



OIM: Oscillator-Based Ising Machines for Solving Combinatorial Optimisation Problems

Tianshi Wang^(✉) and Jaijeet Roychowdhury

Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, USA
{tianshi, jr}@berkeley.edu

Abstract. We present a new way to make Ising machines, *i.e.*, using networks of coupled self-sustaining nonlinear oscillators. Our scheme is theoretically rooted in a novel result that establishes that the phase dynamics of coupled oscillator systems, under the influence of subharmonic injection locking, are governed by a Lyapunov function that is closely related to the Ising Hamiltonian of the coupling graph. As a result, the dynamics of such oscillator networks evolve naturally to local minima of the Lyapunov function. Two simple additional steps (*i.e.*, adding noise, and turning subharmonic locking on and off smoothly) enable the network to find excellent solutions of Ising problems. We demonstrate our method on Ising versions of the MAX-CUT and graph colouring problems, showing that it improves on previously published results on several problems in the G benchmark set. Our scheme, which is amenable to realisation using many kinds of oscillators from different physical domains, is particularly well suited for CMOS IC implementation, offering significant practical advantages over previous techniques for making Ising machines. We present working hardware prototypes using CMOS electronic oscillators.

1 Introduction

The Ising model [1, 2] takes any weighted graph and uses it to define a scalar function called the Ising Hamiltonian. Each vertex in the graph is associated with a *spin*, *i.e.*, a binary variable taking values ± 1 . The Ising problem is to find an assignment of spins that minimises the Ising Hamiltonian (which depends on the spins and on the graph's weights). Solving the Ising problem in general has been shown to be very difficult [3], but devices that can solve it quickly using specialised hardware have been proposed in recent years [4–9]. Such Ising machines have attracted much interest because many classically difficult combinatorial optimisation problems (including all 21 of Karp's well-known list of NP-complete problems [10]) can be mapped to Ising problems [11]. Hence, as Moore's Law nears its limits, Ising machines offer promise as a novel alternative paradigm for solving difficult computational problems efficiently.

We present a new and attractive means for realising Ising machines, *i.e.*, using networks of coupled, self-sustaining nonlinear oscillators. We first establish a key

theoretical result that relates the (continuous) phase dynamics of an oscillator network with the (discrete/combinatorial) Ising Hamiltonian of the graph representing the oscillator couplings [12]. We then build on this result to develop practical oscillator-based Ising machines and demonstrate their effectiveness by solving MAX-CUT and graph colouring combinatorial optimisation problems [13, 14]. We also present working hardware prototypes of our oscillator-based Ising machines.

We first show that the phase dynamics of any network of coupled, self-sustaining, amplitude-stable oscillators can be abstracted using the Generalised Adler model [15, 16], a generalisation of the well-known Kuramoto model [17–19]. The model’s phase dynamics are governed by an associated Lyapunov function, *i.e.*, a scalar function of the oscillators’ phases that is always non-increasing and settles to stable local minima as phase dynamics evolve. If each oscillator’s phase settles to either 0 or π (radians) and these values are associated with spins of ± 1 , we show that this Lyapunov function is essentially identical to the Ising Hamiltonian of the oscillator network’s connectivity graph. In general, however, oscillator phases do not settle to the discrete values $0/\pi$, but span a continuum of values instead. In order to binarise oscillator phases (*i.e.*, get them to settle to values near $0/\pi$), we inject each oscillator with a second harmonic signal (dubbed SYNC) that induces subharmonic injection locking (SHIL), which makes the phase of each oscillator settle to a value near either 0 or π [15, 20–22]. We devise a new Lyapunov function that governs the network’s dynamics with SHIL, then show its equivalence to the Ising Hamiltonian at phase values of $0/\pi$.

Thus we show that when SHIL binarisation is applied, coupled oscillator network dynamics settle naturally to *local* minima of a continised version of the associated Ising Hamiltonian. To evolve the system out of local minima towards the global minimum, we show that a simple scheme, in which the binarising second-harmonic SYNC signal’s amplitude is ramped up and down together with judicious amounts of noise added, works well. We present simulation results on a standard MAX-CUT benchmark set of 54 large problems, demonstrating not only that it finds the best-known results in many cases, but finds **better results than seem to have been previously published for 17 of the 54 problems**. We also demonstrate our method on the graph-colouring problem and present small (up to 32 CMOS oscillators) prototypes built on breadboard that function perfectly, testifying to the ease with which practical hardware implementations can be built.

Our scheme is different from previous Ising machine approaches, which are of 3 types (see Sect. 2): (1) a fibre-optic laser-based scheme known as the Coherent Ising Machine [4–6], (2) the D-WAVE quantum Ising machine [7, 8] and (3) CMOS hardware accelerated simulated annealing chips for solving Ising problems [9, 23–25]. Unlike CIM and D-WAVE, which are large, expensive and ill-suited to low-cost mass production, our approach is a purely classical scheme that does not rely on quantum phenomena or novel nano-devices. Indeed, it can be implemented using conventional CMOS electronics, which has many advantages: scalability/miniaturisability (*i.e.*, very large numbers of spins in a physically small system), well-established design processes and tools that essentially guarantee

first-time working hardware, very low power operation, seamless integration with control and I/O logic, easy programmability via standard interfaces like USB, and low cost mass production. CMOS implementations of our scheme also allow complete flexibility in introducing controlled noise and programming SYNC ramping schedules. Furthermore, implementing oscillator coupling by physical connectivity makes our scheme inherently parallel, unlike CIM, where coupling is implemented via FPGA-based digital computation and is inherently serial. The advantages of CMOS also apply, of course, to hardware simulated annealing engines [9, 23–25], but our scheme has additional attractive features. One key advantage relates to variability, a significant problem in nanoscale CMOS. For oscillator networks, device- and circuit-level variability impacts the system by causing a spread in the natural frequencies of the oscillators. Unlike other schemes, where performance deteriorates due to variability [9], we can essentially eliminate variability by means of simple VCO-based calibration to bring all the oscillators to the same frequency.¹ Another key potential advantage stems from the continuous/analog nature of our scheme (as opposed to purely digital simulated annealing schemes). Computational experiments indicate that the time our scheme takes to find good solutions of the Ising problem grows only very slowly with respect to the number of spins. This is a significant potential advantage over simulated annealing schemes [23] as hardware sizes scale up to large numbers of spins. Note that we can use virtually any type of nonlinear oscillator (not just CMOS) to implement our scheme, including optical, MEMS, biochemical, spin-based, *etc.*, oscillators; however, CMOS seems the easiest and most advantageous implementation route given the current state of technology.

In the remainder of this paper, we first provide a brief summary of the Ising problem and existing Ising machine schemes in Sect. 2. We then present our oscillator-based Ising machine scheme (dubbed OIM, for Oscillator Ising Machine) in Sect. 3, explaining the theory that enables it to work. Then in Sect. 4, we present both computational and hardware examples showing the effectiveness of our scheme for solving several combinatorial optimisation problems.

2 The Ising Problem and Existing Ising Machine Approaches

The Ising model is named after the German physicist Ernest Ising. It was first studied in the 1920s as a mathematical model for explaining domain formation in ferromagnets [1]. It comprises a group of discrete variables $\{s_i\}$, *aka* spins, each taking a binary value ± 1 , such that an associated “energy function”, known as the Ising Hamiltonian, is minimised:

$$\min H \triangleq - \sum_{1 \leq i < j \leq n} J_{ij} s_i s_j - \sum_{i=1}^n h_i s_i, \quad \text{such that } s_i \in \{-1, +1\}, \quad (1)$$

where n is the number of spins; $\{J_{ij}\}$ and $\{h_i\}$ are real coefficients.

¹ Moreover, as we show in Sect. 3.4, our scheme is inherently resistant to variability even without such calibration.

The Ising model is often simplified by dropping the $\{h_i\}$ terms. Under this simplification, the Ising Hamiltonian becomes

$$H = - \sum_{i,j, i < j} J_{ij} s_i s_j. \quad (2)$$

What makes the Ising model particularly interesting is that many hard optimisation problems can be shown to be equivalent to it [26]. In fact, all of Karp's 21 NP-complete problems can be mapped to it by assigning appropriate values to the coefficients [11]. Physical systems that can directly minimise the Ising Hamiltonian, namely Ising machines, thus become very attractive for potentially outperforming conventional algorithms run on CPUs for these problems.

Several schemes have been proposed recently for realising Ising machines in hardware. One well-known example is from D-Wave Systems [7, 8]. Their quantum Ising machines use superconducting loops as spins and connect them using Josephson junction devices [27]. As the machines require a temperature below 80 mK (-273.07°C) to operate [7], they all have a large footprint to accommodate the necessary cooling system. While many question their advantages over simulated annealing run on classical computers [28], proponents believe that through a mechanism known as quantum tunnelling, they can offer large speedups on problems with certain energy landscapes [29].

Other proposals use novel non-quantum devices as Ising spins instead, so that the machines can function at room temperature. Most notable among them is a scheme based on lasers and kilometre long optical fibres [4–6]. The Ising spins are represented using time-multiplexed optical parametric oscillators (OPOs), which are laser pulses travelling on the same fibre. The coupling between these pulses is implemented digitally by measurement and feedback using an FPGA. While these machines can potentially be more compact than D-Wave's machines, it is unclear how they can be miniaturised and integrated due to the use of long fibres. Recent studies have also proposed the use of several novel nanodevices as Ising spins, including MEMS (Micro-Electro-Mechanical Systems) resonators [30] and nanomagnets from Spintronics [31]. Physical realisation of these machines still awaits future development of these emerging device technologies.

