Local-Minima-Preserving Polynomial Relaxation of Ising Problems

D Banerjee   Santanu Mahapatra   Kunal N. Chaudhury

Indian Institute of Science, Bengaluru

ICML 2026

Motivation

Many NP-hard problems reduce to the Ising problem:

$$\min_{\boldsymbol{s}\in\{-1,1\}^n} \mathcal{E}(\boldsymbol{s}) = -\frac{1}{2}\boldsymbol{s}^\top \mathbf{J}\boldsymbol{s}$$

Examples:

- MAX-CUT
- Number Partitioning Problem (NPP)
- Spin glasse models

Challenge:

- Discrete local search is problem-specific and not scalable
- First-order continuous methods are GPU-friendly and highly scalable
- Existing relaxations introduce spurious minima and looses information about the discrete landscape

Goal: Design a *continuous relaxation* whose local minima exactly correspond to discrete one-flip local minima.

Key Idea: Continuous Relaxation with Attractors

We optimize over the λ-scaled hypercube: $\boldsymbol{x} \in [-\lambda, \lambda]^n$, instead of discrete spins: $\boldsymbol{s} \in \{-1, 1\}^n$.

Hamiltonian:

$$\mathcal{H}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^\top \mathbf{J}\boldsymbol{x} + \mathcal{A}(\boldsymbol{x})$$

where

$$\mathcal{A}(\boldsymbol{x}) = \sum_{i=1}^{n} f_\theta(x_i)$$

Continuous formulation:

$$\min_{\boldsymbol{x} \in [-\lambda, \lambda]^n} \mathcal{H}(\boldsymbol{x})$$

We denote the set of all continuous local minima as $\text{loc}(\mathcal{H})$:

$$\text{loc}(\mathcal{H}) = \left\{\boldsymbol{x}^* \in [-\lambda, \lambda]^n : \exists \epsilon > 0, \quad \mathcal{H}(\boldsymbol{x}^*) \leqslant \mathcal{H}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in [-\lambda, \lambda]^n, \|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < \epsilon\right\}$$

Discrete One-Flip Local Minima

A spin state $\boldsymbol{s}^*$ is a one-flip local minimum if:

$$\mathcal{E}(\boldsymbol{s}^*) \leqslant \mathcal{E}(\boldsymbol{s})$$

for all configurations differing in at most one spin (Hamming distance: $d_H(\boldsymbol{s}, \boldsymbol{s}^*) \leqslant 1$). Equivalent fixed-point condition:

$$s_i^*(\mathbf{J}\boldsymbol{s}^*)_i \geqslant 0 \implies s_i^* = \text{sign}((\mathbf{J}\boldsymbol{s}^*)_i)$$

Synchronization score:

$$\mathrm{sync}(\boldsymbol{s}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{s_i = \text{sign}((\mathbf{J}\boldsymbol{s})_i)\}}$$

- $\mathrm{sync} = 1$: valid one-flip local minimum
- Define set of one-flip local minima: $\text{loc}(\mathcal{E}) := \{\boldsymbol{s} \in \{-1, 1\}^n : \mathrm{sync}(\boldsymbol{s}) = 1\}$.

Main question:
Can local minima of $\mathcal{H}(\boldsymbol{x})$ recover exactly these discrete stable states?

Landscape Equivalence Theorem

Admissible attractors satisfy:

$$\underbrace{f_\theta(x) = f_\theta(-x)}_{\textbf{Symmetry}}, \quad \underbrace{f_\theta''(x) < 0}_{\textbf{Concavity}}, \quad \underbrace{f_\theta'(\lambda) < 0}_{\textbf{Boundary Attraction}}, \quad \underbrace{f_\theta'(\lambda) + \lambda\gamma > 0}_{\textbf{Robustness}}.$$

where γ is the discrete stability margin.

