{mi}c_{mk} \left(1 + \frac{1}{2}c_{mj}^2 \right) \sin(\phi_i - \phi_k) \\
& + \frac{1}{2}c_{mi}c_{mj}c_{mk}^2 \sin(\phi_i + \phi_j - 2\phi_k) \\
& + \frac{1}{2}c_{mi}c_{mk}c_{mj}^2 \sin(\phi_i + \phi_k - 2\phi_j) \\
& + \frac{1}{4}c_{mi}^2c_{mk}^2 \left(1 + \frac{1}{2}c_{mj}^2 \right) \sin(2\phi_i - 2\phi_k) \\
& + \frac{1}{4}c_{mi}^2c_{mj}^2 \left(1 + \frac{1}{2}c_{mk}^2 \right) \sin(2\phi_i - 2\phi_j) \\
& \left. + c_{mi}^2c_{mj}c_{mk} \sin(2\phi_i - \phi_j - \phi_k) \right) - \pi A_s \sin(2\phi_i)
\end{aligned} \tag{3.42}$$

Eq. (3.42) describes the phase dynamics of the system which computes the solution to the 3-NAE-SAT problem. The second harmonic injection signal $-\sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i)$ (for an appropriate injection strength A_s) essentially lowers the energy of the system corresponding to $\phi \in \{0, \pi\}$, since the minimization of $-\sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i)$ to $-\sum_{i=1}^N \frac{\pi A_s}{2}$ forces the oscillators to take these binary phase values; this concept was also exploited in designing oscillator-based Ising machines [85]. Thus, when the system achieves ground state, each 2ϕ term in Eq. (3.41) induces a phase difference of 0 or 2π , and hence, the arguments of the corresponding $\cos(\phi_i + \phi_j - 2\phi_k)$ terms can be simplified to $\cos(\phi_i + \phi_j)$. Further, $\cos(2\phi_i - 2\phi_j)$ will take constant values at these specific phase points (represented as C). Additionally, $c_{mi}^2 = c_{mj}^2 = c_{mk}^2 = 1$. Thus, at these discrete

phase points, $E(\phi)$ for a problem in which each clause consists of three literals can be reduced to,

$$\begin{aligned}
E(\phi) = \pi A 2^{-2N+1} & \sum_{i=1}^N \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(3c_{mi}c_{mj} \cos(\phi_i - \phi_j) \right. \\
& + 3c_{mi}c_{mk} \cos(\phi_i - \phi_k) + \frac{1}{2}c_{mi}c_{mj} \cos(\phi_i + \phi_j) \\
& \left. + \frac{1}{2}c_{mi}c_{mk} \cos(\phi_i + \phi_k) + \frac{1}{2}c_{mj}c_{mk} \cos(\phi_j + \phi_k) \right) + C \\
& - \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i)
\end{aligned} \tag{3.43}$$

Rearranging Eq. (3.43),

$$\begin{aligned}
E(\phi) = \pi A 2^{-2N+1} & \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \sum_{i=1}^N \left(3c_{mi}c_{mj} \cos(\phi_i - \phi_j) \right. \\
& + 3c_{mi}c_{mk} \cos(\phi_i - \phi_k) + \frac{1}{2}c_{mi}c_{mj} \cos(\phi_i + \phi_j) \\
& \left. + \frac{1}{2}c_{mi}c_{mk} \cos(\phi_i + \phi_k) + \frac{1}{2}c_{mj}c_{mk} \cos(\phi_j + \phi_k) \right) + C \\
& - \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i) \\
= \sum_{m=1}^M \beta_m(\phi_i, \phi_j, \phi_k) + C & - \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i) = \sum_{m=1}^M \beta_m(\phi_i, \phi_j, \phi_k) + C - C_s
\end{aligned} \tag{3.44}$$

Both C and $C_s \left(= \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i) \right)$ are constants at the phase points, $\phi \in \{0, \pi\}$.

NAE-SAT Clause $x = (x_i, x_j, x_k)$	$E(\phi_i, \phi, \phi_k)$ for a single clause ($\propto [T_{ij} + T_{jk} + T_{ki}]$)			
	$\phi = (0,0,0)$ $\equiv x = (1,1,1)$	$\phi = (0,0,\pi)$ $\equiv x = (1,1,0)$	$\phi = (0,\pi,\pi)$ $\equiv x = (1,0,0)$	$\phi = (\pi,\pi,\pi)$ $\equiv x = (0,0,0)$
C_{NAE} $= (x_i \vee x_j \vee x_k)$ $\cdot (\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$
C_{NAE} $= (x_i \vee x_j \vee \bar{x}_k)$ $\cdot (\bar{x}_i \vee \bar{x}_j \vee x_k)$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$
C_{NAE} $= (x_i \vee \bar{x}_j \vee \bar{x}_k)$ $\cdot (\bar{x}_i \vee x_j \vee x_k)$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$
C_{NAE} $= (\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$ $\cdot (x_i \vee x_j \vee x_k)$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{-51}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 1$	$\left(\frac{189}{256}\right)\pi A - \frac{3}{2}\pi A_s$ $C_{NAE} = 0$

Figure 3.8. $E(\phi_i, \phi_j, \phi_k)$ for a single NAE-3-SAT clause computed for different combination of the literals. It can be observed that the energy is minimum only when the NAE-SAT clause is satisfied. Only selected combinations have been shown here; a detailed table considering all combinations has been shown in Appendix III.

Consequently, $E(\phi)$ is minimized when $\sum_{m=1}^M \beta_m(\phi_i, \phi_j, \phi_k)$ is minimum. Now, for a single clause consisting of 3 literals corresponding to 3 variables x_i, x_j, x_k (here, x_i, x_j, x_k can appear in normal or negated form in the clause), $\beta_m(\phi_i, \phi_j, \phi_k)$ can be written as,

$$\begin{aligned}
\beta_m(\phi_i, \phi_j, \phi_k) &= \pi A 2^{-2N+1} \left(c_{mi}c_{mj} \left(6 \cos(\phi_i - \phi_j) + \frac{3}{2} \cos(\phi_i + \phi_j) \right) \right. \\
&\quad + c_{mj}c_{mk} \left(6 \cos(\phi_j - \phi_k) + \frac{3}{2} \cos(\phi_j + \phi_k) \right) \\
&\quad \left. + c_{mk}c_{mi} \left(6 \cos(\phi_k - \phi_i) + \frac{3}{2} \cos(\phi_k + \phi_i) \right) \right) + C - C_s
\end{aligned} \tag{3.45a}$$

$$\beta_m(\phi_i, \phi_j, \phi_k) = \pi A 2^{-2N+1} (T_{ij} + T_{jk} + T_{ki}) + C - C_s \tag{3.45b}$$

where

$$T_{ij} = c_{mi}c_{mj} \left(6 \cos(\phi_i - \phi_j) + \frac{3}{2} \cos(\phi_i + \phi_j) \right) \tag{3.46}$$

Eq. (3.45b) reveals that $\beta_m(\phi_i, \phi_j, \phi_k)$ is minimum when $T_{ij} + T_{jk} + T_{ki}$ is minimum.

At the phase points $\phi_i, \phi_j, \phi_k \in \{0, \pi\}$, T_{ij} , T_{jk} , T_{ki} and $T_{ij} + T_{jk} + T_{ki}$ are binary in nature and exhibit the property that $T_{ij} + T_{jk} + T_{ki}$, and thus $\beta_m(\phi_i, \phi_j, \phi_k)$ is minimized when $(x_i \oplus x_j) \vee (x_j \oplus x_k) \vee (x_k \oplus x_i) = 1$. This is illustrated in the following paragraph.

However, first, we simplify $(x_i \oplus x_j) \vee (x_j \oplus x_k) \vee (x_k \oplus x_i)$ as,

$$\begin{aligned}
(x_i \oplus x_j) \vee (x_j \oplus x_k) \vee (x_k \oplus x_i) &= (x_i \bar{x}_j \vee \bar{x}_i x_j) \vee (x_j \bar{x}_k \vee \bar{x}_j x_k) \vee (x_k \bar{x}_i \vee \bar{x}_k x_i) \\
&= (x_i \vee x_j \vee x_k) \cdot (\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)
\end{aligned} \tag{3.47}$$

Remarkably, Eq. (3.47) corresponds to a clause of the NAE-3-SAT problem. The terms within the first parentheses in Eq. (3.47) implement the standard SAT constraint while the terms in the second parentheses implement the constraint that at least one literal must be false. Here, we again emphasize that x_i can appear in both normal or negated form;

for example if the clause is $(x_i \vee \bar{x}_j \vee x_k)$ the corresponding NAE-SAT clause will be $(x_i \vee \bar{x}_j \vee x_k) \cdot (\bar{x}_i \vee x_j \vee \bar{x}_k)$.

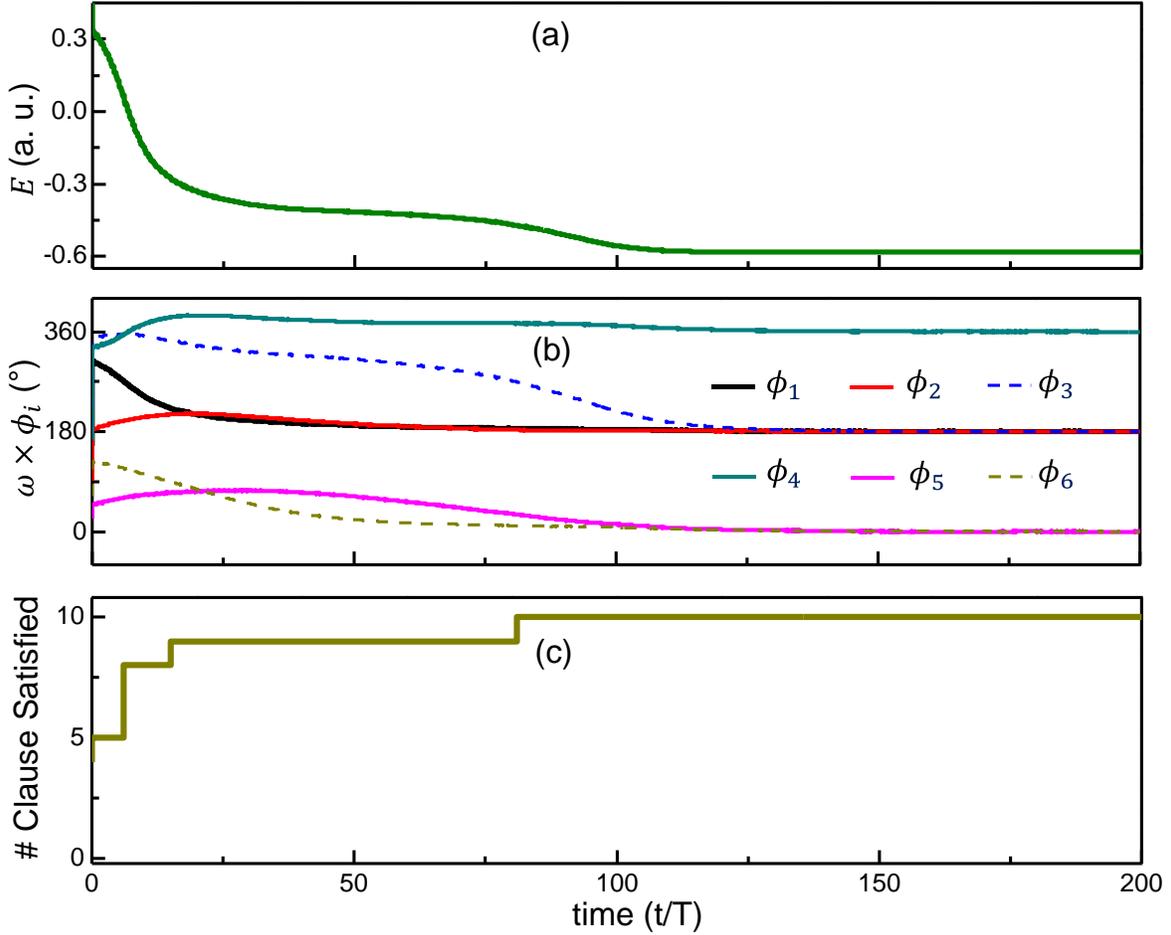


Figure 3.9. Evolution of (a) E , (b) $\omega \cdot \phi_i$, and (c) number of satisfied clauses with time, for an illustrative NAE-3-SAT problem with 6 variables and 10 clauses that is solved using the dynamics shown in Eq. (3.33). In this simulation, $\omega = 2\pi$ is used such that $T=1$.

To show that the energy corresponding to a clause, $T_{ij} + T_{jk} + T_{ki}$, is minimized when an NAE-3-SAT clause is satisfied, we consider the table in Fig. 3.8. It can be observed from the table that an NAE-SAT clause is satisfied only when $T_{ij} + T_{jk} + T_{ki}$ assumes the minimum value. Considering the inherent symmetry in the expression, only selected

cases have been presented here. However, the complete table has been shown in Appendix III.

Consequently, as the system evolves towards the global minimum of $E = \sum_{m=1}^M \beta_m(\phi_i, \phi_j, \phi_k) + C - C_s$, it aims to maximize the number of satisfied NAE-3-SAT clauses (defined by $(x_i \vee x_j \vee x_k).(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$). In other words, it computes the solution to the Max-NAE-3-SAT problem. Figure 3.9 shows the solution for an illustrative NAE-3-SAT problem having 6 variables and 10 clauses. The oscillator dynamics are simulated using Eq. (3.33). However, Eq. (3.42) can also be used to compute the solution, as shown in Appendix IV.

Simulation method

Here, the simulation approach used in the example problems considered in Fig. 3.2 and Fig. 3.9 is described. We use a stochastic differential equation (SDE) solver that allows noise to be considered; noise can help escape local minima in the phase space. The SDE that is used to evaluate the time evolution of oscillators' phases is given by,

$$d\phi_{it} = f(\phi_{it}, t) dt + a_n(\phi_{it}, t)dw_t \quad (3.48)$$

where $f(\phi_{it}, t)$ is the right-hand side of Eq. (3.13b) and the right-hand side of Eq. (3.33), respectively; $a_n(\phi_{it}, t)$ is the amplitude of the noise. A time and phase independent noise amplitude of 5×10^{-4} is used here. w_t is a Wiener process [84]. We use the Runge-Kutta method of order four to develop the differential equation solver [104] in MATLAB. Values of A and A_s used in the simulation are:

	A	A_s
Alternative Dynamical System to Solve SAT (Eq. (3.13b))	$\frac{10}{2\pi}$	N/A
NaAE-SAT Solver (Eq. (3.33))	$\frac{5}{2\pi}$	$\frac{0.01}{2\pi}$

The illustrative 3-SAT / NAE-3-SAT problem used in all the above examples is given by:

$$\begin{aligned}
Y = & (x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee x_6) \wedge (\overline{x_2} \vee \overline{x_5} \vee \overline{x_6}) \\
& \wedge (x_1 \vee \overline{x_2} \vee x_6) \wedge (x_1 \vee x_4 \vee \overline{x_6}) \wedge (x_2 \vee \overline{x_4} \vee x_6) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_4}) \\
& \wedge (\overline{x_4} \vee x_5 \vee \overline{x_6}) \wedge (\overline{x_3} \vee \overline{x_4} \vee \overline{x_5})
\end{aligned} \tag{3.49}$$

3.3. Oscillator Ising machine with higher order interaction among Ising spins

This effort aims to develop a method so that higher order interaction among Ising spin can be directly mapped to oscillator-based dynamical system [105]. Such systems can enable the direct mapping and solving of any COPs expressed by Ising spin (not only SAT). Dynamical system formulations that have been used to ‘solve’ the Ising model typically consider only pair-wise coupling; examples include, oscillator Ising machines, coherent Ising machines etc. From an application standpoint, while these characteristics capture quadratic interactions, the dynamical systems and their supporting computational models cannot be applied *directly* to solve problems that require higher order interaction among the spins [79],[106]. Therefore, the objective of this work is two-fold: (1) define dynamical systems that model higher order (>2) interactions among the Ising spins; and (2) map the resulting dynamics to relevant computational problems. We consider two

examples: computing the solutions for the NAE-K-SAT (Not-All-Equal SAT) problem and the Max-K-Cut of a hypergraph. Our motivation behind selecting these two combinatorial optimization problems was that their objective functions directly map to the solution of the higher order Ising models, and therefore, help illustrate the principle of how dynamical systems for the higher order Ising models can be used in combinatorial optimization.

The general form to represent higher order interactions among the Ising spins can be expressed as,

$$H = - \sum_i J_i^{(1)} s_i - \sum_{i,j} J_{ij}^{(2)} s_i s_j - \sum_{i,j,k} J_{ijk}^{(3)} s_i s_j s_k - \sum_{i,j,k,l} J_{ijkl}^{(4)} s_i s_j s_k s_l \dots \quad (3.50)$$

Where $J_{ij}^{(2)}$ represents the pairwise interaction coefficient between two Ising spins. The first term on the right-hand side ($-\sum_{i,j} J_{ij}^{(2)} s_i s_j$) is usually considered when describing quadratic/pairwise interactions among Ising spins $s = \{-1,1\}^n$; the Zeeman term which considers the interaction of spins with an external magnetic field has been neglected here. Considering the higher order interactions among the spins can help describe the objective functions of several combinatorial optimization problems (COPs) as illustrated here with the example of the NAE-K-SAT problem (without the need for problem decomposition). The NAE-K-SAT problem is a constrained version of the Boolean Satisfiability (SAT) problem where the objective is to find an assignment for the variables of the given Boolean expression (in the conjunctive normal form) such that: (a) at least one variable in every clause is TRUE (i.e., the clause is satisfied; standard SAT constraint); (b) at least one variable in every clause is FALSE [107]; the NAE-K-SAT problem is considered here since it directly maps to the general form of Eq. (3.50), as illustrated further on. Using an approach inspired by SAT, the NAE-K-SAT problem can be expressed as computing an

assignment for the variables such that $Y (= C_1.S_1 \wedge C_2.S_2 \wedge \dots \wedge C_M.S_M) = 1$. Here, $C_i \equiv (x_1 \vee x_2 \vee \bar{x}_3 \dots \bar{x}_N)$, and $S_i \equiv (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \dots x_N)$ (i.e., S_i and C_i have the same variables but in opposite forms).

3.3.a. NAE-K-SAT problem

To illustrate how we can map the NAE-K-SAT problem to higher order interactions among the Ising spins, we first consider the example of the NAE-4-SAT problem where each clause of the NAE-4-SAT problem consists of 4 literals, expressed in the general form as $(x_i \vee x_j \vee x_k \vee x_l).(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k \vee \bar{x}_l) \equiv (x_i \oplus x_j) \vee (x_i \oplus x_k) \vee (x_i \oplus x_l) \vee (x_j \oplus x_k) \vee (x_j \oplus x_l) \vee (x_k \oplus x_l)$, where $x \in \{0,1\}^n$ (x is a set of Boolean variables). $K=4$ is specifically chosen here since it is the lowest K where higher order interactions among the Ising spins are required to formulate the objective function for the problem (shown in Table 3.2). To formulate the problem in terms of Ising spins, we utilize the following property among the Boolean variables and the spins $(x_i \oplus x_j) \equiv \frac{1-s_i s_j}{2}$. Here, the logic level 0 (1) corresponds to an evaluation of -1(1) of the expression on the right-hand side, respectively. Furthermore, the complement of the logical OR among the XOR terms $((x_i \oplus x_j) \vee (x_i \oplus x_k) \vee \dots \vee (x_k \oplus x_l))$ can be expressed as, $\left(1 - \left(\frac{1-s_i s_j}{2}\right)\right) \cdot \left(1 - \left(\frac{1-s_i s_k}{2}\right)\right) \dots \left(1 - \left(\frac{1-s_k s_l}{2}\right)\right)$. Simplifying the above expression yields $\left(\frac{1+s_i s_j}{2}\right) \left(\frac{1+s_i s_k}{2}\right) \left(\frac{1+s_i s_l}{2}\right) \dots \left(\frac{1+s_k s_l}{2}\right) \equiv \frac{1}{8} (1 + s_i s_j + s_i s_k + s_i s_l + s_j s_k + s_j s_l + s_k s_l + s_i s_j s_k s_l)$. It can be observed that besides the second order interaction terms, the resulting expression also contains a 4th order interaction term among the spins. Consequently, the objective function for the NAE-4-SAT problem, over M clauses, can be formulated as the minimization of

$$H_{NAE-4-SAT} = - \sum_{m=1}^M \left(\sum_{\substack{i,j \\ i < j}}^N (-c_{mi}c_{mj}s_i s_j) + \sum_{\substack{i,j,k,l \\ i < j < k < l}}^N (-c_{mi}c_{mj}c_{mk}c_{ml}s_i s_j s_k s_l) \right) \quad (3.51)$$

Here, $c_{mi} = 1(-1)$, if the i^{th} variable appears in the m^{th} clause in the normal (negated) form; $c_{mi} = 0$ if the i^{th} variable is absent from the m^{th} clause. Using the same approach, we derive such expressions for a few other values of K in the NAE-K-SAT problem in Table 3.2. Details of the derivation of the objective function for NAE-5-SAT are shown in Appendix V.