Another broad direction is to build Ising model emulators using digital circuits. A recent implementation [9] uses CMOS SRAM cells as spins, and couples them using digital logic gates. The authors point out, however, that “the efficacy in achieving a global energy minimum is limited” [9] due to variability. The speed-up and accuracy reported by [9] are instead based on deterministic on-chip computation paired with an external random number generator—a digital hardware implementation of the simulated annealing algorithm. More recently, similar digital accelerators have also been tried on FPGAs [32]. These implementations are not directly comparable to the other Ising machine implementations discussed above, which attempt to use interesting intrinsic physics to minimise the Ising Hamiltonian for achieving large speedups.

3 Oscillator-Based Ising Machines

In this section, we show that a network of coupled self-sustaining oscillators can function as an Ising machine. To do so, we first study the response of a single oscillator under injection locking in Sect. 3.1. Specifically, we examine the way the oscillator’s phase locks to that of the external input. While regular injection locking typically aligns the oscillator’s phase with the input, as illustrated in Fig. 1(a) and (b), its variant—subharmonic injection locking (SHIL)—can make the oscillator develop multiple stable phase-locked states (Fig. 1(c) and (d)). As we show in Sect. 3.1, these phenomena can be predicted accurately using the Gen-Adler model [16].

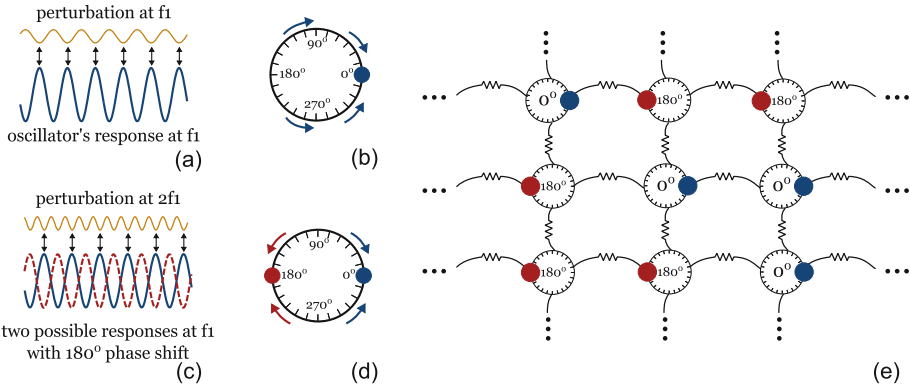


Fig. 1. Illustration of the basic mechanism of oscillator-based Ising machines: (a) oscillator shifts its natural frequency from f_0 to f_1 under external perturbation; (b) oscillator’s phase becomes stably locked to the perturbation; (c) when the perturbation is at $2f_1$, the oscillator locks to its subharmonic at f_1 ; (d) bistable phase locks under subharmonic injection locking; (e) coupled subharmonically injection-locked oscillators settle with binary phases representing an optimal spin configuration for an Ising problem.

The Gen-Adler equation of a single oscillator, when extended to the phase dynamics of coupled oscillator networks, becomes equivalent to a variant of the Kuramoto model. In Sect. 3.2, we show that the model’s dynamics are governed by a global Lyapunov function, a scalar “energy like” quantity that is naturally minimised by the coupled oscillator network. Then in Sect. 3.3, we introduce SHIL into the system to binarise the phases of oscillators. As illustrated in Fig. 1(e), SHIL induces each oscillator to settle to one of two stable phase-locked states. Due to the coupling between them, a network of such binarised oscillators will prefer certain phase configurations over others. We confirm this intuition in Sect. 3.3 by deriving a new Lyapunov function that such a system (*i.e.*, with SHIL) minimises. By examining this function’s equivalence to the Ising Hamiltonian, we show that such a coupled oscillator network under SHIL indeed physically implements an Ising machine. Finally, in Sect. 3.4, we consider

the effect of variability on the system's operation. We show that a spread in the natural frequencies of the oscillators contributes a linear term in the global Lyapunov function, and does affect Ising machine performance by much if the variability is not extreme.

3.1 Injection Locking in Oscillators

When an oscillator with a natural frequency ω_0 is perturbed by a small periodic input at a similar frequency ω_1 , its phase response can be predicted well using the Generalised Adler's model (Gen-Adler) [16]. Gen-Adler has the following form:

$$\frac{d}{dt}\phi(t) = \omega_0 - \omega_1 + \omega_0 \cdot c(\phi(t) - \phi_{in}), \quad (3)$$

where $\phi(t)$ and ϕ_{in} are the phases of the oscillator and the perturbation. $c(\cdot)$ is a 2π -periodic function derived based on an intrinsic quantity of the oscillator known as the Phase Response Curve (PRC) [33] or the Perturbation Projection Vector (PPV) [34]. A detailed derivation of Gen-Adler from the low-level differential equations of an oscillator is provided in a preprint of this paper [35].

The Gen-Adler equation governs the dynamics of the oscillator's phase under periodic inputs; its equilibrium states can be used to accurately predict the injection-locked states of the oscillator. The equilibrium Gen-Adler equation can be derived by rearranging (3):

$$\frac{\omega_1 - \omega_0}{\omega_0} = c(\phi^* - \phi_{in}). \quad (4)$$

where ϕ^* is the solution of phase in equilibrium.

The Left Hand Side (LHS) of (4) is a constant representing the frequency detuning of the oscillator from the input; the Right Hand Side (RHS) is a periodic function of ϕ^* ; its magnitude depends on both the PPV of the oscillator and the strength of the input [16]. By plotting both terms and looking for intersections, one can easily predict whether injection locking will occur, and if it does, what the locked phase of the oscillator ϕ^* will be. Figure 2(a) plots a few examples of LHS/RHS, showing their shapes and magnitudes under different conditions.

As mentioned in Sect. 1, SHIL can occur when the external input is about twice as fast as the oscillator. When the input is at frequency $2\omega_1$, it can be shown that the corresponding $c(\cdot)$ becomes a π -periodic function [15, 36]; a typical example is given in Fig. 2(b), where $c(\cdot)$ takes the shape of $-\sin(2\phi)$. In this case, two of the four LHS-RHS intersections represent stable phase-locked states; it can be shown that they are separated by a phase difference of 180° [15]. Gen-Adler is a powerful technique for predicting and understanding injection locking in oscillators and constitutes an important foundation for the analyses that follow.

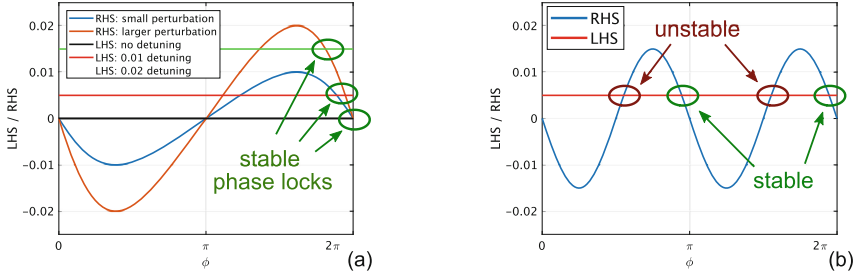


Fig. 2. Illustration of the LHS and RHS of the equilibrium Gen-Adler equation. (a) Under normal injection locking, the intersection of LHS and RHS predicts the only solution of ϕ under different scenarios. (b) Perturbation at $2\omega_1$ changes the shape of $c(\cdot)$ in Gen-Adler; the intersections now predict the locations of two stable phase-locked states.

3.2 Global Lyapunov Function

For an oscillator in a coupled oscillator network, its external perturbations come from the other oscillators connected to it. Its Gen-Adler equation can be written as

$$\frac{d}{dt}\phi_i(t) = \omega_i - \omega^* + \omega_i \cdot \sum_{j=1, j \neq i}^n c_{ij}(\phi_i(t) - \phi_j(t)), \quad (5)$$

where $\{\phi_i\}$ represents the phases of n oscillators; ω_i is the frequency of the oscillator whereas ω^* is the central frequency of the network. $c_{ij}(\cdot)$ is a 2π -periodic function for the coupling between oscillator i and oscillator j .

To simplify exposition, we now assume that the c_{ij} functions are sinusoidal, although in [35], we show that this does not have to be the case for the analysis to hold true.² We further assume zero spread in the natural frequencies of oscillators, *i.e.*, $\omega_i \equiv \omega^*$, and discuss the effect of frequency variability later in Sect. 3.4. With these simplifications, (5) can be written as

$$\frac{d}{dt}\phi_i(t) = -K \cdot \sum_{j=1, j \neq i}^n J_{ij} \cdot \sin(\phi_i(t) - \phi_j(t)). \quad (6)$$

Here, we are using the coefficients $\{J_{ij}\}$ ³ from the Ising model (1) to set the connectivity of the network, *i.e.*, the coupling strength between oscillators i and j is proportional to J_{ij} . The parameter K modulates the overall coupling strength of the network.

There is a global Lyapunov function associated with (6) [37]:

$$E(\vec{\phi}(t)) = -K \cdot \sum_{i,j, i \neq j} J_{ij} \cdot \cos(\phi_i(t) - \phi_j(t)), \quad (7)$$

² More generally, c_{ij} s can be any 2π -periodic odd functions, which are better suited to practical oscillators.

