Benchmarking Max-Cut on Oscillatory Ising Machines with Kuramoto and van der Pol Oscillators

Benchmarking max-cut on oscillatory Ising machines with Kuramoto and van der Pol oscillators

Filip Sabo[a], Aida Todri-Sanial[a]

[a] Eindhoven University of Technology, Eindhoven, the Netherlands

Abstract. Oscillatory neural networks (ONNs) are energy minimizing networks that are attractive for solving combinatorial optimization problems. We investigate two different oscillator models for benchmarking the max-cut problem.

Introduction. Max-cut is one of the classical NP-hard problems. Goemans-Williamson (GW) algorithm is the best algorithm to give at least 87.8% of the solution for a given graph[1]. This paper investigates oscillatory network networks (ONNs) as an oscillatory Ising machine (OIM). ONNs are phase-based networks of the Ising model[2,3] that minimizes its energy as the network evolves. The oscillator is described by a neuron model with Kuramoto[4] and van der Pol[5] models, among others. In this paper, we benchmark the max-cut problem on OIM with two types of neuron models and compare it to the GW algorithm.

Oscillatory neural networks. An ONN is an energy-based network aiming to minimize an underlying function. These single layered networks of coupled oscillators are based on the Ising model[2,3], whose Hamiltonian is as follows:

$$H = -\sum_{(i,j)} J_{ij}\,\sigma_i\sigma_j - \mu\sum_i h_i\,\sigma_i \tag{1}$$

The goal is to obtain the corresponding set of binary spins given a coupling matrix J_{ij} and a magnetic field $\vec{h}$, such that the Hamiltonian is minimized. This is equivalent to the natural state of the system. Classical computing is based on boolean logic with values $[0,1]$. In ONNs, phases of the oscillators encode these binary values ($[0°, 180°]$). The evolution of oscillators' phases θ_i can be expressed with the following ordinary differential equation[6]:

$$\dot{\theta}_i = \omega + V(\theta_i)\sum_{j=1}^{n} W_{ij}\,V\left(\theta_j - \frac{\pi}{2}\right) \tag{2}$$

Natural frequency ω and coupling $W_{i,j}$ play a crucial role in the ODE. Moreover, the 2π-periodic function $V(\theta_i)$ describes the phase response curve (PRC) of the system. This depends on the choice of the neuron model, with the Kuramoto model being one of the most well-known ones[4]. Since the PRC of Kuramoto is a sinusoidal wave and using the phase deviations ϕ_i ($\theta_i(t) = \omega t + \phi_i$) is more convenient, Eq. 2 can be reformulated as:

$$\dot{\phi}_i = \frac{1}{2}\sum_{j=1}^{n} W_{ij}\sin(\phi_j - \phi_i) \tag{3}$$

This system has the following Lyapunov function[2,7]:

$$E = -\frac{1}{2}\sum_{i,j} W_{i,j}\cos(\phi_i - \phi_j) \tag{4}$$

By mapping phases $[0,\pi]$ to $[-1,+1]$, the energy function is proportional to the Ising Hamiltonian. The second oscillator model, the van der Pol oscillator, is represented as[5]:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0 \tag{5}$$

Contrary to the Kuramoto model, the PRC of the van der Pol oscillator is not derived analytically, but rather numerically using the adjoint method[8]. Since the phases do not generally settle at binary values, one applies the so-called sub-harmonic injection locking (SHIL) scaled by a factor K_s to nudge the system towards such states:

$$\dot{\phi}_i = V(\phi_i)\sum_{j=1}^{n} W_{ij}\,V\left(\phi_j - \frac{\pi}{2}\right) - K_s V(2\phi_i) \tag{6}$$

Max-cut problem. The max-cut problem is finding two complementary subgraphs in a graph of N vertices by maximizing the number of cut edges. It is one of the NP-hard problems, i.e. the runtime explodes as N increases[9]. GW is the best algorithm yielding 87.8% of the best cut[1]. There exists a mapping from max-cut to the Ising model by choosing $J_{i,j}$ to be the negative edge weights[2,10]. To obtain the max-cut with OIMs means to solve Eq. 6.