K	Expression for a single clause & objective function for the NAE-K-SAT
2	Expression for a single clause: $(x_i \vee x_j).(\bar{x}_i \vee \bar{x}_j) \equiv s_i s_j$
	Objective function: $H = - \sum_{m=1}^M \sum_{i,j,i < j}^N (-c_{mi}c_{mj}s_i s_j) \equiv - \sum_{m=1}^M \sum_{i,j,i < j}^N J_{ij}s_i s_j$ <p>Where $J_{ij} = -c_{mi}c_{mj}$. It can be observed that when the variables appear only in the normal form i.e., $c_{mi} \geq 0$, the expression represents the solution to the archetypal MaxCut problem.</p>
3	Expression for a single clause: $(x_i \vee x_j \vee x_k).(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k) \equiv s_i s_j + s_i s_k + s_j s_k$
	Objective function: $H = - \sum_{m=1}^M \sum_{i,j,i < j}^N (-c_{mi}c_{mj}s_i s_j)$
4	Expression for a single clause: $(x_i \vee x_j \vee x_k \vee x_l).(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k \vee \bar{x}_l)$ $\equiv s_i s_j + s_i s_k + s_i s_l + s_j s_k + s_j s_l + s_k s_l + s_i s_j s_k s_l$
	Objective function:

	$H = - \sum_{m=1}^M \left(\sum_{\substack{i,j \\ i < j}}^N (-c_{mi}c_{mj}s_i s_j) + \sum_{\substack{i,j,k,l \\ i < j < k < l}}^N (-c_{mi}c_{mj}c_{mk}c_{ml}s_i s_j s_k s_l) \right)$
5	<p>Expression for a single clause:</p> $(x_i \vee x_j \vee x_k \vee x_l \vee x_m) \cdot (\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k \vee \bar{x}_l \vee \bar{x}_m)$ $\equiv s_i s_j + s_i s_k + s_i s_l + s_i s_m + s_j s_k + s_j s_l + s_j s_m + s_k s_l + s_k s_m + s_l s_m + s_i s_j s_k s_l$ $+ s_i s_j s_k s_m + s_i s_j s_l s_m + s_i s_k s_l s_m + s_j s_k s_l s_m$
	<p>Objective function:</p> $H = - \sum_{m=1}^M \left(\sum_{\substack{i,j \\ i < j}}^N (-c_{mi}c_{mj}s_i s_j) + \sum_{\substack{i,j,k,l \\ i < j < k < l}}^N (-c_{mi}c_{mj}c_{mk}c_{ml}s_i s_j s_k s_l) \right)$
<p><i>We note that constants and scalars have not been shown here in the expression for the single clause as well as for the objective function.</i></p>	

Table 3.2. Objective functions for the NAE-K-SAT problem expressed using Ising spins.

Constructing a dynamical system for the NAE-K-SAT problem. We now aim to formulate the dynamical system and the corresponding energy function for the NAE-K-SAT problem. The dynamical system, defined by $-(\nabla_{\phi} E)_i = \frac{d\phi_i}{dt}$, is designed such that the ground state of the ‘energy’ function (more precisely, the Lyapunov function) must correspond to a global optimum of the objective function. To construct this system, we draw inspiration from the dynamics of coupled oscillators under second harmonic injection which effectively forces the oscillator states to assume a binary phase value of 0 or π (details of the second harmonic injection can be found in work by Wang et al. [85]). Without loss of generality, we assume that one spin state (say, $s = +1$) is represented by phase 0 while the other spin state ($s = -1$) is represented by the phase angle π .

Subsequently, the second order interaction terms among the Ising spins $s_i s_j$ can be represented by $\cos(\theta_i - \phi_j)$. When the spins are in opposite states i.e., $s_i = 1(-1); s_j = -1(1)$, $s_i s_j \equiv \cos(\phi_i - \phi_j) = -1$, whereas when the spins are in the same states i.e., $s_i = 1(-1); s_j = 1(-1)$, $s_i s_j \equiv \cos(\phi_i - \phi_j) = 1$. Similarly, the higher order interactions can be modeled as shown in Table 3.3.

Order	Ising interaction term	Equivalent formulation for constructing dynamical system
1	s_i (Single Spin)	$\cos(\phi_i)$
2	$s_i s_j$	$\cos(\phi_i - \phi_j)$
3	$s_i s_j s_k$	$\cos(\phi_i - \phi_j + \phi_k)$
4	$s_i s_j s_k s_l$	$\cos(\phi_i - \phi_j + \phi_k - \phi_l)$
5	$s_i s_j s_k s_l s_m$	$\cos(\phi_i - \phi_j + \phi_k - \phi_l + \phi_m)$
6	$s_i s_j s_k s_l s_m s_n$	$\cos(\phi_i - \phi_j + \phi_k - \phi_l + \phi_m - \phi_n)$

Table 3.3. Equivalent energy function for modeling higher order interactions among Ising spins. The second harmonic signal included as a part of the dynamics (not shown here) helps force ϕ to $\{0, \pi\}$.

The equivalence between the higher order terms and the corresponding energy term is shown in Table 3.4.

Second Order Interactions ($s_i \cdot s_j$)			
$s_i s_j$	$s_i \cdot s_j$	$\phi_i \phi_j$	$\cos(\phi_i - \phi_j)$
-1 -1	+1	$\pi \pi$	+1
-1 +1	-1	$\pi 0$	-1
+1 -1	-1	0π	-1
+1 +1	+1	$0 0$	+1

Third Order Interactions ($s_i \cdot s_j \cdot s_k$)			
$s_i s_j s_k$	$s_i \cdot s_j \cdot s_k$	$\phi_i \phi_j \phi_k$	$\cos(\phi_i - \phi_j + \phi_k)$
-1 -1 -1	-1	$\pi \pi \pi$	-1
-1 -1 +1	+1	$\pi \pi 0$	+1
-1 +1 -1	+1	$\pi 0 \pi$	+1
-1 +1 +1	-1	$\pi 0 0$	-1
+1 -1 -1	+1	$0 \pi \pi$	+1
+1 -1 +1	-1	$0 \pi 0$	-1
+1 +1 -1	-1	$0 0 \pi$	-1
+1 +1 +1	+1	$0 0 0$	+1
Fourth Order Interactions ($s_i \cdot s_j \cdot s_k \cdot s_l$)			
$s_i s_j s_k s_l$	$s_i \cdot s_j \cdot s_k \cdot s_l$	$\phi_i \phi_j \phi_k \phi_l$	$\cos(\phi_i - \phi_j + \phi_k - \phi_l)$
-1 -1 -1 -1	+1	$\pi \pi \pi \pi$	+1
-1 -1 -1 +1	-1	$\pi \pi \pi 0$	-1
-1 -1 +1 -1	-1	$\pi \pi 0 \pi$	-1
-1 -1 +1 +1	+1	$\pi \pi 0 0$	+1
-1 +1 -1 -1	-1	$\pi 0 \pi \pi$	-1
-1 +1 -1 +1	+1	$\pi 0 \pi 0$	+1
-1 +1 +1 -1	+1	$\pi 0 0 \pi$	+1
-1 +1 +1 +1	-1	$\pi 0 0 0$	-1
+1 -1 -1 -1	-1	$0 \pi \pi \pi$	-1
+1 -1 -1 +1	+1	$0 \pi \pi 0$	+1
+1 -1 +1 -1	+1	$0 \pi 0 \pi$	+1
+1 -1 +1 +1	-1	$0 \pi 0 0$	-1
+1 +1 -1 -1	+1	$0 0 \pi \pi$	+1
+1 +1 -1 +1	-1	$0 0 \pi 0$	-1
+1 +1 +1 -1	-1	$0 0 0 \pi$	-1
+1 +1 +1 +1	+1	$0 0 0 0$	+1

Table 3.4. Equivalence between the higher order Ising spin interaction terms and the equivalent energy function.

Using the above relationships developed in Table 3.2, the energy functions for the NAE-K-SAT problem can be formulated as shown in Table 3.5. The corresponding dynamics $\left(\frac{d\phi_i}{dt}\right)$, shown in Table 3.5, can be obtained from the dynamical system equation $\frac{d\phi_i}{dt} = -(\nabla_{\phi} E)_i$. The second harmonic term in the energy function $\left(-\frac{C_s}{2} \sum_{i=1}^N \cos(2\phi_i)\right)$ is added to ensure that the oscillator phases effectively binarize to $\{0, \pi\}$. The energy contribution of this term is minimized ($= -N \frac{C_s}{2}$) at the binary phase points $\phi \in \{0, \pi\}$. Consequently, by using the appropriate strength of the second harmonic injection (C_s), we can ensure that the energy function reaches its minimum for $\phi \in \{0, \pi\}$. We have borrowed this approach from prior work on oscillator based Ising machines (with second order interactions) [85].

K	Objective function, equivalent energy function, and dynamics.
2 & 3	<p>Objective function:</p> $H = - \sum_{m=1}^M \sum_{i,j,i<j}^N (-c_{mi}c_{mj}S_iS_j)$ <p>Energy function:</p> $E = C \sum_{m=1}^M \left[\sum_{i,j,i<j}^N c_{mi}c_{mj} \cos(\phi_i - \phi_j) + 1 \right] - \frac{C_s}{2} \sum_{i=1}^N \cos(2\phi_i)$ <p>Dynamics:</p> $\frac{d\phi_i}{dt} = C \left[\sum_{m=1}^M \sum_{j=1}^N c_{mi}c_{mj} \sin(\phi_i - \phi_j) \right] - C_s \sin(2\phi_i)$
4 & 5	<p>Objective function:</p> $H = - \sum_{m=1}^M \left(\sum_{\substack{i,j \\ i<j}}^N (-c_{mi}c_{mj}S_iS_j) + \sum_{\substack{i,j,k,l \\ i<j<k<l}}^N (-c_{mi}c_{mj}c_{mk}c_{ml}S_iS_jS_kS_l) \right)$ <p>Energy function:</p>

$E = C \sum_{m=1}^M \left[\sum_{i,j,i<j}^N c_{mi}c_{mj} \cos(\phi_i - \phi_j) \right. \\ \left. + \sum_{\substack{i,j,k,l \\ i<j<k<l}}^N c_{mi}c_{mj}c_{mk}c_{ml} \cos(\phi_i - \phi_j + \phi_k - \phi_l) + 1 \right] \\ - \frac{C_s}{2} \sum_{i=1}^N \cos(2\phi_i)$
<p>Dynamics:</p> $\frac{d\phi_i}{dt} = C \sum_{m=1}^M \left[\sum_{j=1}^N c_{mi}c_{mj} \sin(\phi_i - \phi_j) \right. \\ \left. + \sum_{\substack{i \neq j \neq k \neq l \\ j < k < l}}^N c_{mi}c_{mj}c_{mk}c_{ml} \sin(\phi_i - \phi_j + \phi_k - \phi_l) \right] - C_s \sin(2\phi_i)$

Table 3.5. Objective functions, corresponding energy expressions, and system dynamics for NAE-K-SAT problems for $K=2,3,4$, and 5 . We note that while the form of the expressions for $K=2$ and $K=3$, as well as $K=4$ and $K=5$ are similar, the coefficients (c_{mi}) are different. C is the strength of coupling among the nodes whereas C_s represents the strength of the second harmonic injection.

Furthermore, using the dynamical system equation $\frac{d\phi_i}{dt} = -(\nabla_{\phi} E)_i$, we can also show that for the energy functions described in Table 3.5, $\frac{dE}{dt} \leq 0$ i.e., they are Lyapunov functions.

$$\frac{dE}{dt} = \sum_{i=1}^N \frac{\partial E}{\partial \phi_i} \cdot \frac{d\phi_i}{dt} = \sum_{i=1}^N \left(-\frac{d\phi_i}{dt} \right) \frac{d\phi_i}{dt} = - \sum_{i=1}^N \left(\frac{d\phi_i}{dt} \right)^2 \quad (3.52)$$

Figure 3.10 shows an illustrative example of the NAE-4-SAT problem computed using the proposed dynamical system.

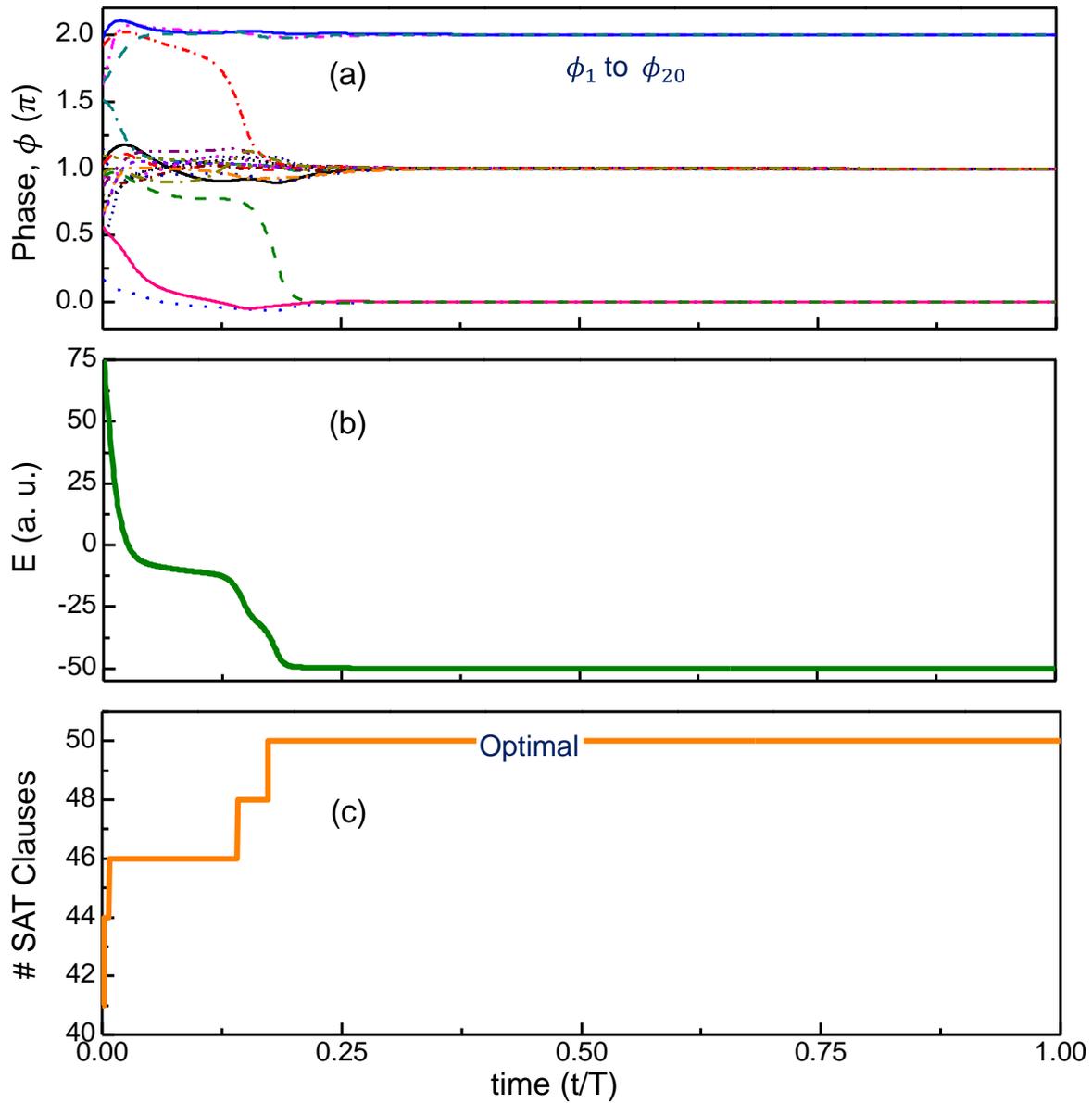


Figure 3.10. Evolution of (a) phases (ϕ); (b) energy; (c) number of satisfied NAE-4-SAT clauses for an illustrative NAE-4-SAT problem (20 variables and 50 clauses) computed using the proposed dynamical system (higher order oscillator Ising machine).

Simulation Method

Values of C and C_s used in the simulation of the NAE-4-SAT are:

Problem Solved	C	C_s
NAE-4-SAT	$\frac{10}{8}$	5

3.3.b. Max-K-Cut on a hypergraph

In the prior section, we exploited the binary nature of the Ising spins (along with higher order interactions among them). We now ‘extend’ the definition of the ‘spin’ in order to facilitate the design of computational models for an even broader spectrum of COPs that would benefit from the use of >2 states for each node/spin. To facilitate this, we express the possible states of a spin as $re^{i\theta_k}$, where $r = 1$, and $\theta_k = \frac{2\pi k}{K}$; $k = 1, 2, \dots, K - 1$. When $K = 2$, the possible states are within $\{1, -1\}$, which represents the traditional definition of an Ising spin. In contrast, when $K > 2$, the ‘spin’ assumes K configurations, represented as complex quantities (e.g., for $K = 3$, the possible states are $1, e^{i\frac{2\pi(1)}{3}}, e^{i\frac{2\pi(2)}{3}}$). While we have utilized this concept for solving combinatorial problems on *graphs* (i.e., problems with quadratic objective functions) [82], here we explore this concept for hypergraphs (that entail higher order interactions) by considering the example of solving the Max-K-Cut of a hypergraph.

Computing the Max-K-Cut on a hypergraph is defined as the challenge of partitioning the nodes of a hypergraph into K partitions in a manner that maximizes the number of hyperedges having nodes that lie in at least two sets created by the partitions [108]. The Max-K-Cut problem and its comparison with the archetypal MaxCut problem are

illustrated in Chapter 1, in Fig. 1.5a and Fig. 1.5b for the case of a graph and a hypergraph, respectively.

To develop the objective function for the problem, each hyperedge of the graph can be expressed as $h_m = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \left(1 - c_{mi} c_{mj} \left(\frac{1 - \text{Re}(s_i s_j^* e^{if_K(\Delta\theta_{ij})})}{2} \right) \right)$, where $s_j = 1e^{i\theta_j}$; θ_j can assume any of the following values from $\frac{2\pi k}{K}$; $k = 1, 2, \dots, K - 1$ enforced by the higher order harmonic injection. $c_{mj} = 1(0)$ if the j^{th} node belongs (does not belong) to the m^{th} hyperedge. We note that the 'i' represents the imaginary number $\sqrt{-1}$ whereas 'i' refers to the index.

$$f_K(\Delta\theta_{ij}) = \lim_{\sigma \rightarrow 0} \sum_{k=1}^{K-1} \left[\left((2k-1)\pi - \frac{2k\pi}{K} \right) \cdot e^{-\left(\frac{(\Delta\theta_{ij} - \frac{2k\pi}{K})^2}{2\sigma^2} \right)} + \left(\frac{2k\pi}{K} - (2k-1)\pi \right) \cdot e^{-\left(\frac{(\Delta\theta_{ij} + \frac{2k\pi}{K})^2}{2\sigma^2} \right)} \right] \quad (3.53)$$

$f_K(\Delta\theta_{ij})$ is designed such that $\text{Re}(s_i s_j^* e^{if_K(\Delta\theta_{ij})}) = -1(1)$, if the nodes i and j are placed in different (same) sets, and essentially reward (penalize) the system in terms of energy, respectively. Additional details about the design and properties of $f_K(\Delta\theta_{ij})$ have been presented in our prior work [82] as well as in Chapter 2 (oscillator Potts machine). Consequently, if the hyperedge satisfies the criterion for the Max-K-Cut i.e., that the

nodes that are connected by it belong to at least two sets, the corresponding h_m assumes a value of 0, else $h_m = 1$. Subsequently, the objective function for the problem, which entails maximizing the number of such hyperedges, can be expressed as minimizing H , where,

$$H = \sum_{m=1}^M h_m \equiv \sum_{m=1}^M \prod_{i=1}^{N-1} \prod_{j=i+1}^N \left(1 - c_{mi}c_{mj} \left(\frac{1 - \operatorname{Re}(s_i s_j^* e^{if_K(\Delta\theta_{ij})})}{2} \right) \right) \quad (3.54)$$

As an example, considering a hypergraph where the maximum number of nodes connected by a hyperedge is 3, the objective function for the Max-K-Cut problem can be expressed as:

$$H = \sum_{\substack{m=1, i \neq j \neq k \\ c_{mi}, c_{mj}, c_{mk} \neq 0}}^M \left(1 - c_{mi}c_{mj} \left(\frac{1 - \operatorname{Re}(s_i s_j^* e^{if_K(\Delta\theta_{ij})})}{2} \right) \right) \left(1 - c_{mi}c_{mk} \left(\frac{1 - \operatorname{Re}(s_i s_k^* e^{if_K(\Delta\theta_{ik})})}{2} \right) \right) \left(1 - c_{mj}c_{mk} \left(\frac{1 - \operatorname{Re}(s_j s_k^* e^{if_K(\Delta\theta_{jk})})}{2} \right) \right) \quad (3.55)$$

where,

$$f_K(\Delta\theta_{ij}) = \lim_{\sigma \rightarrow 0} \sum_{k=1}^2 \left[\left((2k-1)\pi - \frac{2k\pi}{3} \right) \cdot e^{-\left(\frac{(\Delta\theta_{ij} - \frac{2k\pi}{3})^2}{2\sigma^2} \right)} + \left(\frac{2k\pi}{3} - (2k-1)\pi \right) \cdot e^{-\left(\frac{(\Delta\theta_{ij} + \frac{2k\pi}{3})^2}{2\sigma^2} \right)} \right] \quad (3.56)$$

For a hypergraph with hyperedges having more than 3 nodes, the objective function entails the use of higher order interactions among the spins.

To formulate a dynamical system for minimizing the above objective function, we express $Re(s_i s_j^* e^{if_K(\Delta\theta_{ij})})$ as $\cos(\Delta\theta_{ij} + f_K(\Delta\theta_{ij}))$. Furthermore, we restrict the configuration space of θ to $\frac{2\pi k}{K}$ where $k = 1, 2, \dots, K - 1$, by injecting the K^{th} harmonic (of sufficient strength) which lowers the energy at specific phase points, as described in prior work [82]. The resulting energy function can be described as,

$$E = A \sum_{m=1}^M \prod_{i=1}^{N-1} \prod_{j=i+1}^N \left(1 - c_{mi} c_{mj} \left(\frac{1 - \cos(\Delta\phi_{ij} + f_K(\Delta\phi_{ij}))}{2} \right) \right) - \frac{A_s}{K} \sum_{i=1}^N \cos(K\phi_i) \quad (3.57)$$

We note that ϕ has been used to express the energy function for the dynamical system instead of θ which represents the configuration space of the ‘extended spin’. The corresponding dynamics for which the function in Eq. (3.57) is a Lyapunov function are given by:

$$\frac{d\phi_i}{dt} = -\frac{\partial E}{\partial \phi_i} \quad (3.58a)$$

$$\frac{d\phi_i}{dt} = \frac{A}{2} \sum_{m=1}^M \sum_{j=1, j \neq i}^N \left[c_{mi} c_{mj} \sin(\Delta\phi_{ij} + f_K(\Delta\phi_{ij})) \frac{h_m}{\left(1 - c_{mi} c_{mj} \left(\frac{1 - \cos(\Delta\phi_{ij} + f_K(\Delta\phi_{ij}))}{2} \right) \right)} \right] - A_s \sin(K\phi_i) \quad (3.58b)$$

In the derivation of Eq. (3.58b), we exploit the fact that $\frac{\partial f(\Delta\phi_{ij})}{\partial \phi_i} = 0$ [82]. Furthermore,

using Eq. (3.58a), it can be shown that $\frac{dE}{dt} = -\sum_{i=1}^N \left(\frac{d\phi_i}{dt} \right)^2 \leq 0$ (similar to Eq. (3.52)).