³ In the Ising Hamiltonian (1), J_{ij} is only defined when $i < j$; here we assume that $J_{ij} = J_{ji}$ for all i, j .

where $\vec{\phi}(t) = [\phi_1(t), \dots, \phi_n(t)]^T$. Being a global Lyapunov function, it is an objective function the coupled oscillator system always tends to minimise as it evolves over time [38].

If we look at the values of this continuous function $E(\vec{\phi}(t))$ at some discrete points, we notice that it shares some similarities with the Ising Hamiltonian. At points where every ϕ_i is equal to either 0 or π ,⁴ if we map $\phi_i = 0$ to $s_i = +1$ and $\phi_i = \pi$ to $s_i = -1$, we have

$$E(\vec{\phi}(t)) = -K \cdot \sum_{i,j, i \neq j} J_{ij} \cdot \cos(\phi_i(t) - \phi_j(t)) = -K \cdot \sum_{i,j, i \neq j} J_{ij} s_i s_j = -2K \cdot \sum_{i,j, i < j} J_{ij} s_i s_j. \quad (8)$$

If we choose $K = 1/2$, the global Lyapunov function in (7) exactly matches the Ising Hamiltonian in (2) at these discrete points. But this does not mean that coupled oscillators are naturally minimising the Ising Hamiltonian, as there is no guarantee at all that the phases $\{\phi_i(t)\}$ are settling to these discrete points. In fact, networks with more than two oscillators almost always synchronise with analog phases, *i.e.*, $\{\phi_i(t)\}$ commonly settle to continuous values spread out in the phase domain as opposed to converging towards 0 and π . As an example, Fig. 3(a) shows the phase responses of 20 oscillators connected in a random graph. As phases do not settle to the discrete points discussed above, the Lyapunov function they minimise becomes irrelevant to the Ising Hamiltonian, rendering the system ineffective for solving Ising problems. While one may think that the analog phases can still serve as solutions when rounded to the nearest discrete points, experiments in Sect. 4.2 show that the quality of these solutions is very poor compared with our scheme of Ising machines proposed in this paper.

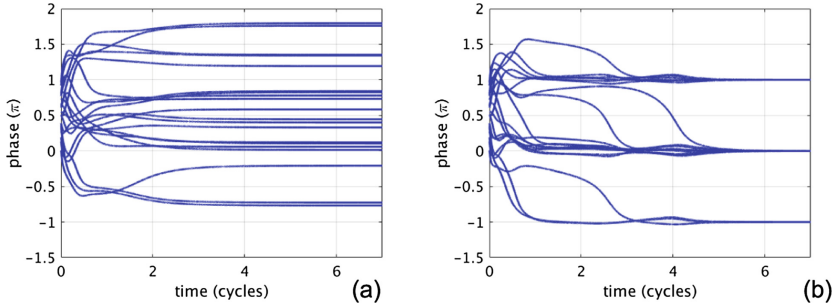


Fig. 3. Phases of 20 oscillators with random $\{J_{ij}\}$ generated by `rudy -rnd_graph 20 50 10001`: (a) without SYNC; (b) with $K_s = 1$.

⁴ More generally, we can use $\{2k\pi \mid k \in \mathbf{Z}\}$ and $\{2k\pi + \pi \mid k \in \mathbf{Z}\}$ to represent the two states for each oscillator's phase.

3.3 Network of Coupled Oscillators Under SHIL and Its Global Lyapunov Function

In our scheme, a common SYNC signal at $2\omega^*$ is injected to every oscillator in the network. Through the mechanism of SHIL, the oscillator phases are binarised. The example shown in Fig. 3(b) confirms that this is indeed the case: under SHIL, the phases of 20 oscillators connected in the same random graph now settle very close to discrete points. To write the model for such a system, we recall from Sect. 3.1 that a $2\omega^*$ perturbation introduces a π -periodic coupling term (e.g., $\sin(2\phi)$) in the phase dynamics. Therefore, we directly write the model as follows and show its derivation in [35].

$$\frac{d}{dt}\phi_i(t) = -K \cdot \sum_{j=1, j \neq i}^n J_{ij} \cdot \sin(\phi_i(t) - \phi_j(t)) - K_s \cdot \sin(2\phi_i(t)), \quad (9)$$

where K_s represents the strength of coupling from SYNC.

Remarkably, there is a global Lyapunov function for this new type of coupled oscillator system. It can be written as

$$E(\vec{\phi}(t)) = -K \cdot \sum_{i,j, i \neq j} J_{ij} \cdot \cos(\phi_i(t) - \phi_j(t)) - K_s \cdot \sum_{i=1}^n \cos(2\phi_i(t)). \quad (10)$$

Now, we show that E in (10) is indeed a global Lyapunov function. To do so, we first differentiate E with respect to $\vec{\phi}$. We observe that the first component of E is the sum of $(n^2 - n)$ number of $\cos()$ terms. Among them, for any given index k , variable ϕ_k appears a total of $2 \cdot (n - 1)$ times. It appears $(n - 1)$ times as the subtrahend inside $\cos()$: these $(n - 1)$ terms are $J_{kl} \cdot \cos(\phi_k(t) - \phi_l(t))$, where $l = 1, \dots, n$ and $l \neq k$. For the other $(n - 1)$ times, it appears as the minuend inside $\cos()$: in $J_{lk} \cdot \cos(\phi_l(t) - \phi_k(t))$, where $l = 1, \dots, n, l \neq k$. So when we differentiate E with respect to ϕ_k , we have

$$\begin{aligned} \frac{\partial E(\vec{\phi}(t))}{\partial \phi_k(t)} &= -K \cdot \sum_{l=1, l \neq k}^n J_{kl} \frac{\partial}{\partial \phi_k(t)} [\cos(\phi_k(t) - \phi_l(t))] - K \cdot \sum_{l=1, l \neq k}^n J_{lk} \frac{\partial}{\partial \phi_k(t)} [\cos(\phi_l(t) - \phi_k(t))] \\ &\quad - K_s \cdot \frac{\partial}{\partial \phi_k(t)} \cos(2\phi_k(t)) \end{aligned} \quad (11)$$

$$= K \cdot \sum_{l=1, l \neq k}^n J_{kl} \sin(\phi_k(t) - \phi_l(t)) - K \cdot \sum_{l=1, l \neq k}^n J_{lk} \sin(\phi_l(t) - \phi_k(t)) + K_s \cdot 2 \cdot \sin(2\phi_k(t)) \quad (12)$$

$$= K \cdot \sum_{l=1, l \neq k}^n J_{kl} \cdot 2 \cdot \sin(\phi_k(t) - \phi_l(t)) + K_s \cdot 2 \cdot \sin(2\phi_k(t)) \quad (13)$$

(using $\sin(x) = -\sin(-x)$ and $J_{lk} = J_{kl}$)

$$= -2 \cdot \frac{d\phi_k(t)}{dt}. \quad (14)$$

Therefore,

$$\frac{\partial E(\vec{\phi}(t))}{\partial t} = \sum_{k=1}^n \left[\frac{\partial E(\vec{\phi}(t))}{\partial \phi_k(t)} \cdot \frac{d\phi_k(t)}{dt} \right] \quad (15)$$

$$= -2 \cdot \sum_{k=1}^n \left(\frac{d\phi_k(t)}{dt} \right)^2 \leq 0. \quad (16)$$

Thus, we have proved that (10) is indeed a global Lyapunov function the coupled oscillators under SHIL naturally minimise over time. A similar but more detailed proof for the general case where we do not assume sinusoidal coupling functions is given in [35].

At the discrete points (phase values of $0/\pi$), because $\cos(2\phi_i) \equiv 1$, (10) reduces to

$$E(\vec{\phi}(t)) \approx -K \cdot \sum_{i,j, i \neq j} J_{ij} \cdot \cos(\phi_i(t) - \phi_j(t)) - n \cdot K_s, \quad (17)$$

where $n \cdot K_s$ is a constant. By choosing $K = 1/2$, we can then make (17) equivalent to the Ising Hamiltonian in (2) with a constant offset.

Note that the introduction of SYNC does not change the relative E levels between the discrete points, but modifies them by the *same* amount. However, with SYNC, all phases can be forced to eventually take values near either 0 or π —the system now tries to reach a binary state that minimises the Ising Hamiltonian, thus functioning as an Ising machine. We emphasise that this is *not* equivalent to running the system without SHIL and then rounding the analog phase solutions to discrete values as a post-processing step. Instead, the introduction of SHIL modifies the energy landscape of E , changes the dynamics of the coupled oscillator system, and as we show in Sect. 4, results in greatly improved minimisation of the Ising Hamiltonian.

Our approach here is analogous to several existing schemes to map a system’s Lyapunov function to a minimisation objective, including the Hopfield-Tank neural network proposed for the travelling salesman problem [39], and the more recent work on designing differential equations for satisfiability (SAT) problems [40, 41]. Based on the equivalence established above of the Ising Hamiltonian and the Lyapunov function (10) of coupled oscillators, our OIM can be used on a wide class of problems with an Ising formulation [11], including the problems addressed in these other schemes.