Results. First, we vary the edge density from 10% up to 90%. Moreover, for each density, we create 100 random graphs. As the oscillator phases are randomly initialized, we solve max-cut for each graph 10 times each using all three algorithms i.e., Kuramoto-ONN, vdP-ONN and GW. Fig. 1 (left and middle) summarizes the results of the unweighted (uniform edge weights) and weighted (with random edge weights between 0 and 1) 14-nodes graphs.

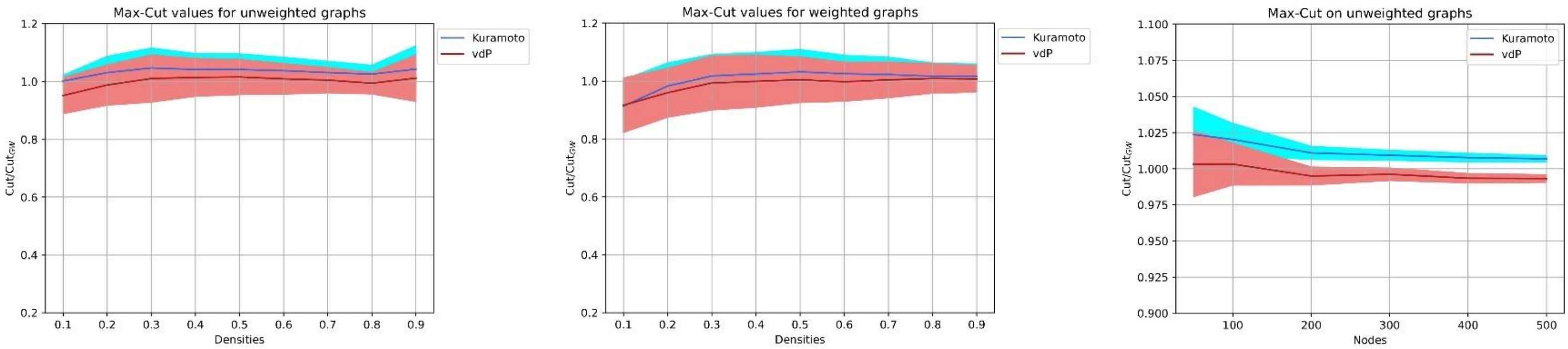

Figure 1: Left: max-cut as a function of densities, unweighted graphs; middle: max-cut as a function of densities, weighted graphs; right: max-cut as a function of number of nodes, unweighted graphs

Both algorithms provide a cut quality as good as GW and for certain densities the quality of cut with Kuramoto ONN is better than GW. Interestingly with the increase of edge density both Kuramoto and van der Pol ONN provide good results. For both models with low edge densities the performance is slightly lower than GW. However, both algorithms still produce competitive results with Kuramoto being the superior one.

We also investigate the scalability of the algorithms with up to 500 nodes graph for a fixed density in an unweighted graph. The results presented in Fig. 1 (right) have been computed using 20 different graphs 10 times. As the number of nodes increases, the performance of both models slightly worsens, though they still remain competitive, especially Kuramoto.

Conclusions. In this paper, we presented a few benchmarks on the max-cut problem using Kuramoto and van der Pol ONNs. This has been done by deriving the phase ODE of the systems and letting it evolve. The results of both models have been compared to the GW algorithm producing competitive results, with Kuramoto being the superior one. Ongoing investigations are focused on other benchmarks and identifying the interplay of problem mapping and neuron model for solving optimization problems with ONNs.

[1] M.X. Goemans, J. Williamson, ACM (1995), 42 (6), 1115–1145.
[2] T. Wang, et al., 2019. https://arxiv.org/abs/1903.07163.
[3] E. Ising, Zeit. Fur Physik 31. (1925).
[4] Y. Kuramoto, Springer Verlag (1984), 8, 156.
[5] B. van der Pol, Selected scientific papers (1960).
[6] F.C. Hoppensteadt et al., IEEE Transactions on NNs (2000), 11 (3), 734–738.
[7] C. Delacour et al., NNPC (2023).
[8] E. Brown, et al, Neural Computation (2004), 16 (4), 673–715.
[9] A. Lucas, Frontiers in physics (2014), 2, 5.
[10] R.M. Karp, Springer US (1972), pp 85–103.