We now evaluate our proposed model on a representative hypergraph. We consider a hypergraph where each hyperedge has 3 vertices. The corresponding dynamics for this case can then be written as,

$$\begin{aligned}
\frac{d\phi_i}{dt} = \frac{A}{2} \sum_{m=1}^M \left[c_{mi}c_{mj} \sin(\Delta\phi_{ij} + f_K(\Delta\phi_{ij})) \left(1 - c_{mi}c_{mk} \left(\frac{1 - \cos(\Delta\phi_{ik} + f_K(\Delta\phi_{ik}))}{2} \right) \right) \right. \\
\left. - c_{mj}c_{mk} \left(\frac{1 - \cos(\Delta\phi_{jk} + f_K(\Delta\phi_{jk}))}{2} \right) \right] + c_{mi}c_{mk} \sin(\Delta\phi_{ik} \\
+ f_K(\Delta\phi_{ik})) \left(1 - c_{mi}c_{mj} \left(\frac{1 - \cos(\Delta\phi_{ij} + f_K(\Delta\phi_{ij}))}{2} \right) \right) \left(1 - c_{mj}c_{mk} \left(\frac{1 - \cos(\Delta\phi_{jk} + f_K(\Delta\phi_{jk}))}{2} \right) \right) \left. - A_s \sin(K\phi_i) \right] \quad (3.59)
\end{aligned}$$

Figure 3.11 also shows the computed Max-K-Cut (for K=2, 3, and 4) for a hypergraph instance (with 10 nodes, and 20 hyperedges). The illustrative problem has a maximum of 4 nodes per hyperedge.

Simulation Method

Here, we describe the simulation approach used to simulate the NAE-4-SAT problem (Fig. 3.10) and the hypergraph Max-K-Cut problem (Fig. 3.11). We solve the dynamics using a stochastic differential equation (SDE) solver implemented in MATLAB; details of its implementation have been described in our previous work [100]. The SDE solver incorporates noise that helps escape local minima in the phase space.

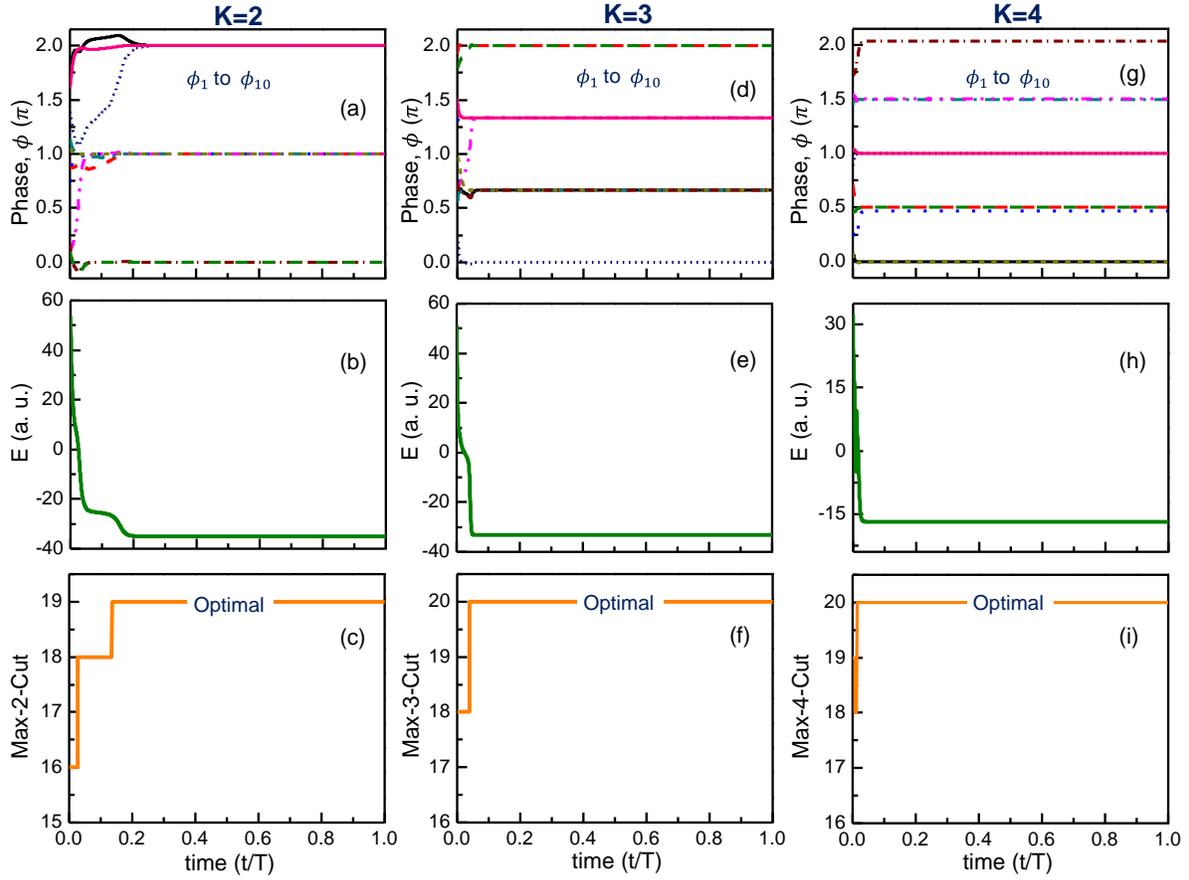


Figure 3.11. Max-K-Cut ($K=2,3,4$) solutions computed using the proposed dynamical system for an illustrative hypergraph. Evolution of phases (ϕ), energy and the Max-K-Cut solution, respectively for (a-c) $K=2$; (d-f) $K=3$; (g-i) $K=4$.

Values of A and A_s used in the simulation of the Max-K-Cut are:

Problem Solved	A	A_s
Hypergraph Max-2-Cut	15	10
Hypergraph Max-3-Cut	15	10
Hypergraph Max-4-Cut	10	10

3.4. Alternative formulations for higher order Ising spin interactions

A general form of the Ising Hamiltonian shown in Eq. (3.50) (also shown below) with the second order interactions terms ($s_i s_j$) being most commonly employed.

$$H = - \sum_i J_i^{(1)} s_i - \sum_{i,j} J_{ij}^{(2)} s_i s_j - \sum_{i,j,k} J_{ijk}^{(3)} s_i s_j s_k - \sum_{i,j} J_{ijk}^{(4)} s_i s_j s_k s_l - \dots$$

where, $s_i \in \{-1, 1\}$, $J_{ij\dots}^{(n)}$ represents the n^{th} order interaction coefficient. As described in the introduction, such higher order interactions can be converted to second order interactions by adding auxiliary / ancillary variables [79].

Here, we propose a non-linear transformation that allows many body spin interactions to be expressed in terms of quadratic (two-body) spin interactions without invoking auxiliary variables. The proposed method can express a higher order Ising spin interaction as a function of the nonlinear transform of the second order and first order interaction terms.

We express the n^{th} order spin interaction term as,

$$S_n = s_1 s_2 s_3 s_4 \dots s_n \quad (3.60)$$

where, $s_i \in \{-1, 1\}$. We have formulated two different alternative expressions for even and odd order Ising spin interaction. They are discussed in the following subsections.

3.4.a. Higher order interactions: even order

When n is an even number i.e., $n = 2p$, $p \in \mathbb{Z}^+$,

$$S_n = 2f_p^2(\cdot) - 1 \quad (3.61)$$

where,

$$f_p(.) = \lim_{k \rightarrow \infty} \tanh \left(k \sum_{\substack{T \subseteq \{1,2,\dots,n\} \\ |T|=p}} \prod_{s_i \in T} s_i \right) \quad (3.62)$$

For instance, in the case of fourth order interactions,

$$\begin{aligned} S_4 &= \lim_{k \rightarrow \infty} \tanh \left(k \sum_{i,j,i \neq j}^4 s_i s_j \right) \\ &= \lim_{k \rightarrow \infty} \tanh(k(s_1 s_2 + s_1 s_3 + s_1 s_4 + s_2 s_3 + s_2 s_4 + s_3 s_4)) \end{aligned} \quad (3.63)$$

For the benefit of the reader, Table 3.6 verifies Eq. (3.63) numerically although a formal proof for this will be presented in Appendix VI.

s_1, s_2, s_3, s_4	S_4 $= s_1 s_2 s_3 s_4$	$\sum_{i,j,i \neq j}^4 s_i s_j$	$f_2(.)$ $= \lim_{k \rightarrow \infty} \tanh \left(k \sum_{i,j,i \neq j}^4 s_i s_j \right)$	$2f_2^2(.) - 1$
-1, -1, -1, -1	1	6	1	1
-1, -1, -1, 1	-1	0	0	-1
-1, -1, 1, -1	-1	0	0	-1
-1, -1, 1, 1	1	-2	-1	1
-1, 1, -1, -1	-1	0	0	-1
-1, 1, -1, 1	1	-2	-1	1
-1, 1, 1, -1	1	-2	-1	1
-1, 1, 1, 1	-1	0	0	-1
1, -1, -1, -1	-1	0	0	-1
1, -1, -1, 1	1	-2	-1	1
1, -1, 1, -1	1	-2	-1	1
1, -1, 1, 1	-1	0	0	-1
1, 1, -1, -1	1	-2	-1	1
1, 1, -1, 1	-1	0	0	-1
1, 1, 1, -1	-1	0	0	-1
1, 1, 1, 1	1	6	1	1

Table 3.6. Original S_4 and S_4 obtained using the proposed method for all the possible spin configurations.

The complete alternative expression for an 8th order Ising spin interaction term is shown below:

$$S_8 = s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 = 2f_4^2 - 1 = 2.(2f_2^2 - 1)^2 - 1 = 8f_2^4 - 8f_2^2 + 1 \quad (3.64a)$$

$$S_8 = \lim_{k \rightarrow \infty} \left[8 \left(\tanh \left(k \left(\sum_{i,j=1, i \neq j}^8 s_i s_j \right) \right) \right)^4 - 8 \left(\tanh \left(k \left(\sum_{i,j=1, i \neq j}^8 s_i s_j \right) \right) \right)^2 + 1 \right] \quad (3.64b)$$

Eq. (3.64b) is the expression of an 8th order Ising spin interaction in terms of pairwise interaction.

3.4.b. Higher order interactions: odd order

When n is an odd number i.e., $n = 2q + 1$, $q \in \mathbb{Z}^+$

$$S_n = f_q(\cdot) f_{q+1}(\cdot) \quad (3.65)$$

where $f_q(\cdot)$ is defined in Eq. (3.62).

For instance, in the case of fifth order interactions, $S_5 = f_2(\cdot) f_3(\cdot)$, where

$f_2(\cdot) = \lim_{k \rightarrow \infty} \tanh(k \sum_{i,j, i \neq j}^5 s_i s_j)$; $f_3(\cdot) = \lim_{k \rightarrow \infty} \tanh(k \sum_{i,j,k, i \neq j \neq k}^5 s_i s_j s_k)$. This has been

verified in Table 3.7 below. A formal verification for this formulation will be provided in

Appendix VII.

s_1, s_2, s_3, s_4, s_5	S_5 $= s_1 s_2$ $\times s_3 s_4 s_5$	$\sum_{i,j, i \neq j}^5 s_i s_j$	$f_2(\cdot) =$ $\lim_{k \rightarrow \infty} \tanh \left(k \sum_{i,j, i \neq j}^5 s_i s_j \right)$	$\sum_{i,j,k, i \neq j \neq k}^5 s_i s_j s_k$	$f_3(\cdot) =$ $\lim_{k \rightarrow \infty} \tanh \left(k \sum_{i,j,k, i \neq j \neq k}^5 s_i s_j s_k \right)$	$f_2(\cdot)$ $\times f_3(\cdot)$
-1, -1, -1, -1, -1	-1	10	1	-10	-1	-1
-1, -1, -1, -1, 1	1	2	1	2	1	1

-1, -1, -1, 1, -1	1	2	1	2	1	1
-1, -1, -1, 1, 1	-1	-2	-1	2	1	-1
-1, -1, 1, -1, -1	1	2	1	2	1	1
-1, -1, 1, -1, 1	-1	-2	-1	2	1	-1
-1, -1, 1, 1, -1	-1	-2	-1	2	1	-1
-1, -1, 1, 1, 1	1	-2	-1	-2	-1	1
-1, 1, -1, -1, -1	1	2	1	2	1	1
-1, 1, -1, -1, 1	-1	-2	-1	2	1	-1
-1, 1, -1, 1, -1	-1	-2	-1	2	1	-1
-1, 1, -1, 1, 1	1	-2	-1	-2	-1	1
-1, 1, 1, -1, -1	-1	-2	-1	2	1	-1
-1, 1, 1, -1, 1	1	-2	-1	-2	-1	1
-1, 1, 1, 1, -1	1	-2	-1	-2	-1	1
-1, 1, 1, 1, 1	-1	2	1	-2	-1	-1
1, -1, -1, -1, -1	1	2	1	2	1	1
1, -1, -1, -1, 1	-1	-2	-1	2	1	-1
1, -1, -1, 1, -1	-1	-2	-1	2	1	-1
1, -1, -1, 1, 1	1	-2	-1	-2	-1	1
1, -1, 1, -1, -1	-1	-2	-1	2	1	-1
1, -1, 1, -1, 1	1	-2	-1	-2	-1	1
1, -1, 1, 1, -1	1	-2	-1	-2	-1	1
1, -1, 1, 1, 1	-1	2	1	-2	-1	-1
1, 1, -1, -1, -1	-1	-2	-1	2	1	-1
1, 1, -1, -1, 1	1	-2	-1	-2	-1	1
1, 1, -1, 1, -1	1	-2	-1	-2	-1	1
1, 1, -1, 1, 1	-1	2	1	-2	-1	-1
1, 1, 1, -1, -1	1	-2	-1	-2	-1	1
1, 1, 1, -1, 1	-1	2	1	-2	-1	-1
1, 1, 1, 1, -1	-1	2	1	-2	-1	-1
1, 1, 1, 1, 1	1	10	1	10	1	1

Table 3.7. Original S_5 and S_5 obtained using the proposed method for all the possible spin configurations.

The complete alternative expression for a 7th order Ising spin interaction term is shown below:

$$S_8 = s_1 s_2 s_3 s_4 s_5 s_6 s_7 = f_4 f_3 = f_1 f_2 (2f_2^2 - 1) = 2f_1 f_2^3 - f_1 f_2 \quad (3.66a)$$

$$S_7 = \lim_{k \rightarrow \infty} \left[2 \left(\tanh \left(k \left(\sum_{i=1}^7 s_i \right) \right) \right) \left(\tanh \left(k \left(\sum_{i,j=1, i \neq j}^7 s_i s_j \right) \right) \right)^3 \right. \\ \left. - \left(\tanh \left(k \left(\sum_{i=1}^7 s_i \right) \right) \right) \left(\tanh \left(k \left(\sum_{i,j=1, i \neq j}^7 s_i s_j \right) \right) \right) \right] \quad (3.66b)$$

Eq. (3.66b) is the expression of a 7th order Ising spin interaction in terms of pairwise interaction.

Chapter 4

4. Dynamical Properties of Coupled Oscillator-based Computing Systems

This chapter discusses the incorporation of control theoretic approaches to analyze the dynamical properties of oscillator-based computing systems, that govern their computational performance.

Coupled oscillator-based computing systems can be thought of as gradient descent systems. While such a system evolves through the high-dimensional solution space towards the ground state energy (corresponding to the optimal solution), it is likely to encounter many local minima where the system can get trapped, and subsequently, give rise to a sub-optimal solution. This is illustrated with the example of an oscillator network with 20 nodes and 114 edges (corresponding to the interactions among the oscillators) as shown in Fig. 4.1(a). Figure 4.1(b) shows the experimentally measured Ising Hamiltonian solutions (H) over 100 separate trials. Experiments are done on a chip of 30 coupled oscillator based Ising machine [69]. Figure 4.2(c) compares the measured H attained by the system (and its frequency) with the entire combinatorial solution space i.e., H corresponding to all the possible spin assignments (grey in Fig. 4.1(c)) for the problem; there are 524,288 possible spin assignments out of which only 7 correspond to the optimal solution (here, self-biasing of spins is not considered). It is evident from Fig. 4.1(c) that the spin configurations measured using the system are sub-optimal with the best solution only equal to 92% of the optimal value. Moreover, it can also be observed

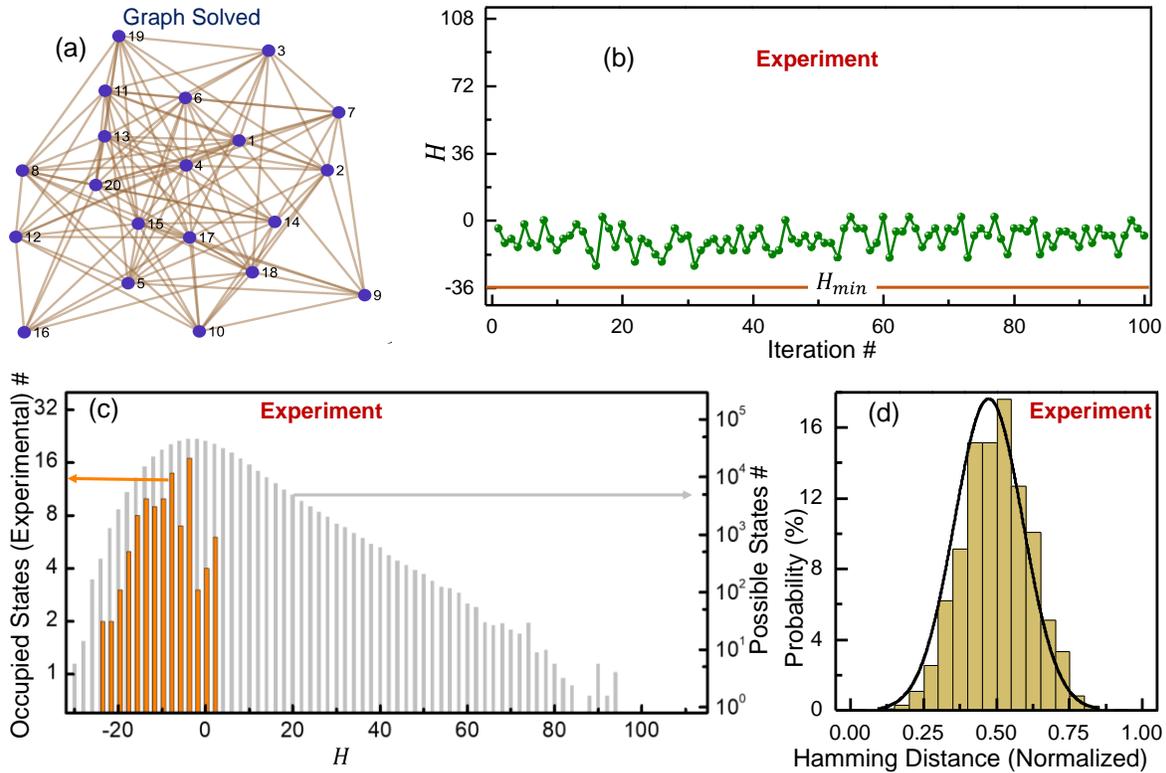


Figure 4.1. (a) A representative network of 20 spins with randomly generated interactions (represented by edges); (b) Experimentally measured H over 100 separate trials (H_{min} =minimum/optimal H); (c) Distribution showing occupied energy states (represented by H) and their frequency (orange) compared to the complete solution space (grey) of the problem (=524,288 possible states); (d) Hamming distance distribution (normalized) between the experimentally measured solutions over 100 runs.

that the system energy (proportional to H) at which the peak of the distribution of the measured solutions (over the 100 trials) occurs, coincides with the H value where the maximum number of local minima states occur. This indicates that in the absence of any external annealing, the system - despite trying to minimize its energy- gets trapped in one of the many local minima of the phase space, consequently, giving rise to sub-optimal spin assignments. Furthermore, we also compute the Hamming distance among the measured spin assignments (Fig. 4.1(d)) to explore if there is a correlation among the solutions (generated in each trial). The resulting Hamming distance exhibits a Gaussian distribution implying that the solutions are widely different from each other, and that the system gets trapped randomly in any one of the many local minima. This also indicates

that the trajectory of the system in the phase space likely changes from run-to-run. These results motivate the further exploration of the dynamical properties of oscillator Ising machines, specifically from theoretical standpoint. Consequently, we have incorporated control theoretic methods to analyze OIM. The following section discusses such an analysis.

4.1. Control theoretic analysis of dynamical systems

Empirically established annealing methods are extensively used to improve solution quality obtained from different oscillator-based dynamical systems. However, control theoretic analysis to better understand the dynamical properties of such computing platforms is largely unexplored. Consequently, in this effort, stability analysis of fixed points has been utilized to analyze the dynamics of oscillator-based computing systems [109]. Fixed points are defined as the points in the phase space where the time derivatives of all the variables become zero. A dynamical system gets stuck in a fixed point if it is stable. Thus, stability analysis could be helpful to choose system parameters so that it can avoid such fixed points by destabilizing them. This research effort is limited to the stability analysis of fixed points of oscillator Ising machines (OIMs). However, a similar analysis can be done for other oscillator-based dynamical systems described in Chapter 2, and 3.

In groundbreaking work by Wang et al. [67], the authors demonstrated that a global minimum of the cost function (referred to as the Lyapunov function by the authors) for a topographically equivalent coupled oscillator network under second harmonic injection can be equivalent to computing a global minimum of H . While the minimization of the cost function in OIM [67], as well as their implementation [50],[51],[71],[110] has been explored

in prior work, the stability of the globally optimal and locally optimal spin (phase) configurations and the resulting impact on the OIM dynamics has been largely unexplored. The works by Erementchouk et al. [111] and Böhm et al. [112] are a few examples that aim to investigate the dynamics of the OIM while a few more works have focused on analyzing the dynamical properties in spiking neural network [113]-[116]. Consequently, understanding the properties of the OIM as a nonlinear dynamical system and elucidating their impact on the computational properties are the primary focus of this effort.