It is worth noting, also, that the Lyapunov function in (10) will, in general, have many local minima and there is no guarantee the oscillator-based Ising machine will settle at or near any global optimal state. However, as we show in [35], when judicious amounts of noise are introduced via a noise level parameter K_n , it becomes more likely to settle to lower minima. Indeed, the several parameters in the Ising machine— K , K_s and K_n —all play an important role in its operation and should be given suitable values. Furthermore, K , K_s , K_n can also be time varying, creating various “annealing schedules”. As we show in

Sect. 4, this feature gives us considerable flexibility in operating oscillator-based Ising machines for good performance.

3.4 Coupled Oscillator Networks with Frequency Variations

A major obstacle to the practical implementation of large-scale Ising machines is variability. While few analyses exist for assessing the effects of variability for previous Ising machine technologies (Sect. 2), the effect of variability on our oscillator-based Ising machine scheme is easy to analyse, predicting that performance degrades gracefully.

One very attractive feature of oscillators is that variability, regardless of the nature and number of elemental physical sources, eventually manifests itself essentially in only one parameter, namely the oscillator’s natural frequency. As a result, the effect of variability in an oscillator network is that there is a spread in the natural frequencies of the oscillators. Taking this into consideration, our model can be revised as

$$\frac{d}{dt}\phi_i(t) = \omega_i - \omega^* - \omega_i \cdot K \cdot \sum_{j=1, j \neq i}^n J_{ij} \cdot \sin(\phi_i(t) - \phi_j(t)) - \omega_i \cdot K_s \cdot \sin(2\phi_i(t)). \quad (18)$$

As it turns out, there is also a global Lyapunov function associated with this system.

$$E(\vec{\phi}(t)) = -K \cdot \sum_{i,j, i \neq j} J_{ij} \cdot \cos(\phi_i(t) - \phi_j(t)) - K_s \cdot \sum_{i=1}^n \cos(2\phi_i(t)) - 2 \sum_{i=1}^n \frac{\omega_i - \omega^*}{\omega_i} \phi_i. \quad (19)$$

This can be proven as follows.

$$\frac{\partial E(\vec{\phi}(t))}{\partial \phi_k(t)} = K \cdot \sum_{l=1, l \neq k}^n J_{kl} \cdot 2 \cdot \sin(\phi_k(t) - \phi_l(t)) + K_s \cdot 2 \cdot \sin(2\phi_k(t)) - 2 \frac{\omega_k - \omega^*}{\omega_k} \quad (20)$$

$$= - \frac{2}{\omega_k} \cdot \frac{d\phi_k(t)}{dt}. \quad (21)$$

Therefore,

$$\frac{dE(\vec{\phi}(t))}{dt} = - \sum_{k=1}^n \frac{2}{\omega_k} \left(\frac{d\phi_k(t)}{dt} \right)^2 \leq 0. \quad (22)$$

Note that (19) differs from (10) only by a weighted sum of ϕ_i —it represents essentially the same energy landscape but tilted linearly with the optimisation variables. While it can still change the locations and values of the solutions, its effects are easy to analyse given a specific combinatorial optimisation problem. Also, as the coupling coefficient K gets larger, the effect of variability can be reduced. Small amounts of variability merely perturb the locations of minima a little, *i.e.*, the overall performance of the Ising machine remains essentially unaffected. Very large amounts of variability can, of course, eliminate minima that would exist if there were no variability. However, another great advantage

of using oscillators is that even in the presence of large variability, the oscillator frequencies can be calibrated (*e.g.*, using a voltage-controlled oscillator (VCO) scheme) prior to each run of the machine. As a result, the spread in frequencies can be essentially eliminated in a practical and easy-to-implement way.

4 Examples

In this section, we demonstrate the feasibility and efficacy of our oscillator-based Ising machine scheme by applying it to several MAX-CUT examples and a graph-colouring problem.

4.1 Small MAX-CUT Problems

Given an undirected graph, the MAX-CUT problem [13, 42] asks us to find a subset of vertices such that the total weights of the cut set between this subset and the remaining vertices are maximised. As an example, Fig. 4 shows a size-8 cubic graph, where each vertex is connected to three others—neighbours on both sides and the opposing vertex. As shown in Fig. 4, dividing the 8 vertices randomly yields a cut size of 5; grouping even and odd vertices, which one may think is the best strategy, results in a cut size of 8; the maximum cut is actually 10, with one of the solutions shown in the illustration. Changing the edge weights to non-unit values can change the maximum cut and also make the solution look less regular, often making the problem more difficult to solve. While the problem may not seem challenging at size 8, it quickly becomes intractable as the size of the graph grows. In fact, MAX-CUT is one of Karp’s 21 NP-complete problems [10].

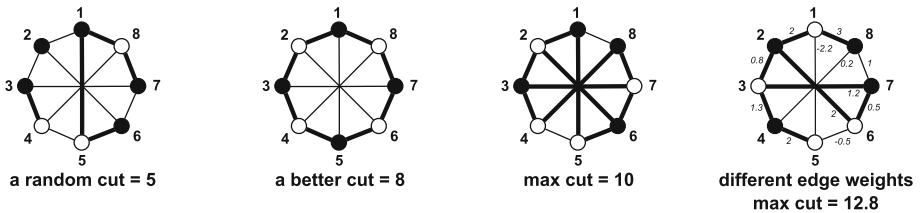


Fig. 4. Illustration of different cut sizes in a 8-vertex cubic graph with unit edge weights, and another one with random weights (rightmost).

The MAX-CUT problem has a direct mapping to the Ising model [10], by choosing J_{ij} to be the opposite of the weight of the edge between vertices i and j , *i.e.*, $J_{ij} = -w_{ij}$. To explain this mapping scheme, we can divide the vertices into two sets— V_1 and V_2 . Accordingly, all the edges in the graph are separated into three groups—those that connect vertices within V_1 , those within V_2 , and the cut set containing edges across V_1 and V_2 . The sums of the weights in these

three sets are denoted by S_1 , S_2 and S_{cut} . Together, they constitute the total edge weights of the graph, which is also the negation of the sum of all the J_{ij} s:

$$S_1 + S_2 + S_{cut} = \sum_{i,j, i < j} w_{ij} = - \sum_{i,j, i < j} J_{ij}. \tag{23}$$

We then map this division of vertices to the values of Ising spins, assigning $+1$ to a spin i if vertex $v_i \in V_1$, and -1 if the vertex is in V_2 . The Ising Hamiltonian in (2) can then be calculated as

$$\begin{aligned} H &= \sum_{i,j, i < j} J_{ij} s_i s_j \\ &= \sum_{i < j, v_i, v_j \in V_1} J_{ij} (+1)(+1) + \sum_{i < j, v_i, v_j \in V_2} J_{ij} (-1)(-1) + \sum_{i < j, v_i \in V_1, v_j \in V_2} J_{ij} (+1)(-1) \\ &= \sum_{i < j, v_i, v_j \in V_1} J_{ij} + \sum_{i < j, v_i, v_j \in V_2} J_{ij} - \sum_{i < j, v_i \in V_1, v_j \in V_2} J_{ij} \\ &= -(S_1 + S_2 - S_{cut}) = \sum_{i,j, i < j} J_{ij} - 2 \cdot S_{cut}. \end{aligned} \tag{24}$$

Therefore, when the Ising Hamiltonian is minimised, the cut size is maximised.

To show that an oscillator-based Ising machine can indeed be used to solve MAX-CUT problems, we simulated the Kuramoto model in (9) while making the J_{ij} s represent the unit-weight cubic graph in Fig. 4. The magnitude of SYNC is fixed at $K_s = 3$, while we ramp up the coupling strength K from 0 to 5. Results from the deterministic model ($K_n = 0$) and the stochastic model ($K_n = 0.1$) are shown in Figs. 5 and 6 respectively. In the simulations, oscillators started with random phases between 0 and π ; after a while, they all settled to one of the two phase-locked states separated by π . These two groups of oscillators represent the two subsets of vertices in the solution. The results for the 8 spins shown in Figs. 5 and 6 are $\{+1, -1, +1, -1, -1, +1, -1, +1\}$

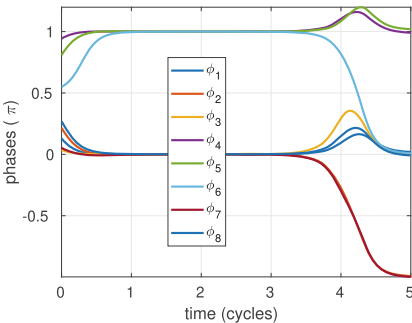


Fig. 5. Phases of oscillators solving a size-8 MAX-CUT problem without noise.

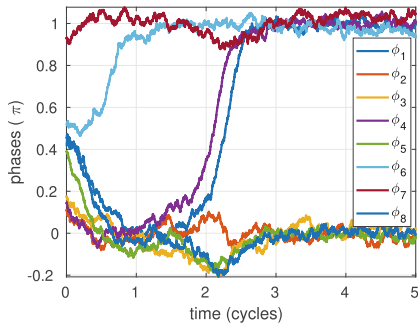


Fig. 6. Phases of oscillators solving a size-8 MAX-CUT problem with noise.

and $\{-1, +1, +1, -1, +1, -1, -1, +1\}$ respectively; both are globally optimal solutions.

A minimal code for reproducing these results is show in [35]. Note that these are simulations on stochastic differential equations with random initial conditions. Every run will return different waveforms; there is no guarantee that the global optimum will be reached on every run.

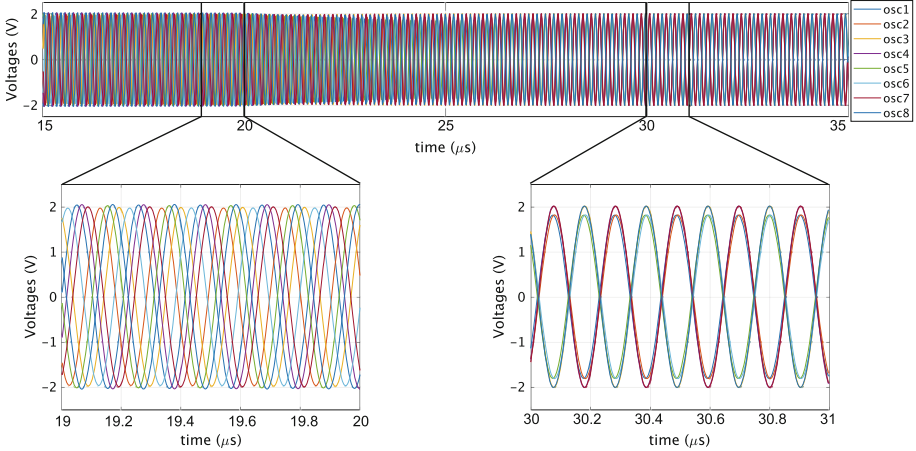


Fig. 7. Simulation results from ngspice on 8 coupled oscillators.

We have also directly simulated coupled oscillators at the SPICE level to confirm the results obtained on phase macromodels. Such simulations are at a lower level than phase macromodels and are less efficient. But they are closer to physical reality and are useful for circuit design. In the simulations, 8 cross-coupled LC oscillators are tuned to a frequency of 5 MHz. They are coupled through resistors, with conductances proportional to the coupling coefficients; in this case, we use $J_{ij} \cdot 1/100 \text{ k}\Omega$. Results from transient simulation using ngspice-28 are shown in Fig. 7. The 8 oscillators' phases settle into two groups $\{1, 4, 6, 7\}$ and $\{2, 3, 5, 8\}$, representing one of the optimal solutions for the MAX-CUT problem. They synchronise within $20 \mu\text{s}$ after oscillation starts, which is about 100 cycles. We have tried this computational experiment with different random initial conditions; like phase-macromodels, the SPICE-level simulations of these coupled oscillators reliably return optimal solutions for this size-8 MAX-CUT problem.

We have also implemented these 8 coupled LC oscillators on a breadboard; a photo of it is shown in Fig. 8. The inductance of the LC oscillators is provided by fixed inductors of size $33 \mu\text{H}$. The capacitance is provided by trimmer capacitors with a maximum value of 50 pF ; we have tuned them to around 30 pF such that the natural frequencies of all oscillators are about 5 MHz. The nonlinearity for sustaining the LC oscillation is implemented by cross-coupled CMOS

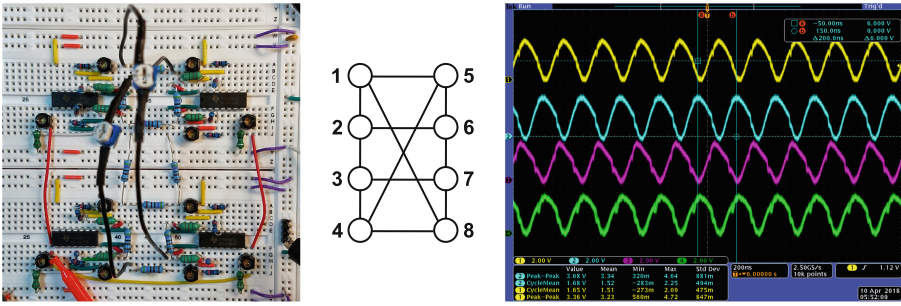


Fig. 8. A simple oscillator-based Ising machine solving size-8 cubic graph MAX-CUT problems: (a) breadboard implementation with 8 CMOS LC oscillators; (b) illustration of the connections; (c) oscilloscope measurements showing waveforms of oscillator 1~4.

inverters from TI SN74HC04N chips. SYNC is supplied through the GND pins of these chips. The results have been observed using two four-channel oscilloscopes; a screenshot of one of them is shown in Fig. 8. Through experiments with various sets of edge weights, we have validated that this is indeed a proof-of-concept hardware implementation of oscillator-based Ising machines for size-8 cubic-graph MAX-CUT problems.

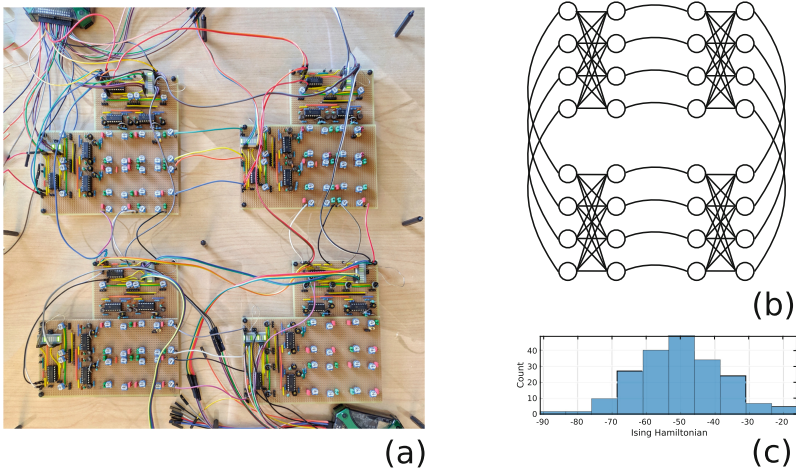


Fig. 9. A size-32 oscillator-based Ising machine: (a) photo of the implementation on perboards; (b) illustration of the connectivity; (c) a typical histogram of the energy values achieved in 200 runs on a random size-32 Ising problem; the lowest energy level is -88 and is achieved once in this case.

Using the same type of oscillators, we have built hardware Ising machines of larger sizes. Figure 9 shows a size-32 example implementing a type of connectivity

known as the Chimera graph, much like the quantum Ising machines manufactured by D-Wave Systems. In this graph, oscillators are organised into groups of 8, with denser connections within the groups and sparse ones in between. The hardware is on perfboards, with components soldered on the boards so that the setup is more permanent than those on breadboards. Connections are implemented using rotary potentiometers. Next to each potentiometer we have designed male pin connectors soldered on the board such that the polarity of each connection can be controlled by shorting different pins using female jumper caps. When encoding Ising problems, we have also colour-coded the jumper caps to make debugging easier, as can be seen in the photo as red and green dots next to the four arrays of white round potentiometers. To read the phases of the oscillators, instead of using multichannel oscilloscopes, we have soldered TI SN74HC86N Exclusive-OR (XOR) gate chips on board. The XOR operation of an oscillator's response and a reference signal converts the oscillating waveform into a high or low voltage level, indicating if the oscillator's phase is aligned with or opposite to the reference phase. The voltage level can then be picked up by a multichannel logic analyzer. The entire setup is powered by two Digilent Analog Discovery 2 devices, which are portable USB devices that integrate power supplies, logic analyzers and function generators. We have tried random Ising problems by programming each connection with a random polarity using the jumper caps. A typical histogram of the Ising Hamiltonians achieved is shown in Fig. 9(c). Note that because J_{ij} s have random polarities, a random solution would have an average energy level of zero. In comparison, the results measured from the hardware are always below 0, and sometimes achieve the global minimum. While such a hand-soldered system is nontrivial to assemble and operate, and its size of 32 cannot be characterised as large scale, it is a useful proof of concept for implementing oscillator-based Ising machines using standard CMOS technologies, and serves as a very solid basis for our future plans to scale the implementations with custom PCBs and custom ICs.

4.2 MAX-CUT Benchmark Problems

In this section, we demonstrate the efficacy of oscillator-based Ising machines for solving larger-scale MAX-CUT problems. Specifically, we have run simulations on all the problems in a widely used set of MAX-CUT benchmarks known as the G-set [43].⁵ Problem sizes range from 800 to 3000.⁶ In the experiments, we operated the Ising machine for all the problems with a single annealing schedule, *i.e.*, we did not tune our Ising machine parameters individually for different problems. Each problem was simulated with 200 random instances. In Table 1, we list the results and runtime alongside those from several heuristic algorithms developed for MAX-CUT—Scatter Search (SS) [44], CirCut [45], and Variable

⁵ The G-set problems are available for download as set1 at <http://www.opticom.es/maxcut>.

⁶ G1~21 are of size 800; G22~42 are of size 2000; G43~47, G51~54 are of size 1000; G48~50 are of size 3000.

Neighbourhood Search with Path Relinking (VNSPR) [13].⁷ We also list the performances of simulated annealing from a recent study [42], the only one we were able to find that contains results for all the G-set problems.

From Table 1, we observe that our oscillator-based Ising machine is indeed effective—it finds best-known cut values for 38 out of the 54 problems, 17 of which are even better than those reported in the above literature. Moreover, in the 200 random instances, the best cut is often reached more than once—the average n_{max} for all benchmarks is 20 out of 200. If we relax the objective and look at the number of instances where 99.9% of the cut value is reached, represented by $n_{0.999}$, the average is 56, more than a quarter of the total trials. The results can in fact be improved further if we tailor the annealing schedule for each problem. But to show the effectiveness and generality of our scheme, we have chosen to use the same annealing schedule for all the problems.