4.1.a. Stability analysis of fixed points of an OIM using the linearization method

The dynamics of the OIM are such that the oscillator phases settle to $\theta \in \{0, \pi\}$, which subsequently, represent $s = \pm 1$ assignment to the nodes. The computational capability of this system arises from the fact that the resulting phase configuration of the oscillators will correspond to a ground state of H . The cost function $E(\theta(t))$ and the corresponding system dynamics are respectively presented as (also shown in Chapter 1)

$$E(\theta(t)) = -K \sum_{i,j=1, j \neq i}^N W_{ij} \cos(\theta_i - \theta_j) - K_s \sum_{i=1}^N \cos(2\theta_i(t)) \quad (4.1)$$

and

$$\frac{d\theta_i(t)}{dt} = -K \sum_{j=1, j \neq i}^N W_{ij} \sin(\theta_i - \theta_j) - K_s \sin(2\theta_i(t)) \quad (4.2)$$

where $[W]$ represents the coupling matrix between nodes, K and K_s represent the strength of coupling among the oscillators and the strength of the second harmonic injection signal, respectively. For the MaxCut problem, the weight of an edge E_{ij} in the input graph is related to the coupling matrix by the relation $W_{ij} = -E_{ij}$. Using equations (4.1) and (4.2),

it can be shown that $\frac{dE(\theta)}{dt} = -2 \sum_{i=1}^N \left(\frac{d\theta_i(t)}{dt} \right)^2 \leq 0$ [67],[85], which consequently implies that the system will evolve towards the ground state, except when $\frac{dE(\theta)}{dt} = 0$ (i.e., $\frac{d\theta(t)}{dt} = 0$). A point in the phase space where $\frac{d\theta(t)}{dt} = 0$ defines a fixed point and there are multiple such points in the phase space. In fact, every possible spin assignment and its equivalent in terms of the oscillator phases $\{\theta_1, \theta_2, \dots, \theta_N\}$, where $\theta_i \in \{0, \pi\}$, can correspond to a fixed point. Consequently, the phase space contains 2^N fixed points in the system (for $\theta \in \{0, \pi\}$); 2^{N-1} points when symmetricity in the solutions is considered. The fixed points lying at the lowest energy, if stable, would correspond to an (globally) optimal solution to the Ising model while stable fixed points that do not lie at the lowest energy would correspond to locally optimal (globally sub-optimal) solutions. Furthermore, even for the same energy (including the ground state), some fixed points (i.e., spin configurations) may be stable while others may not. This implies that the system may intrinsically favor certain Ising solutions over others leading to a *biased OIM*. Consequently, engineering the system stability can have significant impact on the computational characteristics and the performance of the system.

To elucidate our approach, we consider an illustrative randomly generated unweighted graph with 20 nodes and 152 edges as shown in Fig. 4.2a. Figure 4.2b shows a histogram for the energy (quantified using H here) for all possible solutions. It can be observed in Fig. 4.2b that the graph has 22 spin configurations that yield the minimum energy ($H = -28$). However, as alluded to above, the system dynamics may not always be stable for all the 22 globally optimal configurations.

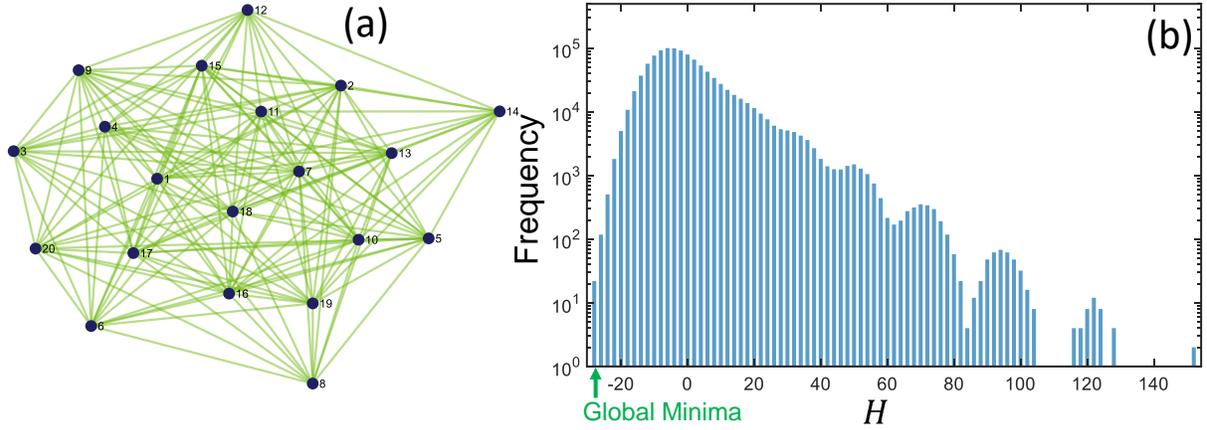


Figure 4.2. (a) An illustrative randomly generated graph with 20 nodes and 152 edges. (b) Corresponding histogram of energy (H) for all (2^{20}) possible spin configurations.

In order to investigate the stability of the globally and locally optimal phase configurations, we analyze the Lyapunov exponents ($\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_N$) for the system dynamics. Lyapunov exponents provide a powerful mathematical tool for analyzing the stability of non-linear dynamical systems [117]. Considering that dynamics are continuous-time, and the corresponding Jacobian matrix is symmetric [118], the Lyapunov exponents are the same as the eigenvalues of the Jacobian matrix. For a phase configuration to be stable, all Lyapunov exponents should be negative. The Jacobian matrix (J) for the OIM (assuming symmetric unweighted edges i.e., $W_{ij} = W_{ji}$) can be defined as,

$$J = \begin{bmatrix} E(1,1) & -KW_{12}\cos(\theta_1 - \theta_2) & -KW_{13}\cos(\theta_1 - \theta_3) & \dots & -KW_{1N}\cos(\theta_1 - \theta_N) \\ -KW_{12}\cos(\theta_1 - \theta_2) & E(2,2) & -KW_{23}\cos(\theta_2 - \theta_3) & \dots & \vdots \\ -KW_{13}\cos(\theta_1 - \theta_3) & -KW_{23}\cos(\theta_2 - \theta_3) & E(3,3) & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ -KW_{1N}\cos(\theta_1 - \theta_N) & \dots & \dots & \dots & E(N,N) \end{bmatrix} \quad (4.3)$$

where $E(i, i) = -K \sum_{j=1, j \neq i}^N W_{ij} \cos(\theta_i - \theta_j) - 2K_s \cos(2\theta_i)$. The Eigenvalues of J for a given point in the phase space where $\frac{d\theta(t)}{dt} = 0$ yield the Lyapunov exponents at that point. Since *all* the Lyapunov exponents need to be negative in order for an energy minimum to

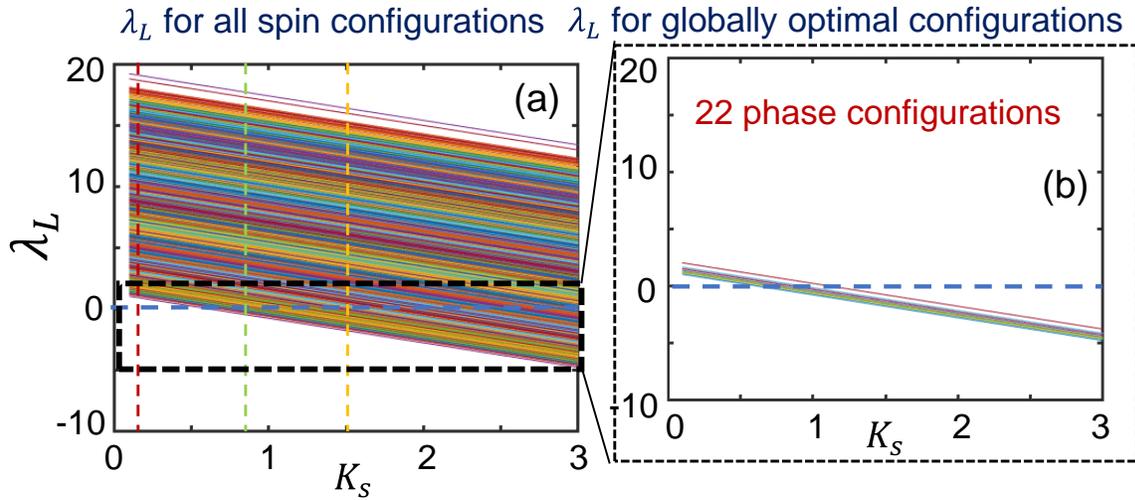


Figure 4.3. (a) Evolution of the largest Lyapunov exponent (λ_L) as a function of K_s for all spin configurations; (b) Evolution of λ_L as a function of the K_s for the subset of globally optimal phase configurations. Note: $\lambda_L > 0$ implies that the particular solution is unstable.

be stable, we focus on the largest Lyapunov exponent (referred to here as λ_L) since all other exponents will be smaller than λ_L .

Figure 4.3a shows the evolution of the *largest* Lyapunov exponent (λ_L) as a function of K_s ($K = 1$) for the representative graph shown in Fig. 4.2a. All the 2^{20} possible phase configurations are considered. The evolution of λ_L for only the globally optimal solutions is emphasized in Fig. 4.3b. It can be observed that the stability of a spin configuration is significantly impacted by the strength K_s (relative to K) of the second harmonic injection signal. In fact, if K_s is small enough (< 0.5 for the graph considered here), then the ground states, i.e., globally optimal configurations themselves can become unstable. In such a scenario, the system will cease to behave as an Ising machine – the ground state energy of the system will then correspond to an oscillator phase configuration where some or all oscillator phases do not settle to 0 or π .

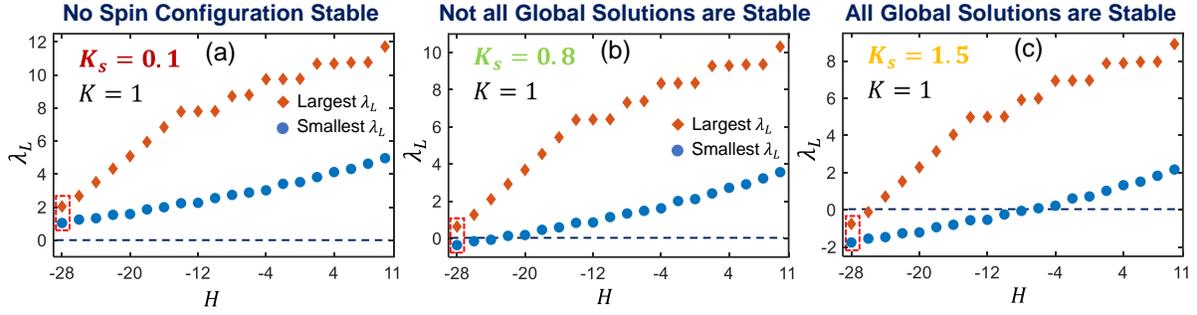


Figure 4.4. Minimum (blue) and maximum (orange) λ_L for phase configurations lying at a particular energy (H). Three values of K_s are considered (a) $K_s = 0.1$: Since all spin configurations (including the globally optimal solutions) are unstable, it implies that the oscillator platform will cease to behave as an Ising machine. (b) $K_s = 0.8$: Some globally optimal solutions are stable while others are unstable. Additionally, a few locally optimal low energy solutions are also stabilized; (c) $K_s = 1.5$: All globally optimal solutions are stable. The red box indicates the globally optimal solutions.

Next, for different K_s ($K = 1$), we analyze the distribution of λ_L for all phase configurations lying at a given energy (H). Figs. 4.4a-c show the maximum and the minimum value of λ_L for phase configurations corresponding to a given H , computed for three different values of K_s (0.1, 0.8, 1.5), respectively. In Fig. 4.4a, it can be observed that since λ_L for all spin configurations (including the globally optimal solutions lying at $H = -28$) are greater than zero, the ground state of the oscillator platform will not be achieved for $\theta \in \{0, \pi\}$. Consequently, it is expected that the oscillator platform will cease to function as an Ising machine. When $K_s = 0.8$ (Fig. 4.4b), it can be observed that the maximum and the minimum value of λ_L for the globally optimal solutions straddle zero, i.e., λ_L for some solutions is less than zero whereas it is greater than zero for others. This implies that only a fraction of the globally optimal spin configurations is stable, and consequently, the system dynamics will preferentially converge to the stable (globally optimal) solutions. This creates a biased OIM that favors the stable states over the unstable ones. Additionally, it is also noteworthy to point out that some of the fixed points corresponding to locally optimal (but globally sub-optimal) solutions lying at low energies

$(H = -24, -26)$ are also stabilized. This indicates that the system may potentially get trapped in one of these states, leading to sub-optimal solutions. However, the value of K_s is such that the solutions lying at higher energies ($H > -24$) are destabilized, preventing the system from getting trapped in those states. Finally, when the strength of the second harmonic injection is increased further to $K_s = 1.5$ (Fig. 4.4c), it can be observed that all the globally optimal solutions are stabilized. Additionally, increasing the strength of the second harmonic injection also increases the number and energy of the locally optimal (globally sub-optimal) solutions where the system dynamics can be stabilized. Consequently, this should increase the probability of the system getting trapped at a local minimum. Here, we only consider the fixed points associated with Ising solutions ($\theta \in \{0, \pi\}$). There may be other fixed points in the phase space corresponding to configurations where $\theta \notin \{0, \pi\}$. We also note that for the OIM dynamics, the stability of the fixed points can also be alternatively analyzed by using the second order partial derivative test.

We verify the system behavior predicted above using simulations shown in Fig. 4.5. We consider an oscillator network that is equivalent to the graph considered in Fig. 4.2a, and subsequently, evaluate the dynamics for different second harmonic injection strengths. We simulate the system dynamics (2) using MATLAB's® SDE (stochastic differential equation) solver, where we consider a time and phase independent noise amplitude of $K_n = 0.005$. When $K_s = 0.1$, the oscillator phases, as expected, do not converge to $\{0, \pi\}$ and the oscillator platform does not behave as an Ising machine. For larger injection strengths ($K_s = 0.8, 1.5$), it can be observed that the oscillator phases are binarized to $\{0, \pi\}$, validating the system's ability to function as an Ising machine.

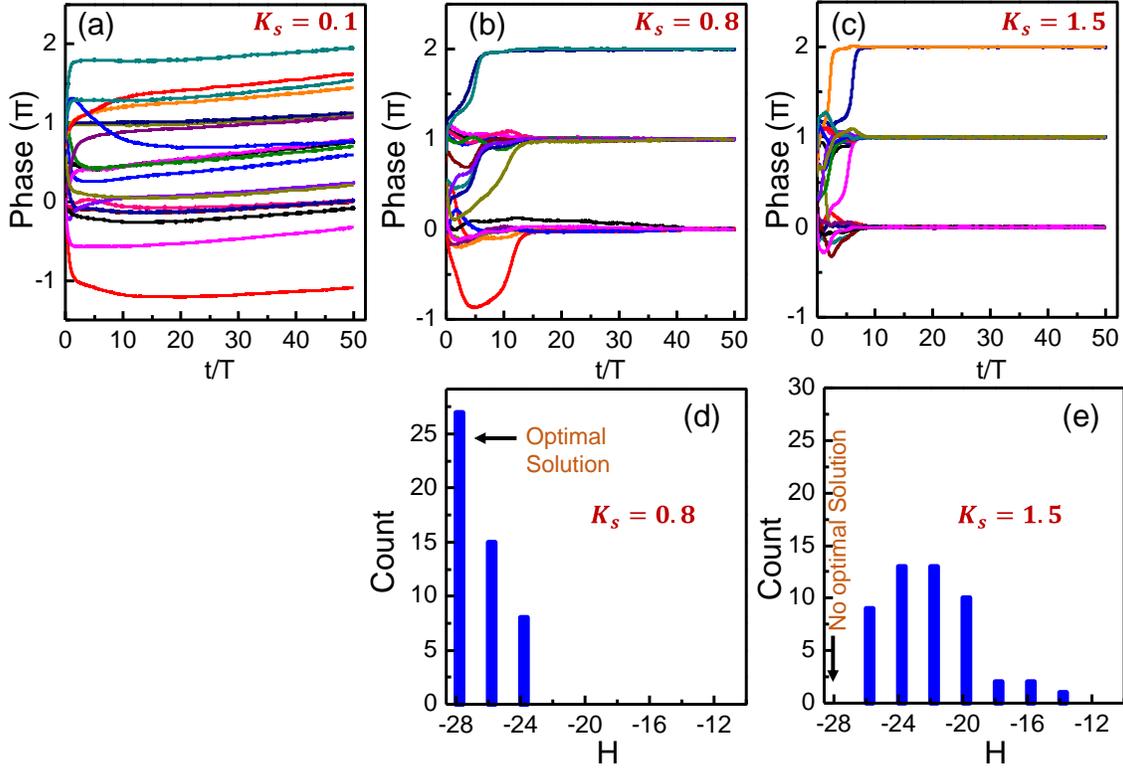


Figure 4.5. Temporal evolution of the oscillator phases for (a) $K_s = 0.1$; (b) $K_s = 0.8$; (c) $K_s = 1.5$, respectively ($K = 1$). The oscillator network is topologically equivalent to the graph considered in Fig. 4.2(a). When $K_s = 0.1$, the phases do not converge to $\{0, \pi\}$, and thus, the system does not behave as an Ising machine. Measured H for (d) $K_s = 0.8$. (e) $K_s = 1.5$ over 50 trials. Since a smaller number of locally optimal solutions are stabilized at $K_s = 0.8$ compared to $K_s = 1.5$, the system yields better solution quality. No solution is obtained for $K_s = 0.1$.

Figures 4.5d and 4.5e show a histogram of the computed H for $K_s = 0.8$ and $K_s = 1.5$, respectively, over 50 trials with randomly generated initial conditions. The spin assignments and the solutions are not computed for $K_s = 0.1$ since the system does not behave as an Ising machine. Since $K_s = 0.8$ only stabilizes some globally optimal solutions and some phase configurations that lie at low energy ($H = -26, -24$), it can be observed that the system dynamics always converge to one of these states. In contrast, increasing the second harmonic injection strength to $K_s = 1.5$ stabilizes all the 22 global solutions as well as many other phase configurations that lie at higher energies (Fig. 4.5c). Consequently, it can be observed in Fig. 4.5e that the system dynamics exhibit a higher

probability of getting trapped at a local minimum (sub-optimal spin configuration). In fact, over 50 trials, the system never converges to a globally optimal solution ($H = -28$). This indicates that the ability to engineer the stability of the local minima can significantly impact the computational performance of the OIM.

4.1.b. Assessing the stability of fixed points of an OIM from energy landscape

The dynamics and the corresponding cost / energy function of an OIM can be respectively described by (also shown in previous subsection):

$$\frac{d\theta_i}{dt} = -K \sum_{i=1, i < j}^N W_{ij} \sin(\theta_i - \theta_j) - K_s \sin(2\theta_i(t)) = f_i(\theta) = -\frac{1}{2} (\nabla E)_i \quad (4.4)$$

$$E(\theta(t)) = -K \sum_{i,j=1, j \neq i}^N W_{ij} \cos(\theta_i - \theta_j) - K_s \sum_{i=1}^N \cos(2\theta_i(t)) \quad (4.5)$$

where, $\theta = (\theta_1, \theta_2, \theta_3, \dots, \theta_N)$ represents the oscillator phases, and K and K_s are the coupling and second harmonic injection strength, respectively. From Eq. (4.4) and (4.5), it can be deduced that,

$$\frac{dE}{dt} = \sum_{i=1}^N \frac{\partial E}{\partial \theta_i} \frac{d\theta_i}{dt} = -2 \sum_{i=1}^N \left(\frac{d\theta_i}{dt} \right)^2 \leq 0 \quad (4.6)$$

The dynamics described in Eq. (4.4) reveal that the system of equations has multiple fixed points, where $\frac{d\theta}{dt} = 0$ (also, $-(\nabla E) = 0$). In fact, every spin configuration $(\theta_1, \theta_2, \theta_3, \dots, \theta_N)$, $\theta_i \in \{0, \pi\}$ is a fixed point, resulting in 2^N such fixed points. We note that there might be other fixed points defined by $(\theta_1, \theta_2, \theta_3, \dots, \theta_N)$ where $\theta_i \notin \{0, \pi\}$. The fixed points $(\theta_1, \theta_2, \theta_3, \dots, \theta_N)$ with the lowest energy represent the optimal solutions to the Ising

Hamiltonian while the rest represent sub-optimal solutions. In the previous section, we analyzed the stability of the fixed points, corresponding to various spin configurations, of the OIM. Specifically, we showed that tuning the ratio of the coupling strength among the oscillators (K) to the strength of the second harmonic injection (K_s) has a dramatic impact on the stability of the globally optimal and sub-optimal fixed points. This analysis was performed by using the linearization method, i.e., by computing the Lyapunov exponents, which in this case, are defined by the eigenvalues of the Jacobian matrix (J). A fixed point (globally optimal or not) is attractive if all eigenvalues at that point are negative; and unstable if at least one eigenvalue is positive.

While the Jacobian analysis essentially entails working with the first order derivatives, the purpose of this work is to present an alternate approach, based on the second order derivatives test of E (using the Hessian Matrix). This novelty of this work shows that the stability of the fixed points of a class of gradient systems such as oscillator Ising machines can be directly determined from their energy landscape. This study demonstrates that an oscillator Ising machine can get stabilized at any energy minima, i.e., all local minima in the energy function act as attractors.

We show that for an OIM whose dynamics are of the form $-\alpha(\nabla E)_i = \frac{d\theta_i}{dt}$ ($\alpha = \frac{1}{2}$ for OIM), the stability of the fixed points can be analyzed using the eigenvalues of the Hessian Matrix (H_E) of the energy function. This is possible since for oscillator Ising machines, the energy function is a sum of sinusoids (shown in Eq. (4.5)). Consequently, it is continuous and infinitely differentiable. A fixed point is attractive if all eigenvalues of H_E are positive. For the unstable case, if some or all of the eigenvalues are negative, then the fixed point

is a saddle point, or a local maximum, respectively. To prove this, we establish the equivalence between the two methods [119].