In the annealing schedule we used, the coupling strength K increases linearly, the noise level K_n steps up from 0 to 1, while SYNC’s amplitude K_s ramps up and down multiple times. Such a schedule was chosen empirically and appears to work well for most G-set problems. Figure 10 shows the behaviour of oscillator phases and the instantaneous cut values under this schedule for solving benchmark problem G1 to its best-known cut size. Some MATLAB[®] code to illustrate the annealing schedule is shown in [35]. The code uses MATLAB[®]’s SDE solver and is thus much slower than an implementation in C++ we used to generate the results in Table 1. We plan to release all our code as open-source software so that others can verify and build on our work.

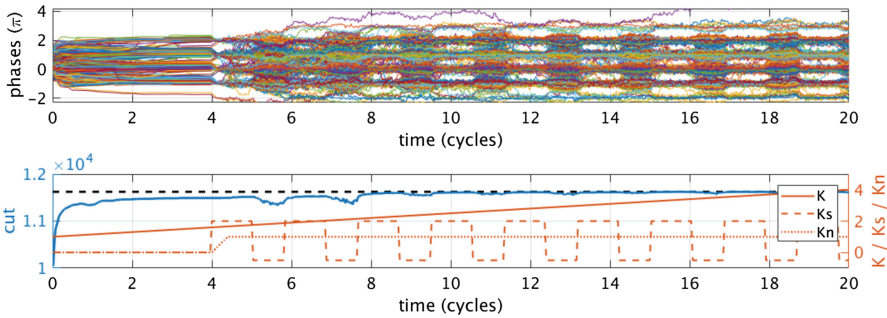


Fig. 10. Coupled oscillators solving MAX-CUT benchmark problem G1 [43] to its best-known cut size 11624.

The fact that we were using a fixed schedule also indicates that the actual hardware time for the Ising machine to solve all these benchmarks is the same, regardless of problem size and connectivity. Note that in Fig. 10, the end time 20 means 20 oscillation cycles, but this end time is predicated on a coupling

⁷ Their results and runtime are available for download at <http://www.opticom.es/maxcut> in the “Computational Experiences” section.

Table 1. Results of oscillator-based Ising machines run on MAX-CUT benchmarks in the G-set, compared with several heuristic algorithms. Time reported in this table is for a single run. n_{max} is the number of runs out of the 200 trials where the cut reaches the maximum of 200; $n_{0.999}$ is the number for it to reach 99.9% of the maximum.

Benchmark	SS	Time	CirCut	Time	VNSPR	Time	SA	Time	OIM	Time	n_{max}	$n_{0.999}$
G1	11624	139	11624	352	11621	22732	11621	295	11624	52.6	43	123
G2	11620	167	11617	283	11615	22719	11612	327	11620	52.7	1	87
G3	11622	180	11622	330	11622	23890	11618	295	11622	52.4	10	117
G4	11646	194	11641	524	11600	24050	11644	294	11646	52.7	20	133
G5	11631	205	11627	1128	11598	23134	11628	300	11631	52.6	3	121
G6	2165	176	2178	947	2102	18215	2178	247	2178	52.8	4	46
G7	1982	176	2003	867	1906	17716	2006	205	2000	52.9	17	21
G8	1986	195	2003	931	1908	19334	2005	206	2004	52.8	2	26
G9	2040	158	2048	943	1998	15225	2054	206	2054	52.6	2	2
G10	1993	210	1994	881	1910	16269	1999	205	2000	52.9	21	58
G11	562	172	560	74	564	10084	564	189	564	6.7	6	6
G12	552	242	552	58	556	10852	554	189	556	6.3	25	25
G13	578	228	574	62	580	10749	580	195	582	6.4	3	3
G14	3060	187	3058	128	3055	16734	3063	252	3061	14.6	27	91
G15	3049	143	3049	155	3043	17184	3049	220	3049	16.1	41	145
G16	3045	162	3045	142	3043	16562	3050	219	3052	14.5	8	53
G17	3043	313	3037	366	3030	18555	3045	219	3046	14.6	5	52
G18	988	174	978	497	916	12578	990	235	990	14.7	3	3
G19	903	128	888	507	836	14546	904	196	906	14.5	13	13
G20	941	191	941	503	900	13326	941	195	941	14.7	160	160
G21	930	233	931	524	902	12885	927	195	931	14.6	10	10
G22	13346	1336	13346	493	13295	197654	13158	295	13356	58.7	3	93
G23	13317	1022	13317	457	13290	193707	13116	288	13333	58.6	8	54
G24	13303	1191	13314	521	13276	195749	13125	289	13329	59.0	6	23
G25	13320	1299	13326	1600	12298	212563	13119	316	13326	58.7	6	45
G26	13294	1415	13314	1569	12290	228969	13098	289	13313	58.9	4	81
G27	3318	1438	3306	1456	3296	35652	3341	214	3323	59.0	18	24
G28	3285	1314	3260	1543	3220	38655	3298	252	3285	61.2	1	5
G29	3389	1266	3376	1512	3303	33695	3394	214	3396	58.9	2	8
G30	3403	1196	3385	1463	3320	34458	3412	215	3402	59.0	12	16
G31	3288	1336	3285	1448	3202	36658	3309	214	3296	59.1	5	15
G32	1398	901	1390	221	1396	82345	1410	194	1402	17.5	5	5
G33	1362	926	1360	198	1376	76282	1376	194	1374	15.9	1	1
G34	1364	950	1368	237	1372	79406	1382	194	1374	15.9	24	24
G35	7668	1258	7670	440	7635	167221	7485	263	7675	37.1	5	29
G36	7660	1392	7660	400	7632	167203	7473	265	7663	37.6	3	58
G37	7664	1387	7666	382	7643	170786	7484	288	7679	37.8	1	15
G38	7681	1012	7646	1189	7602	178570	7479	264	7679	37.7	7	18

(continued)

Table 1. (*continued*)

Benchmark	SS	Time	CirCut	Time	VNSPR	Time	SA	Time	OIM	Time	n_{max}	$n_{0.999}$
G39	2393	1311	2395	852	2303	42584	2405	209	2404	37.2	1	1
G40	2374	1166	2387	901	2302	39549	2378	208	2389	38.1	7	7
G41	2386	1017	2398	942	2298	40025	2405	208	2401	37.8	20	71
G42	2457	1458	2469	875	2390	41255	2465	210	2469	37.3	4	4
G43	6656	406	6656	213	6659	35324	6658	245	6660	29.1	17	129
G44	6648	356	6643	192	6642	34519	6646	241	6648	29.2	21	129
G45	6642	354	6652	210	6646	34179	6652	241	6653	29.1	10	53
G46	6634	498	6645	639	6630	38854	6647	245	6649	29.1	9	13
G47	6649	359	6656	633	6640	36587	6652	242	6656	29.1	16	91
G48	6000	20	6000	119	6000	64713	6000	210	6000	23.2	194	194
G49	6000	35	6000	134	6000	64749	6000	210	6000	23.2	180	180
G50	5880	27	5880	231	5880	147132	5858	211	5874	25.6	10	94
G51	3846	513	3837	497	3808	89966	3841	234	3846	18.4	23	68
G52	3849	551	3833	507	3816	95985	3845	228	3848	18.4	10	49
G53	3846	424	3842	503	3802	92459	3845	230	3846	18.4	9	102
G54	3846	429	3842	524	3820	98458	3845	228	3850	18.5	3	40

strength of $K \sim 1$. The actual value of K for each oscillator depends on the PPV function as well as the amplitude of perturbation from other oscillators, as we show in the derivation of Gen-Adler in [35]. As an example, for the LC oscillators we use in Sect. 4.1 with 100k resistive coupling, $K \approx 0.02$. This indicates that it takes less than 100 cycles for the oscillators to synchronise in phase, which is consistent with measurements. For such a coupled LC oscillator network, a hardware time of 20 in Fig. 10 represents approximately 2000 cycles of oscillation; for 5 MHz oscillators that takes 0.4 ms. If we use GHz nano-oscillators, the computation time can be well within a microsecond. In comparison, the runtime of the several heuristic algorithms listed in Table 1, even with faster CPUs and parallel implementations in the future, is unlikely to ever drop to this range.

As the hardware time is fixed, the runtime we report in Table 1 for our Ising machines is the time for simulating the SDEs of coupled oscillators on CPUs for one run. While we list runtime results for each algorithm in Table 1, note that they come from different sources and are measured on different platforms. Results for SS, CirCut and VNSPR were obtained from Dual Intel Xeon at 3.06 GHz with 3.2 GB of RAM; SA was run on Intel Xeon E3-1245v2 at 3.4 GHz with 32GB of RAM [42]. To make the results generally comparable, we ran our simulations on a modest personal desktop with Intel Xeon E5-1603v3 at 2.8 GHz with 16 GB of RAM. Even so, it came as a nice surprise to us that even by simulating SDEs we were able to solve the benchmarks efficiently. Another notable feature of our method is that unlike other algorithms, SDE simulation does not know about the Ising Hamiltonian or cut value—it never needs to evaluate the energy function or relative energy changes, which are implicit in the dynamics of differential equations, yet it proves effective and fast.

We also ran more computational experiments on the G-set benchmarks in order to study the mechanism of oscillator-based Ising machines. We created several variants of the Ising machine used above by removing different components in its operation. For each variant, we re-ran 200 random instances for each of the 54 benchmarks, generating 10800 cut values. In Fig. 11, we compare the quality of these cut values with results from the unaltered Ising machine by plotting histograms of the distances of the cut values to their respective maxima. In the first variant, we removed noise from the model by setting $K_n \equiv 0$. The solutions become considerably worse, confirming that noise helps the coupled oscillator system settle to lower energy states.

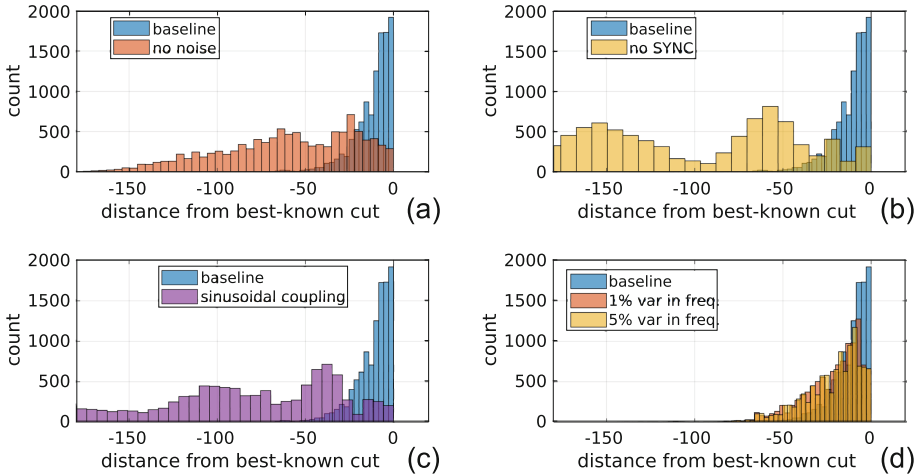


Fig. 11. Histograms of the cut values achieved by several variants of the Ising machine, compared with the baseline results used in Table 1.

In the next variant, we removed SYNC by setting $K_s \equiv 0$. Without SYNC, the system becomes a simple coupled oscillator system with phases that take a continuum of values, as discussed in Sect. 3.2. The settled analog values of the phases that were then thresholded to 0 or π to correspond to Ising spins. As shown in Fig. 11, the results become significantly worse; indeed, none of the best-known results were reached. This indicates that the SYNC signal and the mechanism of SHIL we introduce to the coupled oscillator networks are indeed essential for them to operate as Ising machines.

Our baseline Ising machine actually uses a smoothed square function $\tanh(\sin(\cdot))$ for the coupling, as opposed to the $\sin(\cdot)$ used in the original Kuramoto model, as shown in the code in [35]. This changes the $\cos(\cdot)$ term in the energy function (10) to a triangle function. Such a change appears to give better results than the original, as shown in Fig. 11(c). The change requires designing oscillators with special PPV shapes and waveforms such that their cross-correlation is a square wave, which is not difficult in practice based on

our derivation in [35]. As an example, rotary traveling wave oscillators naturally have square PPVs. Ring oscillators can also be designed with various PPVs and waveforms by sizing each stage individually. We cannot say definitively that the square function we have used is optimal for Ising solution performance, but the significant improvement over sinusoidal coupling functions indicates that a fruitful direction for further exploration may be to look beyond the original Kuramoto model for oscillator-based computing.

The last variant we report here added variability to the natural frequencies of the oscillators, as in (18). We assigned Gaussian random variables to ω_i s with ω^* as the mean, and 0.01 (1%) and 0.05 (5%), respectively, as the standard deviations for two separate runs. From Fig. 11(d), we observe that even with such non-trivial spread in the natural frequencies of oscillators, the performance is affected very little.

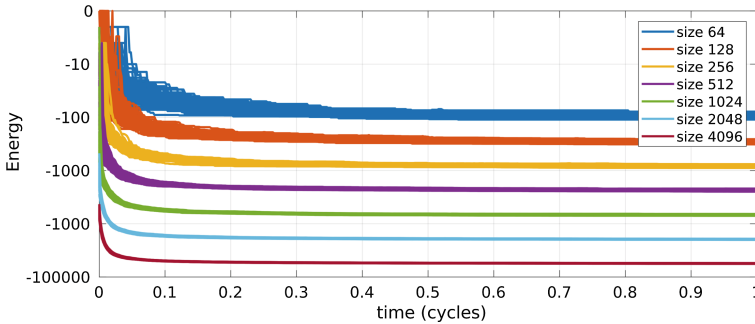


Fig. 12. Speed of energy minimisation for problems of different sizes.

Finally, we conducted a preliminary study of the scaling of the time taken by the Ising machine to reach good solutions as problem sizes increase. As the G-set benchmarks have only a few sizes (800, 1000, 2000 and 3000), we used the program (named `rudy`) that generated them to create more problems of various sizes. All generated problems used random graphs with 10% connectivity and ± 1 coupling coefficients. We simulated all of them, each for 200 instances, with fixed parameters $K = 1$, $K_s = 0.1$, $K_n = 0.01$, and show all their Ising Hamiltonians over time in Fig. 12. Much to our surprise, the speed in which the values settle appears almost constant, regardless of the problem size. While this does not necessarily mean they all converge to the global optima within the same time, this preliminary study is encouraging as it confirms the massively parallel nature of the system. For larger Ising problems, our Ising machine only needs to scale linearly in hardware size with the number of spins, but does not necessarily require much more time to reach a solution.

4.3 A Graph Colouring Example

As mentioned in Sect. 2, many problems other than MAX-CUT can be mapped to the Ising model [11] and solved by an oscillator-based Ising machine. Here we show an example of a graph colouring problem—assigning four colours to the 51 states (including a federal district) of America such that no two adjacent states have the same colour.⁸

Each state is represented as a vertex in the graph. When two states are adjacent, there is an edge in the graph that connects the corresponding vertices. For every vertex i , we assign four spins— s_{iR} , s_{iG} , s_{iB} and s_{iY} to represent its colouring scheme; when only one of them is $+1$, the vertex is successfully coloured as either red, green, blue or yellow. Then we write an energy function H associated with these $4 \times 51 = 204$ spins as follows:

$$\begin{aligned}
 H = & \sum_i^n (2 + s_{iR} + s_{iG} + s_{iB} + s_{iY})^2 \\
 & + \sum_{(i,j) \in \mathbb{E}}^{n_{\mathbb{E}}} [(1 + s_{iR})(1 + s_{jR}) + (1 + s_{iG})(1 + s_{jG}) + (1 + s_{iB})(1 + s_{jB}) + (1 + s_{iY})(1 + s_{jY})],
 \end{aligned}
 \tag{25}$$

where $n = 51$ is the number of vertices, \mathbb{E} represents the edge set, $n_{\mathbb{E}}$ is the number of edges and in this case equal to 220.⁹

The first term of H is a sum of squares never less than zero; it reaches zero only when $\{s_{iR}, s_{iG}, s_{iB}, s_{iY}\}$ contains three -1 s and one $+1$ for every i , *i.e.*, each state has a unique colour. The latter term is also a sum that is always greater than or equal to zero, as each spin can only take a value in $\{-1, +1\}$; it is zero when $s_{iX} = s_{jX} = +1$ never occurs for any edge connecting i and j , and for any colour $X \in \{R, G, B, Y\}$, *i.e.*, adjacent states do not share the same colour. Therefore, when H reaches its minimum value 0, the spin configuration represents a valid colouring scheme—following the indices of the $+1$ spins $\{i, X \mid s_{iX} = +1\}$, we can then assign colour X to state i .

Note that when expanding the sum of squares in (25), we can use the fact $s_{iX}^2 \equiv 1$ to eliminate the square terms. This means H contains only products of two spins—modelled by J_{ij} s, and self terms—modelled by h_i . These Ising coefficients can then be used to determine the couplings in an oscillator-based Ising machine.

We simulated these 204 coupled oscillators and show the results in Fig. 13. In the simulation, we kept K and K_n constant while ramping K_s up and down 5 times. We found the Ising machine to be effective with this schedule as it could colour the map successfully in more than 50% of the random trials and returned many different valid colouring schemes.

⁸ Ising machines can be used on general graph colouring problems, and this four-colouring problem is chosen here for illustrative purposes. Four-colouring a planar graph is actually not NP-hard and there exist polynomial-time algorithms for it [46].

⁹ Hawaii and Alaska are considered adjacent such that their colours will be different in the map.

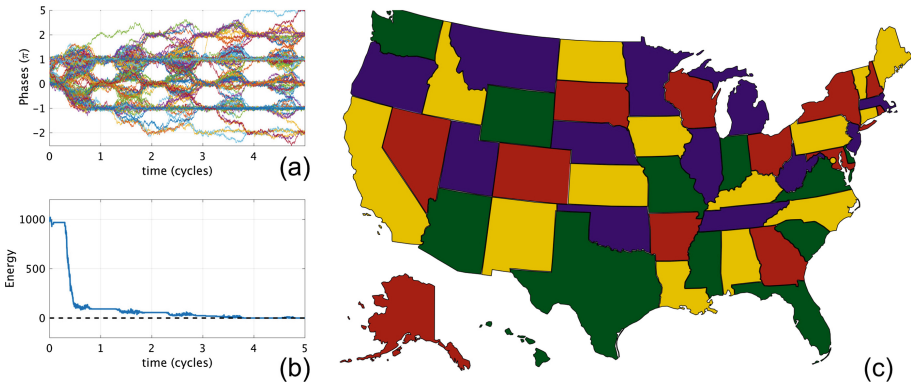


Fig. 13. Coupled oscillators colouring the states in the US map: (a) phases of oscillators evolve over time; (b) energy function (25) decreases during the process; (c) the resulting US map colouring scheme.