The Hessian matrix for $E(\theta)$ can be computed as,

$$\begin{aligned}
 H_E &= \begin{bmatrix} \frac{\partial^2 E}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_N} \\ \frac{\partial^2 E}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial \theta_N \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_N \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_N} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\partial^2 E}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_1} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_1} \\ \frac{\partial^2 E}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial \theta_1 \partial \theta_N} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_N} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_N} \end{bmatrix} \tag{4.7}
 \end{aligned}$$

Here, the symmetricity of the second derivatives ($\frac{\partial^2 E}{\partial \theta_i \partial \theta_j} = \frac{\partial^2 E}{\partial \theta_j \partial \theta_i}$) is used. Using Eq. (4.4),

$\frac{d\theta_i}{dt} = -\frac{1}{2}(\nabla E)_i = -\frac{1}{2}\frac{\partial E}{\partial \theta_i} = f_i$, Eq. (4.7) can now be expressed as:

$$H_E = \begin{bmatrix} \frac{\partial^2 E}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_1} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_1} \\ \frac{\partial^2 E}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial \theta_1 \partial \theta_N} & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_N} & \cdots & \frac{\partial^2 E}{\partial \theta_N \partial \theta_N} \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} \frac{\partial(-2f_1)}{\partial\theta_1} & \frac{\partial(-2f_1)}{\partial\theta_2} & \cdots & \frac{\partial(-2f_1)}{\partial\theta_N} \\ \frac{\partial(-2f_2)}{\partial\theta_1} & \frac{\partial(-2f_2)}{\partial\theta_2} & \cdots & \frac{\partial(-2f_2)}{\partial\theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial(-2f_N)}{\partial\theta_1} & \frac{\partial(-2f_N)}{\partial\theta_2} & \cdots & \frac{\partial(-2f_N)}{\partial\theta_N} \end{bmatrix} \\
&= -2 \begin{bmatrix} \frac{\partial f_1}{\partial\theta_1} & \frac{\partial f_1}{\partial\theta_2} & \cdots & \frac{\partial f_1}{\partial\theta_N} \\ \frac{\partial f_2}{\partial\theta_1} & \frac{\partial f_2}{\partial\theta_2} & \cdots & \frac{\partial f_2}{\partial\theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial\theta_1} & \frac{\partial f_N}{\partial\theta_2} & \cdots & \frac{\partial f_N}{\partial\theta_N} \end{bmatrix} = -2J \tag{4.8}
\end{aligned}$$

Eq. (4.8) reveals that in OIMs, the Jacobian matrix is half of the negative of the Hessian matrix ($J = -\frac{1}{2}H_E$). Consequently, the magnitude of the eigenvalues of the Jacobian matrix (λ_J) will be half of the magnitude of the eigenvalues of the Hessian matrix (λ_{H_E}) but with the opposite sign, i.e., $\lambda_J = -\frac{1}{2}\lambda_{H_E}$. This implies that the condition for a fixed point to be attractive, when analyzing the Hessian matrix, is that all eigenvalues be positive. When some eigenvalues are negative, it entails that the fixed point is a saddle point; when all eigenvalues are negative, it can be inferred that the fixed point is a maximum. For the general class of gradient descent systems defined by $-\alpha(\nabla E)_i = \frac{d\theta_i}{dt}$ ($\alpha > 0$), $\lambda_J = -\alpha\lambda_{H_E}$.

Chapter 5

5. Future Work

This chapter discusses the potential of coupled oscillators in statistical sampling.

The primary focus of this dissertation is formulating oscillator-based dynamical systems to directly solve many COPs and performing theoretical analysis for such systems. Solving COPs is associated with the ground state search process. However, the extensive phase space search ability of dynamical systems inspires their exploration to utilize them in the broader field of statistical sampling. In this effort, we have preliminary performed some analysis to explore if oscillator Ising machines (OIMs) can be utilized to perform statistical sampling. This analysis presents promising results which in turn provide broader directions for future work on dynamical system-based computing.

5.1. Statistical sampling with OIM

Statistical sampling is an essential part of machine learning. For instance, Boltzmann machines utilize Gibbs sampling for training [120]. Since phase spaces of objects like images are tremendously large, they cannot be identified deterministically, rather identified statistically. For that, a neural network is trained with a set of objects. The expectation from a training is that it will replicate the true distribution (distribution of the complete phase space) with high accuracy. In this dissertation, preliminary investigations are conducted to obtain statistical samples with OIM. In a Boltzmann machine, the energy function for the spin / neuronal configuration σ , can be written as,

$$E_{\sigma} = - \sum_{i=1}^{n_v} a_i s_i - \sum_{j=1}^{n_h} b_j s_j - \sum_{i=1}^{n_v} \sum_{j=1}^{n_h} w_{ij} s_i s_j \quad (5.1)$$

where n_v and n_h are the numbers of visible and hidden layer neurons (spins), respectively; a_i and b_j are biases to the visible and hidden layer neurons, respectively; w_{ij} is the weight between the i^{th} visible layer and j^{th} hidden layer. A schematic diagram of a restricted Boltzmann machine (RBM) [120], a specific type of Boltzmann machine, is shown in Fig. 5.1.

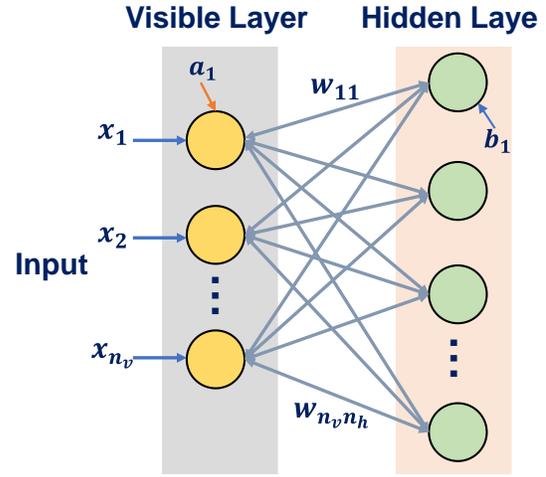


Figure 5.1. Schematic depiction of a restricted Boltzmann machine (RBM).

We have described throughout the earlier chapters that coupled oscillators can directly map graphs. Neural networks like RBM can be represented as graphs as well. Hence, oscillator-based dynamical systems can be explored to perform statistical sampling. Here, we mainly consider OIM for performing statistical sampling. Figure 5.2 shows an equivalence between an OIM and a statistical sampler. Oscillators can encode two neuronal states with their binary phases; 0 (π) phase encodes active (inactive) neuronal output or vice versa.

Very few works have demonstrated the generation of statistical samples with Ising machines. For instance, optical oscillator based coherent Ising machines have been studied for performing statistical sampling and training Boltzmann machines [121],[122]. In these works, noise has been considered as a key control parameter while obtaining

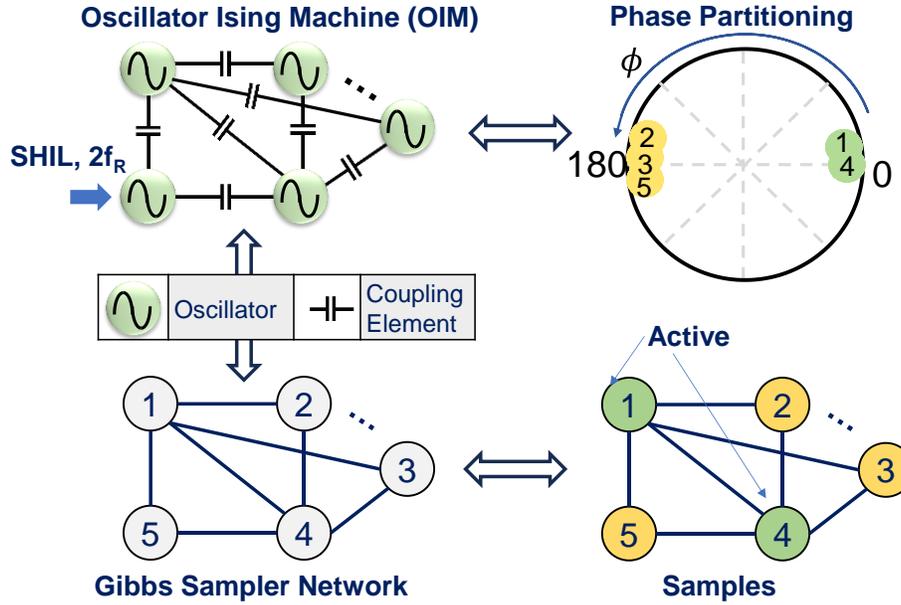


Figure 5.2. Equivalence between an OIM and a Gibbs sampling network.

samples. Gibbs sampling follows the Boltzmann distribution, where the probability of obtaining a configuration σ_i is computed by,

$$P(\sigma_i) = \frac{e^{-\frac{E_{\sigma_i}}{K_B T}}}{Z} = \frac{e^{-\frac{E_{\sigma_i}}{K_B T}}}{\sum_j e^{-\frac{E_{\sigma_j}}{K_B T}}} \quad (5.2)$$

where K_B is Boltzmann constant (a parameter usually considered 1), T is temperature, and the partition function Z is evaluated as the sum of the probabilities for all the possible configurations. Since the number of possible states grows exponentially with network size, it is impossible to determine Z for any reasonable network. Consequently, finding a finite distribution to replicate a true distribution is challenging. Gibbs sampling is an algorithm that is used to obtain a good-quality distribution from a finite number of samples. The Markov chain Monte Carlo (MCMC) algorithm (e.g., the Metropolis-Hastings (MH)) algorithm [123] is a widely used algorithm to obtain such samples and is therefore used for training neural networks [124]. These algorithms need serial updates which hinders

the training speed. This motivates the design of methods to parallelize statistical sampling obtaining approaches. Dynamical systems exhibit parallel updates and therefore can be considered as possible alternative candidates for obtaining fast samples.

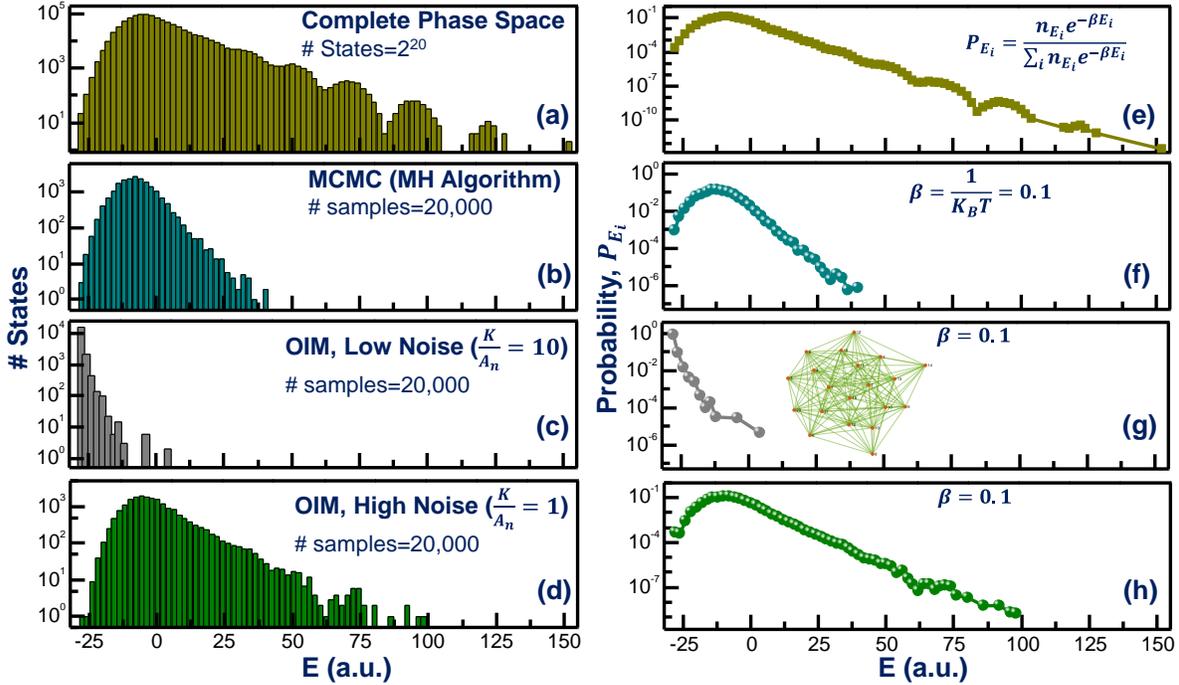


Figure 5.3. Gibbs sampling for a randomly generated 20-node network (inset Fig. 5.3g). (a) Complete phase space distribution, phase space distribution obtained from (b) MH algorithm, (c) OIM with low noise, (d) OIM with high noise. (e)-(h) Probability distribution of the energy states for all the cases shown in (a)-(d).

Here, we investigate the possibility of parallelizing Gibbs sampling with OIM. The OIM dynamics are shown in Eq. (1.6). With noise (Weiner process dw_t) it can be written as,

$$\frac{d\phi_i}{dt} = -k \sum_{j=1, j \neq i}^N J_{ij} \sin(\phi_i(t) - \phi_j(t)) - k_s \sin(2\phi_i(t)) + A_n dw_t \quad (5.3)$$

where A_n is noise amplitude. Here, we consider $K = 1$ and $\beta = \frac{1}{K_B T}$. Figure 5.3 shows the Gibbs sampling results for a 20-node randomly generated network, obtained by simulating an OIM. Figures 5.3a-d show the complete phase space distribution, phase

space distribution obtained using the MH algorithm, phase space distribution obtained using an OIM with low noise ($\frac{K}{A_n} = 10$), and phase space distribution obtained using an OIM with 10 times larger noise ($\frac{K}{A_n} = 1$). Figures 5.3e-h present the probability distribution of the states for all the cases shown in Fig. 5.3a-d. It can be observed that OIM with low noise mainly occupies the lower energy states and hence is good for solving the Ising problem, i.e., solving COPs. On the contrary, when high noise is used the probability distribution generates the true distribution with very good matching. Thus, OIM can provide Gibbs samples with the help of high noise.

5.2. Parallelization of statistical sampling with OIM

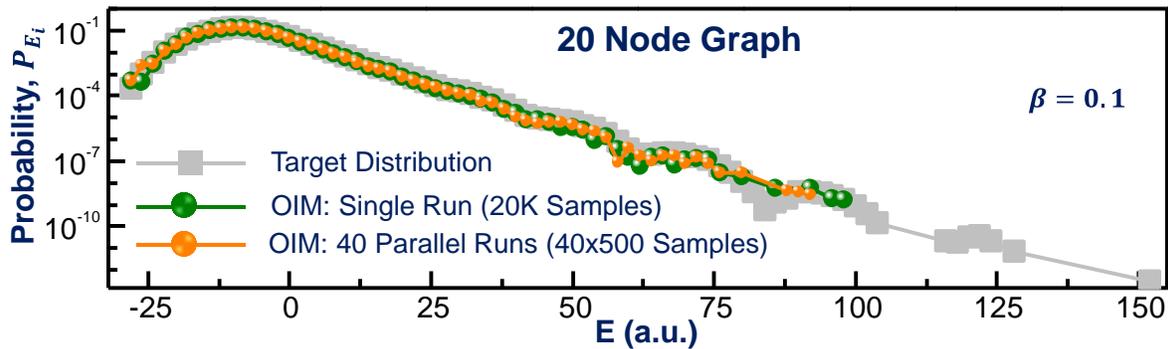


Figure 5.4. Comparison of the probability distribution of the 20-node network (Fig. 5.3g) obtained using 40 parallel OIM samplers with samples obtained using a single run OIM sampler and the true / target distribution.

Subsequently, we evaluate the possibility of obtaining OIM-based samples by running multiple OIMs in parallel to enhance sampling speed. Figure 5.4 shows the probability distribution for the 20-node network (Fig. 5.3g) obtained using 40 parallel OIM runs and compares it with the probability distribution obtained from a single OIM run as well as with the true distribution. Here, the total number of samples obtained from the parallel runs is

kept the same as the number of samples obtained from the serial run. We show that parallel runs provide similar matching as the single (serial) run. However, parallel runs can accelerate the sampling significantly. For instance, if we use n OIM hardware (or an n -core processor), we will be able to achieve around n times speedup while obtaining samples. Thus, these findings propel the exploration of oscillator-based dynamical systems in the field of statistical sampling and machine learning.

Chapter 6

6. Conclusion

In summary, this dissertation work expands the design methodologies and advances the theoretical foundation for coupled oscillator-based analog computing. Prior works established that emerging methods for solving combinatorial optimization mainly rely on Ising machines. In principle, such Ising machines are capable of solving any combinatorial optimization problem (COP) and are therefore envisioned as a universal COP-solving computer. However, in practice, Ising machines are not efficient in solving broader classes of COPs due to the necessity of problem size expansion through a preconversion to Ising problems and thus demand more computing resources while solving them. Being motivated by the limitations of Ising machines, in this dissertation, a novel approach is developed to design oscillator Potts machines with a phase sensitive coupling function that can be modified to map a wide range of COPs without converting them to other problems. To assess the efficacy of the oscillator Potts machine, the computational model for the Max-K-Cut is implemented as an algorithm on an FPGA. The FPGA implementation demonstrates up to 390x speedup in solving Max-K-Cut over a state-of-the-art GPU-based Ising solver for benchmarking graphs up to 10,000 nodes while maintaining similar solution quality.

Another key focus of this dissertation is to map higher order interaction among spins / variables onto dynamical systems. With this objective, multiple dynamical systems are formulated to map several higher order COPs, including the Boolean SAT problem. In order to create a more general method, an extension of the oscillator Ising machine is

developed that can directly map any higher order Ising spin interaction. Leveraging methods from combinatorics, this dissertation also develops alternative expressions to represent higher order Ising spin interactions with nonlinear transforms of second order interactions.

Since dynamical properties of oscillator-based dynamical systems have been mostly unexplored, this dissertation incorporates control theoretic tools to analyze the dynamical properties of such oscillator-based systems. The stability of the fixed points is evaluated using the linearization technique to understand the role of the system parameters on the computational performance. The theoretical analysis results provide insights into the dynamical properties of coupled oscillator-based computing systems, providing toolsets to improve computational performance of such approaches using theoretical analysis, while prior works mainly relied on empirically established methods.

Finally, this dissertation explores the possibility of utilizing oscillator-based dynamical systems to perform statistical sampling such as Gibbs sampling. Such sampling is performed by introducing large noise in oscillator-based dynamical systems (e.g., Ising machines). The results motivate the further exploration of coupled oscillators for sampling and machine learning.

Thus, this dissertation broadens the state-of-the-art of dynamical system-based computing by introducing new methods to design such systems and bringing theoretical toolsets to improve their computational performance.

Appendices

Appendix I

Details of the derivation of the dynamics of the oscillator-based NAE-SAT solver

Here, we describe in detail the steps involved in the derivation of Eq. (3.35) (Chapter 3, Section 3.2). The dynamics described by Eq. (3.34) (Chapter 3, Section 3.2) are shown here (again) in:

$$\begin{aligned} \frac{d\phi_i}{dt} = & -\frac{A}{2} \left[\sum_{m=1}^M \left(c_{mi} \sin(t + \phi_i) \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \right) \right. \\ & \left. - \sum_{m=1}^M \left(\frac{1}{2} c_{mi}^2 \sin(2(t + \phi_i)) \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \right) \right] \\ & - \sin(2t + 2\phi_i) \cdot [A_s \cos(2t)] \end{aligned} \quad (A1)$$

The $\cos(\cdot)$ terms in Eq. (A1) are expressed by the following equation,

$$\begin{aligned} & \left(\prod_{j=1; j \neq i}^N \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right) \right)^2 \\ & = \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \cos \left(\left(\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \right) t + \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j \right) \end{aligned} \quad (A2)$$

Substituting (A2) in (A1),

$$\begin{aligned}
& \frac{d\phi_i}{dt} \\
&= -\frac{A}{2} \left[\sum_{m=1}^M \left(c_{mi} \sum_{\mu_N=-2}^2 \cdots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \sin(t + \phi_i) \cdot \cos \left(\left(\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \right. \right. \right. \right. \\
& \left. \left. \left. + \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j \right) \right) \right) \\
& - \sum_{m=1}^M \left(\frac{1}{2} c_{mi}^2 \sum_{\mu_N=-2}^2 \cdots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \sin(2(t + \phi_i)) \cdot \cos \left(\left(\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \right. \right. \right. \\
& \left. \left. \left. + \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j \right) \right) \right) \right] - \sin(2t + 2\phi_i) \cdot [A_S \cos(2t)]
\end{aligned} \tag{A3}$$

Eq. (A3) can be written as,

$$\begin{aligned}
& \frac{d\phi_i}{dt} \\
&= -\frac{A}{2} \left[\sum_{m=1}^M \left(c_{mi} \sum_{\mu_N=-2}^2 \cdots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \chi^{(1)}(t, \phi_i) \cdot B(t, \phi) \right) \right) \\
& - \sum_{m=1}^M \left(\frac{1}{2} c_{mi}^2 \sum_{\mu_N=-2}^2 \cdots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2, \dots, \mu_N; \neq \mu_i} \chi^{(2)}(t, \phi_i) \cdot B(t, \phi) \right) \right) \\
& - \sin(2t + 2\phi_i) \cdot [A_S \cos(2t)]
\end{aligned} \tag{A4}$$

where, $\chi^{(1)}(t, \phi_i)$ and $\chi^{(2)}(t, \phi_i)$ are the first and the second harmonics of the perturbation projection vector (PPV) of the oscillator, respectively; $B(t, \phi)$ is a perturbation which will have components ranging from the first harmonic to the $(2 \sum_{j=1, j \neq i}^N |c_{mj}|)^{th}$ harmonic. As

mentioned in Chapter 3, we assume that the phase evolution happens on a much slower time scale than the oscillation frequency. Wang et al. [85] showed that such an equation (Eq. (A4) here) can be formulated as a Multi-time Partial Differential Equation (MPDE) and can be approximated by averaging over the fast time. The resulting approximation is essentially a cross-correlation of the PPV and the perturbation. Since cross correlation of the i^{th} harmonic of the PPV with j^{th} harmonic of the perturbation will be zero in cases where $i \neq j$, Eq. (A4) reduces to,

$$\begin{aligned} \frac{d\phi_i}{dt} = & -\frac{A}{2} \left[\sum_{m=1}^M \left(c_{mi} \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i} \cdot \chi^{(1)}(t, \phi_i) \cdot B^{(1)}(t, \phi) \right) \right. \\ & \left. - \sum_{m=1}^M \left(\frac{1}{2} c_{mi}^2 \sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i} \cdot \chi^{(2)}(t, \phi_i) \cdot B^{(2)}(t, \phi) \right) \right] \\ & - \sin(2t + 2\phi_i) \cdot [A_S \cos(2t)] \end{aligned} \quad (A5)$$

Now, replacing χ and B in (A5) with their original expressions from Eq. (A3), and by taking the average with respect to the fast time, Eq. (A5) can be rewritten as,

$$\begin{aligned} \frac{d\phi_i}{dt} = & -A \sum_{m=1}^M \left(\sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi} Q_1 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^1 \sin \left(\left(\phi_i \right. \right. \right. \\ & \left. \left. \left. - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right|_{Q_1} \right) \right) \\ & + \\ & A \sum_{m=1}^M \left(\sum_{\mu_N=-2}^2 \dots \sum_{\mu_2=-2}^2 \sum_{\mu_1=-2}^2 c_{mi}^2 Q_2 C_{\mu_1, \mu_2 \dots \mu_N; \neq \mu_i}^2 \sin \left(2\phi_i - \sum_{j=1; j \neq i}^N |c_{mj}| \mu_j \phi_j(t) \right|_{Q_2} \right) \end{aligned} \quad (A6)$$

$$-A_{s1} \sin(2\phi_i)$$

where, $Q_1 = 1$ when $\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 1$ else $Q_1 = 0$; $Q_2 = 1$ when $\sum_{j=1; j \neq i}^N |c_{mj}| \mu_j = 2$ else $Q_2 = 0$. Eq. (A6) is the result shown in Eq. (3.35) in Chapter 3.