5 Conclusion

In this paper, we have proposed a novel scheme for implementing Ising machines using self-sustaining nonlinear oscillators. We have shown how coupled oscillators naturally minimise an “energy” represented by their global Lyapunov function, and how introducing the mechanism of subharmonic injection locking modifies this function to encode the Ising Hamiltonian for minimisation. The validity and feasibility of the scheme have been examined via multiple levels of simulation and proof-of-concept hardware implementations. Simulations run on larger-scale benchmark problems have also shown promising results in both speed and the quality of solutions. We believe that our scheme constitutes an important and practical means for the implementation of scalable Ising machines.

Acknowledgements. The authors would like to thank the reviewers for the useful comments and in particular anonymous reviewer No. 2 for pointing us to Ercsey-Ravasz/Toroczkai and Yin’s work on designing dynamical systems to solve NP-complete problems.

References

1. Ising, E.: Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei* **31**(1), 253–258 (1925)
2. Brush, S.G.: History of the Lenz-Ising Model. *Rev. Mod. Phys.* **39**, 883–893 (1967)
3. Barahona, F.: On the computational complexity of Ising spin glass models. *J. Phys. A: Math. Gen.* **15**(10), 3241 (1982)
4. Marandi, A., Wang, Z., Takata, K., Byer, R.L., Yamamoto, Y.: Network of time-multiplexed optical parametric oscillators as a coherent Ising machine. *Nat. Photonics* **8**(12), 937–942 (2014)

5. McMahon, P.L., et al.: A fully-programmable 100-spin coherent Ising machine with all-to-all connections. *Science* **354**, 5178 (2016)
6. Inagaki, T., et al.: A coherent Ising machine for 2000-node optimization problems. *Science* **354**(6312), 603–606 (2016)
7. Johnson, M.W., et al.: Quantum annealing with manufactured Spins. *Nature* **473**(7346), 194 (2011)
8. Bian, Z., Chudak, F., Israel, R., Lackey, B., Macready, W.G., Roy, A.: Discrete optimization using quantum annealing on sparse Ising models. *Front. Phys.* **2**, 56 (2014)
9. Yamaoka, M., Yoshimura, C., Hayashi, M., Okuyama, T., Aoki, H., Mizuno, H.: A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing. *IEEE J. Solid-State Circuits* **51**(1), 303–309 (2016)
10. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) *Complexity of Computer Computations*, pp. 85–103. Springer, Boston (1972)
11. Lucas, A.: Ising formulations of many NP problems. arXiv preprint [arXiv:1302.5843](https://arxiv.org/abs/1302.5843) (2013)
12. Wang, T., Roychowdhury, J.: Oscillator-based Ising Machine. arXiv preprint [arXiv:1709.08102](https://arxiv.org/abs/1709.08102) (2017)
13. Festa, P., Pardalos, P.M., Resende, M.G.C., Ribeiro, C.C.: Randomized heuristics for the MAX-CUT problem. *Optim. Methods Softw.* **17**(6), 1033–1058 (2002)
14. Jensen, T.R., Toft, B.: *Graph Coloring Problems*. Wiley, New York (2011)
15. Neogy, A., Roychowdhury, J.: Analysis and Design of Sub-harmonically Injection Locked Oscillators. In: *Proceedings of the IEEE DATE*, March 2012. <http://dx.doi.org/10.1109/DATE.2012.6176677>
16. Bhansali, P., Roychowdhury, J.: Gen-Adler: The generalized Adler’s equation for injection locking analysis in oscillators. In: *Proceedings of the IEEE ASP-DAC*, pp. 522–227, January 2009. <http://dx.doi.org/10.1109/ASPDAC.2009.4796533>
17. Kuramoto, Y.: Self-entrainment of a population of coupled non-linear oscillators. In: Araki, H. (ed.) *International Symposium on Mathematical Problems in Theoretical Physics*, pp. 420–422. Springer, Heidelberg (1975)
18. Kuramoto, Y.: *Chemical Oscillations, Waves and Turbulence*. Dover, New York (2003)
19. Acebrón, J.A., Bonilla, L.L., Vicente, C.J.P., Ritort, F., Spigler, R.: The kuramoto model: a simple paradigm for synchronization phenomena. *Rev. Mod. Phys.* **77**(1), 137 (2005)
20. Wang, T., Roychowdhury, J.: PHLOGON: PHase-based LOGic using oscillatory nano-systems. In: Ibarra, O.H., Kari, L., Kopecki, S. (eds.) *UCNC 2014*. LNCS, vol. 8553, pp. 353–366. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08123-6_29
21. Wang, T.: Sub-harmonic Injection Locking in Metronomes. arXiv preprint [arXiv:1709.03886](https://arxiv.org/abs/1709.03886) (2017)
22. Wang, T.: Achieving Phase-based Logic Bit Storage in Mechanical Metronomes. arXiv preprint [arXiv:1710.01056](https://arxiv.org/abs/1710.01056) (2017)
23. Aramon, M., Rosenber, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. [arXiv:1806.08815](https://arxiv.org/abs/1806.08815) [physics.comp-ph] August 2018
24. Gyoten, H., Hiromoto, M., Sato, T.: Area efficient annealing processor for ising model without random number generator. *IEICE Trans. Inf. Syst.* **E101.D**(2), 314–323 (2018)

25. Gyoten, H., Hiromoto, M., Sato, T.: Enhancing the solution quality of hardware Ising-model solver via parallel tempering. In: Proceedings of the ICCAD, ICCAD 2018, pp. 70:1–70:8. ACM, New York (2018)
26. Bian, Z., Chudak, F., Macready, W.G., Rose, G.: The Ising model: teaching an old problem new tricks. *D-Wave Syst.* **2**, 1–32 (2010)
27. Harris, R., et al.: Experimental demonstration of a robust and scalable flux qubit. *Phys. Rev. B* **81**(13), 134–510 (2010)
28. Rønnow, T.F., et al.: Defining and detecting quantum speedup. *Science* **345**(6195), 420–424 (2014)
29. Denchev, V.S., et al.: What is the computational value of finite-range tunneling? *Phys. Rev. X* **6**(3), 031015 (2016)
30. Mahboob, I., Okamoto, H., Yamaguchi, H.: An electromechanical Ising Hamiltonian. *Sci. Adv.* **2**(6), e1600236 (2016)
31. Camsari, K.Y., Faria, R., Sutton, B.M., Datta, S.: Stochastic p-bits for invertible logic. *Phys. Rev. X* **7**(3), 031014 (2017)
32. Yamamoto, K., Huang, W., Takamaeda-Yamazaki, S., Ikebe, M., Asai, T., Motomura, M.: A time-division multiplexing Ising machine on FPGAs. In: Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies, p. 3. ACM (2017)
33. Winfree, A.: Biological rhythms and the behavior of populations of coupled oscillators. *Theor. Biol.* **16**, 15–42 (1967)
34. Demir, A., Mehrotra, A., Roychowdhury, J.: Phase noise in oscillators: a unifying theory and numerical methods for characterization. *IEEE Trans. Circuits Syst.- I: Fund. Th. Appl.* **47**, 655–674 (2000). <http://dx.doi.org/10.1109/81.847872>
35. Wang, T., Roychowdhury, J.: OIM: Oscillator-based Ising Machines for Solving Combinatorial Optimisation Problems. arXiv preprint [arXiv:1903.07163](https://arxiv.org/abs/1903.07163) (2019)
36. Wang, T., Roychowdhury, J.: Design tools for oscillator-based computing systems. In: Proceedings IEEE DAC, pp. 188:1–188:6 (2015). <http://dx.doi.org/10.1145/2744769.2744818>
37. Shinomoto, S., Kuramoto, Y.: Phase transitions in active rotator systems. *Progress Theoret. Phys.* **75**(5), 1105–1110 (1986)
38. Lyapunov, A.M.: The general problem of the stability of motion. *Int. J. Control* **55**(3), 531–534 (1992)
39. Hopfield, J.J., Tank, D.W.: “Neural” computation of decisions in optimization problems. *Biol. Cybern.* **52**(3), 141–152 (1985)
40. Ercsey-Ravasz, M., Toroczkai, Z.: Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nat. Phys.* **7**(12), 966 (2011)
41. Yin, X., Sedighi, B., Varga, M., Ercsey-Ravasz, M., Toroczkai, Z., Hu, X.S.: Efficient analog circuits for boolean satisfiability. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **26**(1), 155–167 (2018)
42. Myklebust, T.: Solving maximum cut problems by simulated annealing. arXiv preprint [arXiv:1505.03068](https://arxiv.org/abs/1505.03068) (2015)
43. Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. *SIAM J. Optim.* **10**(3), 673–696 (2000)
44. Martí, R., Duarte, A., Laguna, M.: Advanced scatter search for the max-cut problem. *INFORMS J. Comput.* **21**(1), 26–38 (2009)
45. Burer, S., Monteiro, R., Zhang, Y.: Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM J. Optim.* **12**(2), 503–521 (2002)
46. Robertson, N., Sanders, D.P., Seymour, P., Thomas, R.: Efficiently four-coloring planar graphs. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 571–575. ACM (1996)