Appendix II

Derivation of the energy expression for a 3-SAT clause (oscillator-based NAE-SAT solver)

Here, the energy expression (Eq. (3.41) in Chapter 3) for a clause containing 3 literals is derived. We assume that the 3 literals correspond to three distinct variables (x_i, x_j, x_k). Thus, Eq. (3.34) from Chapter 3 can be written as,

$$\begin{aligned}
 \frac{d\phi_i}{dt} = & -\frac{A}{2} 2^{-2N+6} \left[\sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(c_{mi} \sin(t \right. \right. \\
 & + \phi_i) \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right)^2 \left(\frac{1 - c_{mk} \cos(t + \phi_k)}{2} \right)^2 \Big) \\
 & - \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(\frac{1}{2} c_{mi}^2 \sin(2t \right. \\
 & + 2\phi_i) \left(\frac{1 - c_{mj} \cos(t + \phi_j)}{2} \right)^2 \left(\frac{1 - c_{mk} \cos(t + \phi_k)}{2} \right)^2 \Big) - \sin(2t \\
 & \left. + 2\phi_i) [A_S \cos(2t)] \right] \quad (A7)
 \end{aligned}$$

Eq. (A7) can be simplified to,

$$\begin{aligned}
\frac{d\phi_i}{dt} = & -\frac{A}{2} 2^{-2N+2} \left[\sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(c_{mi} \sin(t + \phi_i) \left(1 + \frac{1}{2} c_{mj}^2 - 2c_{mj} \cos(t + \phi_j) \right. \right. \right. \\
& + \frac{1}{2} c_{mj}^2 \cos(2t + 2\phi_j) \left. \left. \left(1 + \frac{1}{2} c_{mk}^2 - 2c_{mk} \cos(t + \phi_k) \right. \right. \right. \\
& \left. \left. \left. + \frac{1}{2} c_{mk}^2 \cos(2t + 2\phi_k) \right) \right) \right. \\
& - \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left(\frac{1}{2} c_{mi}^2 \sin(2t + 2\phi_i) \left(1 + \frac{1}{2} c_{mj}^2 - 2c_{mj} \cos(t + \phi_j) \right. \right. \\
& + \frac{1}{2} c_{mj}^2 \cos(2t + 2\phi_j) \left. \left. \left(1 + \frac{1}{2} c_{mk}^2 - 2c_{mk} \cos(t + \phi_k) \right. \right. \right. \\
& \left. \left. \left. + \frac{1}{2} c_{mk}^2 \cos(2t + 2\phi_k) \right) \right) \right] - \sin(2t + 2\phi_i) \cdot [A_s \cos(2t)]
\end{aligned} \tag{A8}$$

After subsequent simplification of the terms and averaging over the fast time, Eq. (A8) can be expressed as,

$$\begin{aligned}
\frac{d\phi_i}{dt} = & -\frac{\pi A}{2} 2^{-2N+2} \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left[-2c_{mi}c_{mj} \left(1 + \frac{1}{2} c_{mk}^2 \right) \sin(\phi_i - \phi_j) \right. \\
& - 2c_{mi}c_{mk} \left(1 + \frac{1}{2} c_j^2 \right) \sin(\phi_i - \phi_k) - \frac{1}{2} c_{mi}c_{mj}c_{mk}^2 \sin(\phi_i + \phi_j - 2\phi_k) \\
& \left. - \frac{1}{2} c_{mi}c_{mk}c_{mj}^2 \sin(\phi_i + \phi_k - 2\phi_j) \right] \\
& + \frac{\pi A}{2} 2^{-2N+2} \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left[\frac{1}{4} c_{mi}^2 c_{mk}^2 \left(1 + \frac{1}{2} c_{mj}^2 \right) \sin(2\phi_i - 2\phi_k) \right. \\
& + \frac{1}{4} c_{mi}^2 c_{mj}^2 \left(1 + \frac{1}{2} c_{mk}^2 \right) \sin(2\phi_i - 2\phi_j) \\
& \left. + c_{mi}^2 c_{mj} c_{mk} \sin(2\phi_i - \phi_j - \phi_k) \right] - \pi A_s \sin(2\phi_i)
\end{aligned} \tag{A9}$$

Eq. (A9) can be further simplified as,

$$\begin{aligned}
\frac{d\phi_i}{dt} = & \frac{\pi A}{2} 2^{-2N+2} \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left[2c_{mi}c_{mj} \left(1 + \frac{1}{2}c_{mk}^2 \right) \sin(\phi_i - \phi_j) \right. \\
& + 2c_{mi}c_{mk} \left(1 + \frac{1}{2}c_{mj}^2 \right) \sin(\phi_i - \phi_k) \\
& + \frac{1}{2}c_{mi}c_{mj}c_{mk}^2 \sin(\phi_i + \phi_j - 2\phi_k) \\
& + \frac{1}{2}c_{mi}c_{mk}c_{mj}^2 \sin(\phi_i + \phi_k - 2\phi_j) \\
& + \frac{1}{4}c_{mi}^2c_{mk}^2 \left(1 + \frac{1}{2}c_{mj}^2 \right) \sin(2\phi_i - 2\phi_k) \\
& + \frac{1}{4}c_{mi}^2c_{mj}^2 \left(1 + \frac{1}{2}c_{mk}^2 \right) \sin(2\phi_i - 2\phi_j) \\
& \left. + c_{mi}^2c_{mj}c_{mk} \sin(2\phi_i - \phi_j - \phi_k) \right] - \pi A_s \sin(2\phi_i)
\end{aligned} \tag{A10}$$

Using Eq. (3.38) from Chapter 3, the Lyapunov function for the above dynamics can be defined as,

$$\begin{aligned}
E(\phi) = & \pi A \cdot 2^{-2N+1} \sum_{i=1}^N \sum_{\substack{m=1; i \neq j \neq k; c_{mi} \neq 0 \\ c_{mj} \neq 0, c_{mk} \neq 0}}^M \left[2c_{mi}c_{mj} \left(1 + \frac{1}{2}c_{mk}^2 \right) \cos(\phi_i - \phi_j) \right. \\
& + 2c_{mi}c_{mk} \left(1 + \frac{1}{2}c_{mj}^2 \right) \cos(\phi_i - \phi_k) + \frac{1}{2}c_{mi}c_{mj}c_{mk}^2 \cos(\phi_i + \phi_j - 2\phi_k) \\
& + \frac{1}{2}c_{mi}c_{mk}c_{mj}^2 \cos(\phi_i + \phi_k - 2\phi_j) \\
& + \frac{1}{8}c_{mi}^2c_{mk}^2 \left(1 + \frac{1}{2}c_{mj}^2 \right) \cos(2\phi_i - 2\phi_k) \\
& + \frac{1}{8}c_{mi}^2c_{mj}^2 \left(1 + \frac{1}{2}c_{mk}^2 \right) \cos(2\phi_i - 2\phi_j) \\
& \left. + \frac{1}{2}c_{mi}^2c_{mj}c_{mk} \cos(2\phi_i - \phi_j - \phi_k) \right] - \sum_{i=1}^N \frac{\pi A_s}{2} \cos(2\phi_i)
\end{aligned} \tag{A11}$$

Eq. (A11) expresses the energy function for the NAE-3-SAT problem. Eq. (A11) is the same as the result shown in Eq. (3.41).

Appendix III

Energy vs clause status for an NAE-3-SAT clause (oscillator-based NAE-SAT solver)

Here, we show that an NAE-3-SAT clause is satisfied only when the corresponding energy term associated with the clause is minimized (for all cases).

It can be observed from the table in Fig. A1 that an NAE-3-SAT clause is satisfied only when the corresponding energy term associated with the clause is minimized. Consequently, the decreasing nature of the energy function ensures that the system evolves towards a state that maximizes the number of satisfied NAE-3-SAT clauses.

NAE-SAT Clause $x = (x_i, x_j, x_k)$	$E(\phi_i, \phi_j, \phi_k)$ for a single clause ($\propto (T_{ij} + T_{jk} + T_{ki})$)							
	(0,0,0)	(0,0, π)	(0, π ,0)	(π ,0,0)	(π , π ,0)	(0, π , π)	(π ,0, π)	(π , π , π)
	$\equiv x$ = (1,1,1)	$\equiv x$ = (1,1,0)	$\equiv x$ = (1,0,1)	$\equiv x$ = (0,1,1)	$\equiv x$ = (0,0,1)	$\equiv x$ = (1,0,0)	$\equiv x$ = (0,1,0)	$\equiv x$ = (0,0,0)
$C_{NAE} = (x_i \vee x_j \vee x_k)$ $\cdot (\overline{x_i} \vee \overline{x_j} \vee \overline{x_k})$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$
$C_{NAE} = (x_i \vee x_j \vee \overline{x_k})$ $\cdot (\overline{x_i} \vee \overline{x_j} \vee x_k)$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$
$C_{NAE} = (x_i \vee \overline{x_j} \vee x_k)$ $\cdot (\overline{x_i} \vee x_j \vee \overline{x_k})$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$
$C_{NAE} = (\overline{x_i} \vee x_j \vee x_k)$ $\cdot (x_i \vee \overline{x_j} \vee \overline{x_k})$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$
$C_{NAE} = (\overline{x_i} \vee \overline{x_j} \vee x_k)$ $\cdot (x_i \vee x_j \vee \overline{x_k})$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$
$C_{NAE} = (x_i \vee \overline{x_j} \vee \overline{x_k})$ $\cdot (\overline{x_i} \vee x_j \vee x_k)$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$
$C_{NAE} = (\overline{x_i} \vee x_j \vee \overline{x_k})$ $\cdot (x_i \vee \overline{x_j} \vee x_k)$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$
$C_{NAE} = (\overline{x_i} \vee \overline{x_j} \vee \overline{x_k})$ $\cdot (x_i \vee x_j \vee x_k)$	E_1 $C_{NAE} = 0$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_2 $C_{NAE} = 1$	E_1 $C_{NAE} = 0$
$E_1 = \frac{189}{256}\pi A - \frac{3}{2}\pi A_s$				$E_2 = \frac{-51}{256}\pi A - \frac{3}{2}\pi A_s$				

Figure A1. $E(\phi_i, \phi_j, \phi_k)$ corresponding to a single NAE-3-SAT clause computed for all the possible combinations of the literals and phases. Here, $E_1 > E_2$.

Appendix IV

NAE-SAT solution with relative phase-based dynamics (oscillator-based NAE-SAT solver)

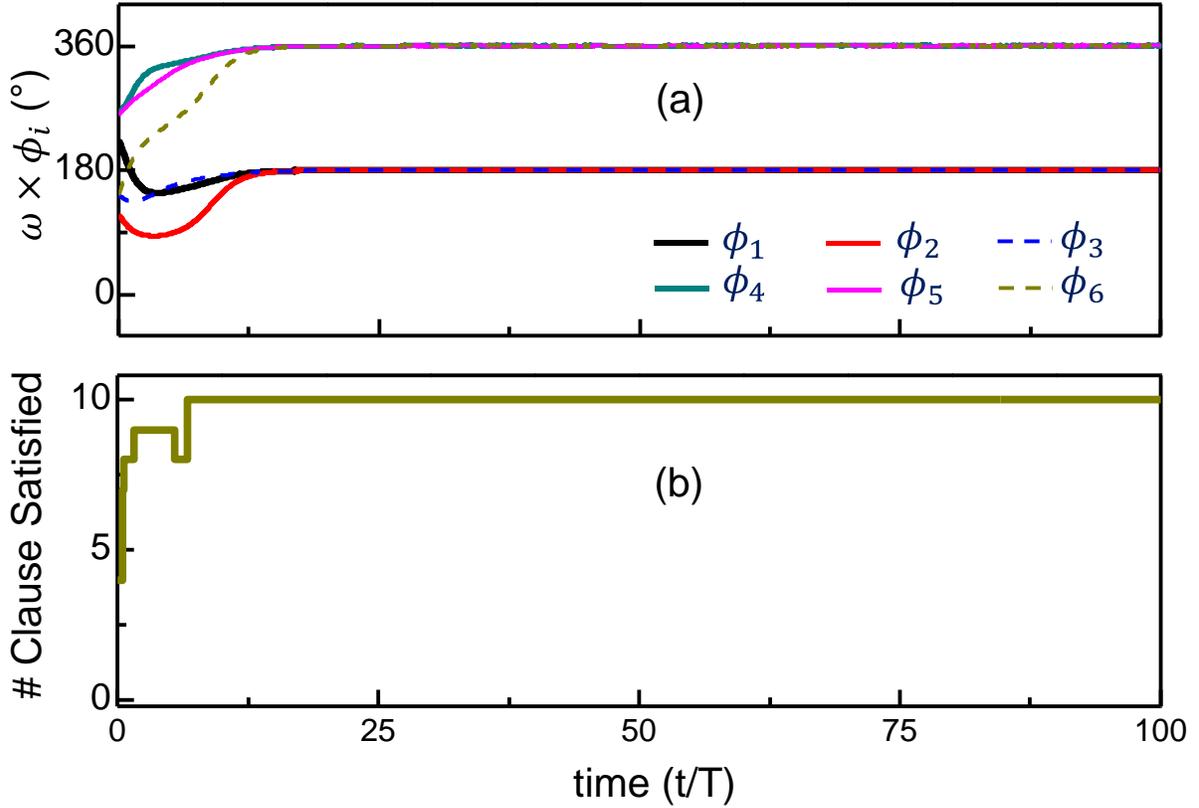


Figure A2. Evolution of (a) oscillator phases, calculated using Eq. (3.42); and (b) No. of satisfied clauses, with time. $\omega = 2\pi$ is used such that $T = 1$. The problem considered here is the same as that considered in Fig. 3.9.

We show in Fig. A2 that the NAE-3-SAT solution can also be computed using Eq. (3.42) from Chapter 3.

Appendix V

Dynamical system for the NAE-5-SAT problem (higher order Ising machine)

Here, we develop the formulation of the objective function and the corresponding dynamical system for the NAE-5-SAT (K is an odd number) problem. An NAE-5-SAT clause can be represented as,

$$C = (x_i \oplus x_j) \vee (x_i \oplus x_k) \vee (x_i \oplus x_l) \vee (x_i \oplus x_m) \vee (x_j \oplus x_k) \vee (x_j \oplus x_l) \vee (x_j \oplus x_m) \vee (x_k \oplus x_l) \vee (x_k \oplus x_m) \vee (x_l \oplus x_m) \quad (\text{A12})$$

In terms of Ising spins, the complement of C can be written as,

$$\begin{aligned} & \left(\frac{1+s_i s_j}{2}\right) \left(\frac{1+s_i s_k}{2}\right) \left(\frac{1+s_i s_l}{2}\right) \left(\frac{1+s_i s_m}{2}\right) \left(\frac{1+s_j s_k}{2}\right) \left(\frac{1+s_j s_l}{2}\right) \left(\frac{1+s_j s_m}{2}\right) \\ & \left(\frac{1+s_k s_l}{2}\right) \left(\frac{1+s_k s_m}{2}\right) \left(\frac{1+s_l s_m}{2}\right) \\ & = \frac{1}{2^4} (1 + s_i s_j + s_i s_k + s_i s_l + s_i s_m + s_j s_k + s_j s_l + s_j s_m + s_k s_l + s_k s_m \\ & \quad + s_l s_m + s_i s_j s_k s_l + s_i s_j s_k s_m + s_i s_j s_l s_m + s_i s_k s_l s_m + s_j s_k s_l s_m) \end{aligned} \quad (\text{A13})$$

Thus, the objective function for an NAE-5-SAT problem with M clauses can be written as,

$$H = - \sum_{m=1}^M \left(\sum_{\substack{i,j \\ i < j}}^N (-c_{mi} c_{mj} s_i s_j) + \sum_{\substack{i,j,k,l \\ i < j < k < l}}^N (-c_{mi} c_{mj} c_{mk} c_{ml} s_i s_j s_k s_l) \right) \quad (\text{A14})$$

where $c_{mi} = -1$ (+1) if the i^{th} variable appears in inverted (normal) form in the m^{th} clause; $c_{mi} = 0$ if the i^{th} variable is absent in the m^{th} clause.

Using the approach described in Chapter 3 (3.3), the corresponding Lyapunov function and the system dynamics can be formulated as,

Energy:

$$\begin{aligned}
 E = C \sum_{m=1}^M \left[\sum_{i,j,l < j}^N c_{mi} c_{mj} \cos(\phi_i - \phi_j) \right. \\
 \left. + \sum_{\substack{i,j,k,l \\ i < j < k < l}}^N c_{mi} c_{mj} c_{mk} c_{ml} \cos(\phi_i - \phi_j + \phi_k - \phi_l) + 1 \right] \\
 - \frac{C_s}{2} \sum_{i=1}^N \cos(2\phi_i)
 \end{aligned} \tag{A15}$$

Dynamics:

$$\begin{aligned}
 \frac{d\phi_i}{dt} = C \sum_{m=1}^M \left[\sum_{j=1}^N c_{mi} c_{mj} \sin(\phi_i - \phi_j) \right. \\
 \left. + \sum_{\substack{i \neq j \neq k \neq l \\ j < k < l}}^N c_{mi} c_{mj} c_{mk} c_{ml} \sin(\phi_i - \phi_j + \phi_k - \phi_l) \right] - C_s \sin(2\phi_i)
 \end{aligned} \tag{A16}$$

Equations (A14), (A15), and (A16) are also shown in Table 3.5.

Appendix VI

Verification of the expression for even order interaction (Alternative formulation of higher order Ising spin interaction)

We first numerically verify the formulation shown in Eq. (3.61), for all the possible spin configurations for even order interaction up to 20. Figure A3(a) shows the number of possible spin configurations for different even order interactions. Figure A3(b) shows the errors between the interactions obtained using the proposed formulation and the original interactions. For all the cases, the proposed transformation provides the exact spin interaction.

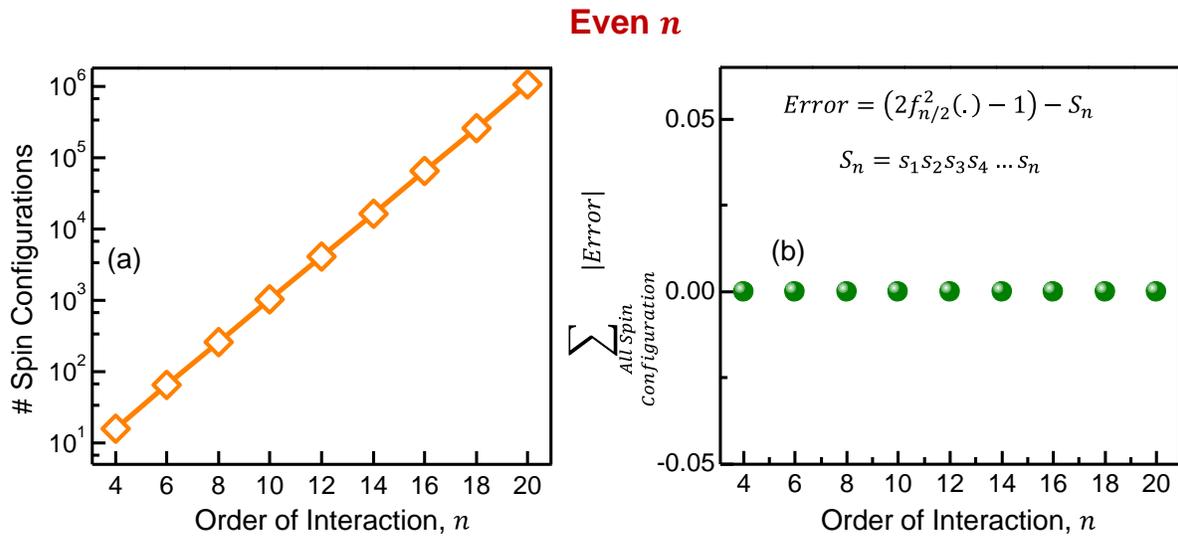


Figure A3. Numerical verification of the proposed formulation for even order spin interactions (4-20). (a) Number of possible spin configurations. (b) Sum of absolute errors over all the possible spin configurations indicating exact match.

Now we will analytically verify the formulation in Eq. (3.61). An Ising spin interaction term becomes minimum ($= -1$) when an odd number of spins in it are -1 . With the $2q$ spins

(s_1, \dots, s_{2q}) of S_{2q} , $\binom{2q}{q}$ number of q -order interaction terms $(\prod_{T \subseteq \{1,2,\dots,n\}} s_i)$ can be

$$\sum_{\substack{|T|=q \\ s_i \in T}}$$

produced. Here, $\binom{2q}{q}$ is an even number. We will prove that if any odd number of spins among the $2q$ spins are -1 , i.e., if $S_{2q} = -1$, $\frac{1}{2}\binom{2q}{q}$ number of q -order terms will be -1 .

Hence, $\sum_{\substack{T \subseteq \{1,2,\dots,n\} \\ |T|=q}} \prod_{s_i \in T} s_i = 0$, which will then lead to $f_q(\cdot) = 0$. This way, $S_{2q} = 2f_q^2(\cdot) -$

$1 = -1$ holds. On the contrary, if $S_{2q} = 1$, then $\sum_{\substack{T \subseteq \{1,2,\dots,n\} \\ |T|=q}} \prod_{s_i \in T} s_i \neq 0$. In this case, $f_q(\cdot) =$

1 . Again $S_{2q} = 2f_q^2(\cdot) - 1 = 1$ holds. These 2 conditions can be proved by establishing the following identities:

Identity I:

$$\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y-1}{x-2r-1} = \frac{1}{2} \binom{2x}{x} \quad (\text{A17})$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$; $x > y \geq r$.

Identity II:

$$\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y}{x-2r-1} \neq \frac{1}{2} \binom{2x}{x} \quad (\text{A18})$$

where $x, y \in \mathbb{Z}^+$; $r \in \mathbb{Z}_0^+$; $x \geq y > r$.

To prove Identity I, we can consider the Vandermonde's Identity [125] shown below:

$$\begin{aligned}
\binom{2x}{x} &= \sum_{r=0}^{2y+1} \binom{2y+1}{r} \binom{2x-2y-1}{x-r} \\
&= \binom{2y+1}{0} \binom{2x-2y-1}{x-0} + \binom{2y+1}{1} \binom{2x-2y-1}{x-1} \\
&\quad + \binom{2y+1}{2} \binom{2x-2y-1}{x-2} + \binom{2y+1}{3} \binom{2x-2y-1}{x-3} + \dots \quad (\text{A19}) \\
&\quad + \binom{2y+1}{2y-2} \binom{2x-2y-1}{x-2y+2} + \binom{2y+1}{2y-1} \binom{2x-2y-1}{x-2y+1} \\
&\quad + \binom{2y+1}{2y} \binom{2x-2y-1}{x-2y} + \binom{2y+1}{2y+1} \binom{2x-2y-1}{x-2y-1}
\end{aligned}$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$; $x > y$. This expansion has an even number of terms ($= 2y + 2$).

It can be shown that:

$$\binom{2y+1}{2y+1} \binom{2x-2y-1}{x-2y-1} = \binom{2y+1}{0} \binom{2x-2y-1}{x-0} \quad (\text{A20a})$$

$$\binom{2y+1}{2y} \binom{2x-2y-1}{x-2y} = \binom{2y+1}{1} \binom{2x-2y-1}{x-1} \quad (\text{A20b})$$

$$\binom{2y+1}{2y-1} \binom{2x-2y-1}{x-2y+1} = \binom{2y+1}{2} \binom{2x-2y-1}{x-2} \quad (\text{A20c})$$

$$\binom{2y+1}{2y-2} \binom{2x-2y-1}{x-2y+2} = \binom{2y+1}{3} \binom{2x-2y-1}{x-3} \quad (\text{A20d})$$

Hence, for the series in Eq. (A19), each even entry corresponds to an odd entry over r .

Thus, Eq. (A19) can be written as,

$$\begin{aligned}
\binom{2x}{x} &= 2 \binom{2y+1}{1} \binom{2x-2y-1}{x-1} + 2 \binom{2y+1}{3} \binom{2x-2y-1}{x-3} + \dots \\
&\quad + 2 \binom{2y+1}{2y-1} \binom{2x-2y-1}{x-2y+1} + 2 \binom{2y+1}{2y+1} \binom{2x-2y-1}{x-2y-1} \quad (\text{A21})
\end{aligned}$$

Subsequently,

$$\binom{2y+1}{1} \binom{2x-2y-1}{x-1} + \binom{2y+1}{3} \binom{2x-2y-1}{x-3} + \dots$$

$$+ \binom{2y+1}{2y-1} \binom{2x-2y-1}{x-2y+1} + \binom{2y+1}{2y+1} \binom{2x-2y-1}{x-2y-1} = \frac{1}{2} \binom{2x}{x} \quad (\text{A22a})$$

$$\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y-1}{x-2r-1} = \frac{1}{2} \binom{2x}{x} \quad (\text{A22b})$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x > y$.

Eq. (A22b) is the Identity I.

We will now verify Identity II. Using binomial algebra, we can write:

$$\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y}{x-2r-1} = \sum_{r=0}^{y-1} \binom{2y}{2r+1} [z^{x-2r-1}] (1+z)^{2x-2y}$$

$$= [z^x] (1+z)^{2x-2y} \sum_{r=0}^{y-1} \binom{2y}{2r+1} z^{2r+1}$$

$$= [z^x] (1+z)^{2x-2y} \frac{1}{2} ((1+z)^{2y} - (1-z)^{2y}) \quad (\text{A23})$$

$$= \frac{1}{2} [z^x] ((1+z)^{2x} - (1+z)^{2x-2y} (1-z)^{2y})$$

$$= \frac{1}{2} [z^x] (1+z)^{2x} - \frac{1}{2} [z^x] ((1+z)^{2x-2y} (1-z)^{2y})$$

Eq. (A23) can be written as,

$$\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y}{x-2r-1} = \frac{1}{2} \binom{2x}{x} - \frac{1}{2} \sum_{r=0}^x (-1)^r \binom{2y}{r} \binom{2x-2y}{x-r} \quad (\text{A24})$$

Now we can consider the following relationship,

$$\sum_{r=0}^x (-1)^r \binom{2y}{r} \binom{2x-2y}{x-r} \neq 0 \quad (\text{A25})$$

where $x, y \in \mathbb{Z}^+$; $x \geq y$.

Thus,

$$\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y}{x-2r-1} \neq \frac{1}{2} \binom{2x}{x} \quad (\text{A26})$$

where $x, y \in \mathbb{Z}^+$; $r \in \mathbb{Z}_0^+$ and $x \geq y$.

Eq. (A26) is Identity II.

Appendix VII

Verification of the expression for odd order interactions (Alternative formulation of higher order Ising spin interaction)

We first numerically verify the transformation shown in Eq. (3.65), for all the possible spin configurations for odd order interaction up to 20. Figure A4(a) shows the number of possible spin configurations for different odd order interactions. Figure A4(b) shows the errors between the interactions obtained using the proposed formulation and the original interactions. For all the cases, the proposed formulation provides the exact spin interaction.

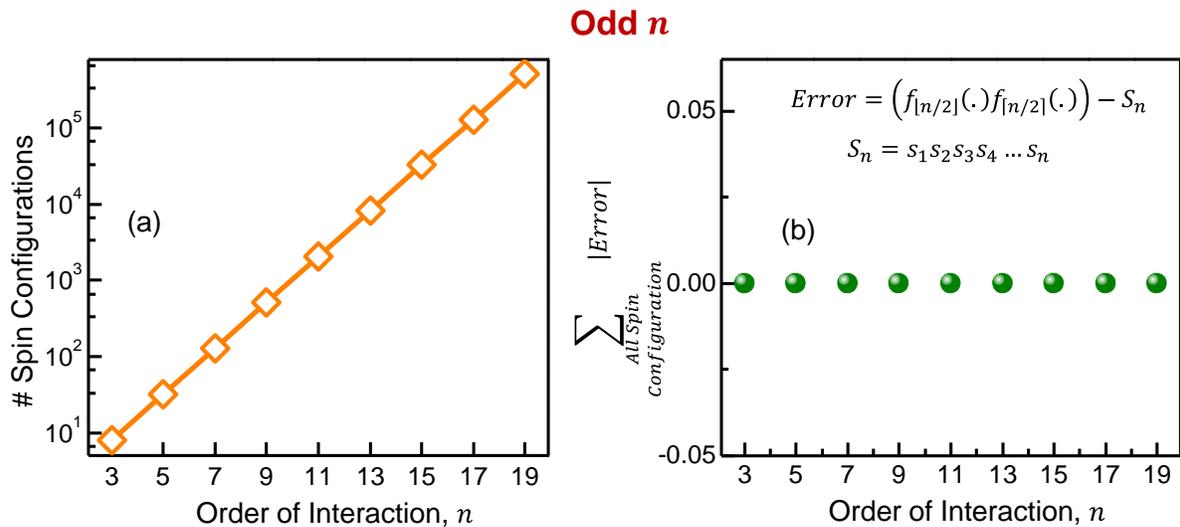


Figure A4. Numerical verification of the proposed formulation for odd order spin interactions (3-19). (a) Number of possible spin configurations. (b) Sum of absolute errors over all the possible spin configurations indicating exact match.

Now we will analytically verify the formulation in Eq. (3.65). As we discussed earlier, an Ising spin interaction term becomes minimum ($= -1$) when an odd number of spins in it are -1 . From the $2q + 1$ spins (s_1, \dots, s_{2q+1}) of S_{2q+1} , $\binom{2q+1}{q}$ number of q -order interaction

terms $(\prod_{\substack{T \subseteq \{1,2,\dots,n\} \\ |T|=q \\ s_i \in T}} s_i)$ and $\binom{2q+1}{q+1}$ number of $(q+1)$ -order interaction term $(\prod_{\substack{T \subseteq \{1,2,\dots,n\} \\ |T|=q+1 \\ s_i \in T}} s_i)$

can be made. We will prove that if $S_{2q+1} = -1$, sum of the q -order interaction terms will have the opposite sign of the sum of the $(q+1)$ -order interaction terms. In this way, at a minimum $f_q(\cdot)$ and $f_{q+1}(\cdot)$ will have opposite signs and thus $f_q(\cdot) \times f_{q+1}(\cdot)$ will map S_{2q+1} . In any other cases, they will have the same sign and consequently result 1. These 2 conditions can be proved by establishing the following identities:

Identity III:

$$\left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r-1} \right) \right] \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r} \right) \right] < 0 \quad (\text{A27})$$

where, $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x > y \geq r$.

Identity IV:

$$\left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y+1}{x-2r-1} \right) \right] \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x-2y+1}{x-2r} \right) \right] \geq 0 \quad (\text{A28})$$

where, $x, y \in \mathbb{Z}^+$; $r \in \mathbb{Z}_0^+$ and $x \geq y > r$.

To verify Identity III, using binomial series-based algebra we can write,

$$\begin{aligned}
\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r-1} &= \sum_{r=0}^y \binom{2y+1}{2r+1} [z^{x-2r-1}] (1+z)^{2x-2y} \\
&= \sum_{r=0}^y \binom{2y+1}{2r+1} [z^x] z^{2r+1} (1+z)^{2x-2y} \\
&= [z^x] (1+z)^{2x-2y} \sum_{r=0}^y \binom{2y+1}{2r+1} z^{2r+1} \\
&= [z^x] (1+z)^{2x-2y} \frac{1}{2} \{(1+z)^{2y+1} - (1-z)^{2y+1}\} \\
&= \frac{1}{2} [z^x] (1+z)^{2x+1} - \frac{1}{2} [z^x] (1+z)^{2x-2y} (1-z)^{2y+1} \\
&= \frac{1}{2} \binom{2x+1}{x} - \frac{1}{2} [z^x] (1+z)^{2x-2y} (1-z)^{2y+1}
\end{aligned} \tag{A29}$$

Similarly, we can write,

$$\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r} = \frac{1}{2} \binom{2x+1}{x+1} - \frac{1}{2} [z^{x+1}] (1+z)^{2x-2y} (1-z)^{2y+1} \tag{A30}$$

Thus,

$$\begin{aligned}
&\left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r-1} \right) \right] \\
&\quad \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r} \right) \right] \\
&= \frac{1}{2} [z^x] (1+z)^{2x-2y} (1-z)^{2y+1} \times \frac{1}{2} [z^{x+1}] (1+z)^{2x-2y} (1-z)^{2y+1} \\
&= \frac{1}{4} ([z^x] (1+z)^{2x-2y} (1-z)^{2y+1}) \times ([z^{x+1}] (1+z)^{2x-2y} (1-z)^{2y+1})
\end{aligned} \tag{A31}$$

Using combinatorics Eq. (A31) can be written as,

$$\begin{aligned}
& \left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r-1} \right) \right] \\
& \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r} \right) \right] \\
& = \frac{1}{4} \left(\sum_{r=0}^x (-1)^r \binom{2y+1}{r} \binom{2x-2y}{x-r} \right) \left(\sum_{r=0}^{x+1} (-1)^r \binom{2y+1}{r} \binom{2x-2y}{x+1-r} \right)
\end{aligned} \tag{A32}$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x > y$.

Now we can use the following relationship,

$$\left(\sum_{r=0}^x (-1)^r \binom{2y+1}{r} \binom{2x-2y}{x-r} \right) \left(\sum_{r=0}^{x+1} (-1)^r \binom{2y+1}{r} \binom{2x-2y}{x+1-r} \right) < 0 \tag{A33}$$

Thus, combining Eq. (A32) and Eq. (A33) we can write that,

$$\begin{aligned}
& \left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r-1} \right) \right] \\
& \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^y \binom{2y+1}{2r+1} \binom{2x-2y}{x-2r} \right) \right] < 0
\end{aligned} \tag{A34}$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x > y$.

Eq. (A34) is Identity III.

To verify Identity IV, using binomial series-based algebra we can write,

$$\begin{aligned}
\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r-1} &= \sum_{r=0}^{y-1} \binom{2y}{2r+1} [z^{x-2r-1}] (1+z)^{2x+1-2y} \\
&= \sum_{r=0}^{y-1} \binom{2y}{2r+1} [z^x] z^{2r+1} (1+z)^{2x+1-2y} \\
&= [z^x] (1+z)^{2x+1-2y} \sum_{r=0}^{y-1} \binom{2y}{2r+1} z^{2r+1} \\
&= [z^x] (1+z)^{2x+1-2y} \frac{1}{2} \{ (1+z)^{2y} - (1-z)^{2y} \} \\
&= \frac{1}{2} [z^x] (1+z)^{2x+1} - \frac{1}{2} [z^x] (1+z)^{2x+1-2y} (1-z)^{2y} \\
&= \frac{1}{2} \binom{2x+1}{x} - \frac{1}{2} [z^x] (1+z)^{2x+1-2y} (1-z)^{2y}
\end{aligned} \tag{A35}$$

Similarly, we can write,

$$\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r} = \frac{1}{2} \binom{2x+1}{x+1} - \frac{1}{2} [z^{x+1}] (1+z)^{2x+1-2y} (1-z)^{2y} \tag{A36}$$

Thus,

$$\begin{aligned}
&\left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r-1} \right) \right] \\
&\quad \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r} \right) \right] \\
&= \frac{1}{2} [z^x] (1+z)^{2x+1-2y} (1-z)^{2y} \times \frac{1}{2} [z^{x+1}] (1+z)^{2x+1-2y} (1-z)^{2y} \\
&= \frac{1}{4} ([z^x] (1+z)^{2x+1-2y} (1-z)^{2y}) \times ([z^{x+1}] (1+z)^{2x+1-2y} (1-z)^{2y})
\end{aligned} \tag{A37}$$

Using combinatorics Eq. (A37) can be written as,

$$\begin{aligned}
& \left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r-1} \right) \right] \\
& \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r} \right) \right] \\
& = \frac{1}{4} \left(\sum_{r=0}^x (-1)^r \binom{2y}{r} \binom{2x+1-2y}{x-r} \right) \left(\sum_{r=0}^{x+1} (-1)^r \binom{2y}{r} \binom{2x+1-2y}{x+1-r} \right)
\end{aligned} \tag{A38}$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x \geq y$.

Now we will use the following relationship,

$$\left(\sum_{r=0}^x (-1)^r \binom{2y}{r} \binom{2x+1-2y}{x-r} \right) \left(\sum_{r=0}^{x+1} (-1)^r \binom{2y}{r} \binom{2x+1-2y}{x+1-r} \right) \geq 0 \tag{A39}$$

Combining Eq. (A38) and Eq. (A39) we can write that,

$$\begin{aligned}
& \left[\frac{1}{2} \binom{2x+1}{x} - \left(\sum_{r=0}^{y-1} \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r-1} \right) \right] \\
& \times \left[\frac{1}{2} \binom{2x+1}{x+1} - \left(\sum_{r=0}^y \binom{2y}{2r+1} \binom{2x+1-2y}{x-2r} \right) \right] \geq 0
\end{aligned} \tag{A40}$$

where $x \in \mathbb{Z}^+$; $y, r \in \mathbb{Z}_0^+$ and $x \geq y$.

Eq. (A40) is Identity IV.

7. Publications & Patents

Journal Papers

- [J1] **Bashar, M. K.**, Lin, Z., & Shukla, N. Stability of Oscillator Ising Machines: Not All Solutions Are Created Equal. *Journal of Applied Physics* 134, 144901 (2023). [Link](#)
- [J2] **Bashar, M. K.**, & Shukla, N. Constructing Dynamical Systems to Model Higher Order Ising Spin Interactions and their Application in Solving Combinatorial Optimization Problems. *Scientific Reports* 13, 9558 (2023). [Link](#)
- [J3] **Bashar, M. K.**, Lin, Z., & Shukla, N. Oscillator-Inspired Dynamical Systems to Solve Boolean Satisfiability. *IEEE JxCDC* 9, 12-20 (2023). [Link](#)
- [J4] **Bashar, M. K.**, Mallick, A., Ghosh, A.W. & Shukla, N. Dynamical system-based computational models for solving combinatorial optimization on hypergraphs. *IEEE JxCDC* 9, 21-28 (2023). [Link](#)
- [J5] **Bashar, M. K.***, Mallick, A.* , Lin, Z., & Shukla, N. Synchronized Oscillator Inspired Computational Models to Solve Combinatorial Optimization Problems. *Physical Review Applied* 17 (2022). *Equal Contribution. [Link](#)
- [J6] **Bashar, M. K.**, Mallick A., and Shukla N. Experimental Investigation of the Dynamics of Coupled Oscillators as Ising Machines. *IEEE Access* 9, 148184 - 148190 (2021). [Link](#)
- [J7] **Bashar, M. K.**, Mallick, A., Truesdell, D. S. et al. Experimental Demonstration of a Reconfigurable Coupled Oscillator Platform to Solve the Max-Cut Problem. *IEEE JxCDC* 6, 116-121 (2020). [Link](#)
- [J8] **Bashar, M. K.**, Lin, Z., & Shukla, N. A note on analyzing the stability of oscillator Ising machines. *Electronics Letters* 59, e13054 (2023). [Link](#)
- [J9] Vaidya, J., **Bashar, M. K.**, & Shukla, N. Using noise to augment synchronization among oscillators. *Scientific Reports* 11, 1-8 (2021). [Link](#)

- [J10] Mallick, A., **Bashar, M. K.**, Truesdell, D. S. et al. Using synchronized oscillators to compute the maximum independent set. *Nature Communications* 11, 1-7 (2020). [Link](#)
- [J11] Mallick, A., Zhao, Z., **Bashar, M. K.** et al. CMOS-Compatible Ising Machines built using Bistable Latches Coupled through Ferroelectric Transistor Arrays. *Scientific Reports* 13, 1515 (2023). [Link](#)
- [J12] Ji, J., Ren, X., Gomez, J., **Bashar, M. K.** et al. Large-Scale Cardiac Muscle Cell-Based Coupled Oscillator Network for Vertex Coloring Problem. *Advanced Intelligent Systems* 5, 2200356 (2023). [Link](#)
- [J13] Ren, X., Gomez, J., **Bashar, M. K.**, Ji, J. et al. Cardiac Muscle Cell-Based Coupled Oscillator Network for Collective Computing. *Advanced Intelligent Systems* 3, 2000253 (2021). [Link](#)

Conference Proceedings

- [C1] **Bashar, M. K.***, Li, Z.*, Narayanan, V., Shukla, N. An FPGA-based Max-K-Cut Accelerator Exploiting Oscillator Synchronization Model. Accepted in *ISQED 24* (2024). *Equal Contribution
- [C2] **Bashar, M. K.** et al. Solving the maximum independent set problem using coupled relaxation oscillators. In *2019 Device Research Conference (DRC)*, 187-188 (IEEE,2019). [Link](#)
- [C3] **Bashar, M. K.**, Vaidya, J. et al. Ferroelectric-based Accelerators for Computationally Hard Problems. In *Proceedings of the 2021 on Great Lakes Symposium on VLSI (GLSVLSI)*, 485-489 (2021). Invited Paper. [Link](#)
- [C4] Mallick, A., **Bashar, M. K.** et al. Overcoming the Accuracy vs. Performance Trade-off in Oscillator Ising Machines. In *2021 IEEE International Electron Devices Meeting (IEDM)*, 40-2 (IEEE, 2021). [Link](#)

[C5] Mallick, A., **Bashar, M. K.** et al. Graph Coloring using Coupled Oscillator-based Dynamical Systems. In 2021 *IEEE International Symposium on Circuits and Systems (ISCAS)*, 1-5 (IEEE, 2021). [Link](#)

Patents

[P1] **Bashar, M. K.**, Mallick, A., & Shukla, N. S. U.S. Patent 11,436,280 (2022). [Link](#)

Bibliography

1. Moore, G. E. Cramming more components onto integrated circuits (1965).
2. Koomey, J., Berard, S., Sanchez, M., & Wong, H. Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing* **33**, 46-54 (2010).
3. Neumann, F., & Witt, C. Combinatorial optimization and computational complexity. In *Bioinspired computation in combinatorial optimization*, 9-19, Springer, Berlin, Heidelberg (2010).
4. Lemaréchal, C. Chapter vii nondifferentiable optimization. *Handbooks in operations research and management science* **1**, 529-572 (1989).
5. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* **104**, 148-175 (2015).
6. Tosuni, B. Graph Coloring Problems in Modern Computer Science. *European Journal of Interdisciplinary Studies* **1**, 87-95 (2015).
7. Fyrbiak, M., Wallat, S., Reinhard, S., Bissantz, N., & Paar, C. Graph similarity and its applications to hardware security. *IEEE Transactions on Computers* **69**, 505-519 (2019).
8. Oulasvirta, A. User interface design with combinatorial optimization. *Computer* **50**, 40-47 (2017).
9. Padmanabhan, B., & Tuzhilin, A. On the use of optimization for data mining: Theoretical interactions and eCRM opportunities. *Management Science* **49**, 1327-1343 (2003).
10. Greenberg, H. J., Hart, W. E., & Lancia, G. Opportunities for combinatorial optimization in computational biology. *INFORMS Journal on Computing* **16**, 211-231 (2004).
11. Fallas-Monge, J. J., Chavarría-Molina, J., & Alpízar-Brenes, G. Combinatorial metaheuristics applied to infectious disease models. In *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, pp. 1-6 (IEEE, 2016).
12. Siddhaye, S., Camarda, K., Southard, M., & Topp, E. Pharmaceutical product design using combinatorial optimization. *Computers & Chemical Engineering* **28**, 425-434 (2004).
13. Rhodes, N., Willett, P., Calvet, A., Dunbar, J. B., & Humblet, C. CLIP: similarity searching of 3D databases using clique detection. *Journal of chemical information and computer sciences* **43**, 443-448 (2003).
14. Kell, D. B. Scientific discovery as a combinatorial optimisation problem: how best to navigate the landscape of possible experiments?. *Bioessays* **34**, 236-244 (2012).
15. Boixo, S. et al. Characterizing quantum supremacy in near-term devices. *Nature Physics* **14**, 595-600 (2018).
16. IBM Quantum System One, Retrieved on Sep 21, 2021. At: <https://www.research.ibm.com/interactive/system-one/>
17. D-Wave Announces First Order For 2000Q Quantum Computer, Retrieved on Sep 21, 2021. At: <https://www.idquantique.com/d-wave-2000q-quantum-computer/>
18. Finnila, A. B., Gomez, M. A., Sebenik, C., Stenson, C., & Doll, J. D. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical physics letters* **219**, 343-348 (1994).

19. Boixo, S., Albash, T., Spedalieri, F. M., Chancellor, N., & Lidar, D. A. Experimental signature of programmable quantum annealing. *Nature communications* **4**, 1-8 (2013).
20. Patton, R., Schuman, C., & Potok, T. Efficiently embedding QUBO problems on adiabatic quantum computers. *Quantum Information Processing* **18** (2019).
21. Dorband, J. E. (2018). A method of finding a lower energy solution to a qubo/ising objective function. At: <https://arxiv.org/abs/1801.04849>.
22. T. Albash et al. Consistency tests of classical and quantum models for a quantum annealer. *Physical Review A* **91**, 042314 (2015).
23. Knill, E., & Laflamme, R. Theory of quantum error-correcting codes. *Physical Review A* **55**, 900 (1997).
24. MacLennan, B. J. The promise of analog computation. *International Journal of General Systems* **43**, 682-696 (2014).
25. Siegelmann, H. T., & Fishman, S. Analog computation with dynamical systems. *Physica D: Nonlinear Phenomena* **120**, 214-235 (1998).
26. Legenstein, R., & Maass, W. What makes a dynamical system computationally powerful. New directions in statistical signal processing: *From systems to brain*, 127-154 (2007).
27. Stepney, S. Nonclassical Computation-A Dynamical Systems Perspective. *Handbook of natural computing* **2** (2012).
28. Crutchfield, J. P., Ellison, C. J., James, R. G., & Mahoney, J. R. Synchronization and control in intrinsic and designed computation: An information-theoretic analysis of competing models of stochastic computation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **20**, 037105 (2010).
29. Fukami, S., Borders, W. A., Pervaiz, A. Z., Camsari, K. Y., Datta, S., & Ohno, H. Probabilistic computing based on spintronics technology. In *2020 IEEE Silicon Nanoelectronics Workshop (SNW)*, 21-22, (IEEE, 2020).
30. Camsari, K. Y., Sutton, B. M., & Datta, S. P-bits for probabilistic spin logic. *Applied Physics Reviews*, **6**, 011305 (2019).
31. Solli, D. R., & Jalali, B. Analog optical computing. *Nature Photonics* **9**, 704-706 (2015).
32. Wang, Z., Marandi, A., Wen, K., Byer, R. L., & Yamamoto, Y. Coherent Ising machine based on degenerate optical parametric oscillators. *Physical Review A* **88**, 063853 (2013).
33. Yao, P. et al. Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641-646 (2020).
34. Xia, Q., & Yang, J. J. Memristive crossbar arrays for brain-inspired computing. *Nature materials* **18**, 309-323 (2019).
35. Cai, F., Kumar, S., Van Vaerenbergh, T., Sheng, X., Liu, R., Li, C. et al. Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nature Electronics* **3**, 409-418 (2020).

36. Chen, J., Li, J., Li, Y., & Miao, X. Multiply accumulate operations in memristor crossbar arrays for analog computing. *Journal of Semiconductors* **42**, 013104 (2021).
37. Csaba, G. & Porod, W. Coupled oscillators for computing: A review and perspective. *Applied Physics Reviews* **7**, 011302 (2020).
38. Bertsimas, D., & Tsitsiklis, J. Simulated annealing. *Statistical science* **8**, 10-15 (1993).
39. Goto, H., Tatsumura, K., & Dixon, A. R. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. *Science advances* **5**, eaav2372 (2019).
40. Ferreira, A. M., García, J. A., López-Salas, J. G., & Vázquez, C. An efficient implementation of parallel simulated annealing algorithm in GPUs. *Journal of global optimization* **57**, 863-890 (2013).
41. Simulated Bifurcation Machine, retrieved on May 10, 2023. At: https://aws.amazon.com/marketplace/pp/prodview-f3hbaz4q3y32y?ref=_ptnr_web_en#pdp-reviews
42. Malakonakis, P., Smerdis, M., Sotiriades, E., & Dollas, A. An FPGA-based Sudoku Solver based on simulated annealing methods. In *2009 International Conference on Field-Programmable Technology*, 522-525 (IEEE, 2009).
43. Tatsumura, K., Dixon, A. R., & Goto, H. FPGA-based simulated bifurcation machine. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, 59-66 (IEEE, 2019).
44. Yamaoka, M., Okuyama, T., Hayashi, M., Yoshimura, C., & Takemoto, T. CMOS annealing machine: an in-memory computing accelerator to process combinatorial optimization problems. In *2019 IEEE Custom Integrated Circuits Conference (CICC)*, 1-8 (IEEE, 2019).
45. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., & Katzgraber, H. G. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics* **7**, 48 (2019).
46. Yamamoto, K., Ando, K., Mertig, N., Takemoto, T., Yamaoka, M., Teramoto, H. et al. 7.3 STATICA: A 512-spin 0.25 M-weight full-digital annealing processor with a near-memory all-spin-updates-at-once architecture for combinatorial optimization with complete spin-spin interactions. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, 138-140 (IEEE, 2020).
47. Hamerly, R., Inagaki, T., McMahon, P. L., Venturelli, D., Marandi, A., Onodera, T. et al. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Science advances* **5**, eaau0823 (2019).
48. Takata, K., Marandi, A., Hamerly, R., Haribara, Y., Maruo, D., Tamate, S. et al. A 16-bit coherent Ising machine for one-dimensional ring and cubic graph problems. *Scientific reports* **6**, 1-7 (2016).
49. Yin, X., Sedighi, B., Varga, M., Ercsey-Ravasz, M., Toroczkai, Z., & Hu, X. S. Efficient analog circuits for boolean satisfiability. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **26**, 155-167 (2017).

50. Mallick, A., Bashar, M. K., Truesdell, D. S., Calhoun, B. H., & Shukla, N. Overcoming the accuracy vs. performance trade-off in oscillator using machines. In *2021 IEEE International Electron Devices Meeting (IEDM)*, 40-2 (IEEE, 2021).
51. Bashar, M. K., Mallick, A., Truesdell, D. S., Calhoun, B. H., Joshi, S., & Shukla, N. Experimental Demonstration of a Reconfigurable Coupled Oscillator Platform to Solve the Max-Cut Problem. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* **6**, 116-121 (2020).
52. Camsari, K. Y., Debashis, P., Ostwal, V., Pervaiz, A. Z., Shen, T., Chen, Z. et al. From charge to spin and spin to charge: Stochastic magnets for probabilistic switching. *Proceedings of the IEEE*, **108**, 1322-1337 (2020).
53. Takemoto, T., Hayashi, M., Yoshimura, C., & Yamaoka, M. A 2x30k-Spin Multi-Chip Scalable CMOS Annealing Processor Based on a Processing-in-Memory Approach for Solving Large-Scale Combinatorial Optimization Problems. *IEEE Journal of Solid-State Circuits* **55**, 145-156 (2019).
54. Ji, J., Ren, X., Gomez, J., Bashar, M. K., Shukla, N., Datta, S., & Zorlutuna, P. Large-Scale Cardiac Muscle Cell-Based Coupled Oscillator Network for Vertex Coloring Problem. *Advanced Intelligent Systems* **5**, 2200356 (2023).
55. Goto, E. The parametron, a digital computing element which utilizes parametric oscillation. *Proceedings of the IRE* **47.8**, 1304-1316 (1959).
56. Strogatz, S. H. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena* **143**, 1-20 (2000).
57. Rodrigues, F. A., Peron, T. K. D., Ji, P., & Kurths, J. The Kuramoto model in complex networks. *Physics Reports* **610**, 1-98 (2016).
58. Acebrón, J. A., Bonilla, L. L., Vicente, C. J. P., Ritort, F., & Spigler, R. The Kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of modern physics* **77**, 137 (2005).
59. Wu, C. W. Graph coloring via synchronization of coupled oscillators. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **45**, 974-978 (1998).
60. Lee, S. A. k-Phase oscillator synchronization for graph coloring. *Mathematics in Computer Science* **3**, 61-72 (2010).
61. Wu, J., Jiao, L., Li, R., & Chen, W. Clustering dynamics of nonlinear oscillator network: Application to graph coloring problem. *Physica D: Nonlinear Phenomena* **240**, 1972-1978 (2011).
62. Parihar, A., Shukla, N., Jerry, M., Datta, S., & Raychowdhury, A. Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Scientific reports* **7**, 911 (2017).
63. Bashar, M. K., Hrды, R., Mallick, A., Hassanzadeh, F. F., & Shukla, N. Solving the maximum independent set problem using coupled relaxation oscillators. In *2019 Device Research Conference (DRC)*, 187-188 (IEEE, 2019).
64. Mallick, A., Bashar, M. K., Truesdell, D. S., Calhoun, B. H., Joshi, S., & Shukla, N. Graph coloring using coupled oscillator-based dynamical systems. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1-5 (IEEE, 2021).

65. Mallick, A., Bashar, M. K., Truesdell, D. S., Calhoun, B. H., Joshi, S., & Shukla, N. Using synchronized oscillators to compute the maximum independent set. *Nature communications* **11**, 1-7 (2020).
66. Wang, T. and Roychowdhury, J. OIM: Oscillator-based Ising machines for solving combinatorial optimisation problems. in *International Conference on Unconventional Computation and Natural Computation*, 232-256. Springer, Cham (2019).
67. Wang, T., Wu, L., Nobel, P., & Roychowdhury, J. Solving combinatorial optimisation problems using oscillator based Ising machines. *Natural Computing* **20**, 287-306 (2021).
68. Bhansali, P., & Roychowdhury, J. Gen-Adler: The generalized Adler's equation for injection locking analysis in oscillators. In *2009 Asia and South Pacific Design Automation Conference*, 522-527 (IEEE, 2009).
69. Bashar, M. K., Mallick, A., & Shukla, N. Experimental investigation of the dynamics of coupled oscillators as ising machines. *IEEE Access* **9**, 148184-148190 (2021).
70. Wang, T., Wu, L., & Roychowdhury, J. New computational results and hardware prototypes for oscillator-based ising machines. In *Proceedings of the 56th Annual Design Automation Conference 2019*, 1-2 (2019).
71. Chou, J., Bramhavar, S., Ghosh, S., & Herzog, W. Analog coupled oscillator based weighted Ising machine. *Scientific reports* **9**, 14786 (2019).
72. Dutta, S., Khanna, A., Assoa, A. S., Paik, H., Schlom, D. G., Toroczkai, Z. et al. An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators. *Nature Electronics* **4**, 502-512 (2021).
73. Ahmed, I. Chiu, P. W., and Kim, c. H. A probabilistic self-annealing compute fabric based on 560 hexagonally coupled ring oscillators for solving combinatorial optimization problems. In *2020 IEEE Symposium on VLSI Circuits*, 1-2 (IEEE, 2020).
74. Lo, H., Moy, W., Yu, H., Sapatnekar, S., & Kim, C. H. An Ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture. *Nature Electronics* **6**, 771-778 (2023).
75. McGoldrick, B. C., Sun, J. Z., & Liu, L. Ising machine based on electrically coupled spin Hall nano-oscillators. *Physical Review Applied* **17**, 014006 (2022).
76. Albertsson, D. I. et al. Ultrafast Ising Machines using spin torque nano-oscillators. *Applied Physics Letters* **118**.11 (2021).
77. Kanao, T., & Goto, H. High-accuracy Ising machine using Kerr-nonlinear parametric oscillators with local four-body interactions. *npj Quantum Information* **7**, 18 (2021).
78. Lucas, A. Ising formulations of many NP problems. *Frontiers in physics* **2**, 5 (2014).
79. Glover, F., Kochenberger, G., & Du, Y. (2018). A tutorial on formulating and using QUBO models. At <https://arxiv.org/abs/1811.11538>
80. Mandal, Avradip, et al. Compressed quadratization of higher order binary optimization problems. *Proceedings of the 17th ACM International Conference on Computing Frontiers* (2020).

81. Wu, F. Y. The potts model. *Reviews of modern physics* **54**, 235 (1982).
82. Mallick, A., Bashar, M. K., Lin, Z., & Shukla, N. Computational models based on synchronized oscillators for solving combinatorial optimization problems. *Physical Review Applied* **17**, 064064 (2022).
83. Bashar, M. K., Li, Z., Narayanan, V., Shukla, N. An FPGA-based Max-K-Cut Accelerator Exploiting Oscillator Synchronization Model. Accepted in *ISQED 2024*.
84. Szabados, T. An elementary introduction to the Wiener process and stochastic integrals. arXiv preprint arXiv:1008.1510 (2010).
85. Wang, T., and Roychowdhury, J. Oscillator-based Ising machine. arXiv preprint arXiv:1709.08102 (2017).
86. Bashar, M. K. et al. Dynamical System-based Computational Models for Solving Combinatorial Optimization on Hypergraphs. *IEEE JxCDC* **9**, 21-28 (2023).
87. Trapezoidal rule (differential equations), Retrieved at May, 10, 2023. At: [https://en.wikipedia.org/wiki/Trapezoidal_rule_\(differential_equations\)](https://en.wikipedia.org/wiki/Trapezoidal_rule_(differential_equations))
88. Lee, D-U. et al. A hardware Gaussian noise generator using the Box-Muller method and its error analysis. *IEEE Trans. Comput.*, **55**, 659-671 (2006).
89. Yavorsky, A. et al. Highly parallel algorithm for the Ising ground state searching problem. arXiv preprint arXiv:1907.05124 (2019).
90. Mousavi, S. F. A GPU-Accelerated Population-Based Boltzmann Machine for Solving Combinatorial Optimization Problems,” Diss. University of Toronto (Canada) (2020).
91. Zou, Y., and Lin, M. Massively simulating adiabatic bifurcations with FPGA to solve combinatorial optimization. In *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 65-75 (2020).
92. Gyoten, H., Hiromoto, M., and Sato, T. Area efficient annealing processor for ising model without random number generator. *IEICE TRANSACTIONS on Information and Systems* **101**, 314-323 (2018).
93. Benchmarking the MAX-CUT problem on the Simulated Bifurcation Machine, retrieved on May 10, 2023. At: <https://medium.com/toshiba-sbm/benchmarking-the-max-cut-problem-on-the-simulated-bifurcation-machine-e26e1127c0b0>
94. Ercsey-Ravasz, M. and Toroczkai, Z. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nat. Phys.* **7**, 966–970 (2011).
95. B. Molnár, and M. Ercsey-Ravasz, Asymmetric continuous-time neural networks without local traps for solving constraint satisfaction problems, *PLoS one* **8**, e73400 (2013).
96. Grimaldi, A., Sánchez-Tejerina, L., Aadit, N.A., Chiappini, S., Carpentieri, M., Camsari, K., and Finocchio, G. Spintronics-compatible Approach to Solving Maximum-Satisfiability Problems with Probabilistic Computing, Invertible Logic, and Parallel Tempering, *Physical Review Applied* **17**, 024052 (2022).

97. Reformulating a Problem, retrieved on June 17, 2022. Available at: https://docs.dwavesys.com/docs/latest/handbook_reformulating.html#:~:text=The%20D%2DWave%20Problem%2DSolving,2i%20with%20xi
98. Gabor, T. Zielinski, S., Feld, S., Roch, C., Seidel, C., Neukart, F., Galter, I., Mauerer, W., and Linnhoff-Popien, C. Assessing solution quality of 3SAT on a quantum annealing platform. In *Proceedings of the International Workshop on Quantum Technology and Optimization Problems*, 23 (2019).
99. Sanyal, S., Reduction from SAT to 3SAT, Retrieved on June 17, 2022, Available at <https://cse.iitkgp.ac.in/~palash/2018AlgoDesignAnalysis/SAT-3SAT.pdf>
100. Bashar, M. K., Lin, Z., & Shukla, N. Oscillator-Inspired Dynamical Systems to Solve Boolean Satisfiability. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* **9**, 12 - 20 (2023).
101. Borders, W. A., Pervaiz, A. Z., Fukami, S., Camsari, K. Y., Ohno, H., & Datta, S. Integer factorization using stochastic magnetic tunnel junctions. *Nature* **573**, 390-393 (2019).
102. Tinhofer, G. Graph isomorphism and theorems of Birkhoff type. *Computing* **36**, 285 (1986).
103. Neogy, A. and Roychowdhury, J. Analysis and design of sub-harmonically injection locked oscillators. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 1209-1214 (2012).
104. Butcher J.C. A history of Runge-Kutta methods. *Applied Numerical Mathematics* **20**, 247-260 (1996).
105. Bashar, M. K., & Shukla, N. Constructing Dynamical Systems to Model Higher Order Ising Spin Interactions and their Application in Solving Combinatorial Optimization Problems. *Scientific Reports* **13**, 9558 (2023)
106. Rodriguez Heck, E. Linear and quadratic reformulations of nonlinear optimization problems in binary variables (2018).
107. Ding, J., Sly, A. and Sun, N. Satisfiability threshold for random regular NAE-SAT. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 814-822 (2014).
108. Ageev, A. A. and Sviridenko, M. I. An approximation algorithm for hypergraph max k-cut with given sizes of parts. In *European Symposium on Algorithms*, 32-41, Springer, Berlin, Heidelberg (2000).
109. Bashar, M. K., Lin, Z., & Shukla, N. Stability of Oscillator Ising Machines: Not All Solutions Are Created Equal. *Journal of Applied Physics* **134**, 144901 (2023).
110. Mohseni, N., McMahon, P. L., & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics* **4**, 363-379 (2022).
111. Erementchouk, M., Shukla, A., & Mazumder, P. On computational capabilities of Ising machines based on nonlinear oscillators. *Physica D: Nonlinear Phenomena* **437**, 133334 (2022).
112. Böhm, F., Vaerenbergh, T. V., Verschaffelt, G., & Van der Sande, G. Order-of-magnitude differences in computational performance of analog Ising machines induced by the choice of nonlinearity. *Communications Physics* **4**, 149 (2021).

113. Frady, E. P., & Sommer, F. T. Robust computation with rhythmic spike patterns. *Proceedings of the National Academy of Sciences* **116**, 18050-18059 (2019).
114. Gu, Q. L. L., Tian, Z. Q. K., Kovačič, G., Zhou, D., & Cai, D. The dynamics of balanced spiking neuronal networks under Poisson drive is not chaotic. *Frontiers in Computational Neuroscience* **12**, 47 (2018).
115. Nobukawa, S., Nishimura, H., Yamanishi, T., & Liu, J. Q. Analysis of chaotic resonance in Izhikevich neuron model. *PloS one* **10**, e0138919 (2015).
116. Nobukawa, S., Nishimura, H., & Yamanishi, T. Routes to chaos induced by a discontinuous resetting process in a hybrid spiking neuron model. *Scientific reports* **8**, 379 (2018).
117. Sándor, B., Schneider, B., Lázár, Z. I., & Ercsey-Ravasz, M. A novel measure inspired by Lyapunov exponents for the characterization of dynamics in state-transition networks. *Entropy* **23**, 103 (2021).
118. Arrowsmith, D., & Place, C. M. Dynamical systems: differential equations, maps, and chaotic behaviour (Vol. 5). *CRC Press* (1992).
119. Bashar, M. K., Lin, Z., & Shukla, N. A note on analyzing the stability of oscillator Ising machines. *Electronics Letters* **59**, e13054 (2023).
120. Fischer, A., & Igel, C. An introduction to restricted Boltzmann machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 17th Iberoamerican Congress, CIARP 2012*, proceedings 17 (pp. 14-36). Springer Berlin Heidelberg (2012).
121. Böhm, F., Alonso-Urquijo, D., Verschaffelt, G., & Van der Sande, G. Noise-injected analog Ising machines enable ultrafast statistical sampling and machine learning. *Nature Communications* **13**, 5847 (2022).
122. Böhm, F., Inagaki, T., Inaba, K., Honjo, T., Enbutsu, K., Umeki, T. et al. Understanding dynamics of coherent Ising machines through simulation of large-scale 2D Ising models. *Nature communications* **9**, 5020 (2018).
123. Chib, S., & Greenberg, E. Understanding the metropolis-hastings algorithm. *The american statistician* **49**, 327-335 (1995).
124. Park, H., Lee, S., Kim, S., Park, J., Jeong, J., Kim, K. M., ... & Kim, H. J. Metropolis-hastings data augmentation for graph neural networks. *Advances in Neural Information Processing Systems* **34**, 19010-19020 (2021).
125. Vandermonde's identity, Retrieved on January 4, 2024.
At: https://en.wikipedia.org/wiki/Vandermonde%27s_identity