Main Theorem

Landscape Equivalence

If the attractor is admissible, then

$$\boldsymbol{x}^* \in \mathrm{loc}(\mathcal{H}) \iff \mathrm{sign}(\boldsymbol{x}^*) \in \mathrm{loc}(\mathcal{E})$$

Consequences:

- Every continuous local minimum corresponds to a discrete one-flip minimum
- No spurious local minima on the interior $(-\lambda, \lambda)^n$
- Enables scalable first-order optimization

MiP-CRIM Solver

MiP-CRIM: Minima-**P**reserving **C**ontinuous **R**elaxation of **I**sing **M**odel

Optimization pipeline:

1. Given $\mathbf{J}, \mathcal{H} = \mathcal{E} + \mathcal{A}$, initialize $\boldsymbol{x}^{(0)}$
2. Optimize $\mathcal{H}(\boldsymbol{x})$ using ADAM: $\boldsymbol{x} \leftarrow \text{Opt}\big(\boldsymbol{x}, \nabla\mathcal{H}(\boldsymbol{x})\big)$
3. Project to box $[-\lambda, \lambda]^n$: $\boldsymbol{x} \leftarrow \text{Proj}_{[-\lambda,\lambda]^n}(\boldsymbol{x})$
4. Map to spins: $\boldsymbol{s} = \text{sign}(\boldsymbol{x})$
5. Verify one-flip optimality: $\boldsymbol{s} \odot (\mathbf{J}\boldsymbol{s}) \geqslant \mathbf{0}$
6. If not optimal set: $\boldsymbol{x}^{(0)} \leftarrow \mathcal{N}[\boldsymbol{x}, \sigma\mathbf{I}]$

Important properties

- GPU-friendly matrix-vector operations
- Saddle points are unstable
- Basin-hopping improves exploration

Experimental Results: Main Benchmarks

Method	SK Model		MAX-CUT		NPP	
	Energy ↓	sync ↑	Cut ↑	sync ↑	Disc ↓	sync ↑
GW-SDP	-15053	0.937	32018	0.945	1.41	0.489
D-WAVE	-16593	**1.000**	33775	**1.000**	0.0096	0.997
FEM	**-16814**	**1.000**	34038	**1.000**	313.8	0.132
MiP-CRIM	-16689	**1.000**	**34137**	**1.000**	**0.0015**	**1.000**

Observation

- MiP-CRIM achieves state-of-the-art performance across diverse NP-hard tasks
- Perfect synchronization ($\text{sync} = 1$) on almost all runs
- Competitive runtime with GPU-friendly first-order optimization

Experimental Results: Comparison with IAMP

GOE and SK spin-glass benchmarks (10^3-spins)

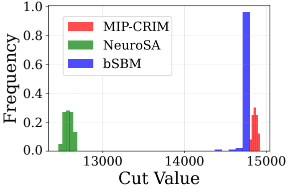
Model	Solver	Avg Energy	sync	Runtime
GOE-1000	IAMP	-643.85	0.943	0.179s
	MiP-CRIM	**-728.27**	**1.000**	**0.137s**
SK-1000	IAMP	-10895.32	0.870	0.144s
	MiP-CRIM	**-16269.86**	**1.000**	**0.138s**

Main takeaway

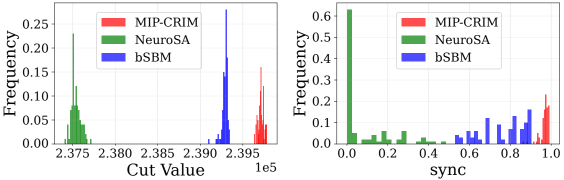
- MiP-CRIM consistently achieves:
  - lower Ising energy
  - exact one-flip synchronization ($\mathrm{sync} = 1$)
- IAMP provides approximate solutions ($\mathrm{sync} < 1$)
- The landscape-equivalence theorem translates continuous local minima directly into discrete one-flip stable states.

Experimental Results: MAX-CUT on ER Graphs

100 independent runs on Erdős–Rényi graphs



Sparse graph: $p = 0.05$



Dense graph: $p = 0.95$

Observations

- MiP-CRIM consistently achieves higher cut values
- Synchronization remains concentrated near:

$$\text{sync} = 1$$

- Robust across sparse and dense graph regimes

Takeaways

Main Contributions

- Introduced attractor-based continuous (polynomial) relaxation
- Proved landscape (local-minima) equivalence theorem
- Developed scalable differentiable solver: MiP-CRIM

Main Take-way

Efficient continuous optimization can be applied to hard combinatorial problems that preserve discrete local optimality when the relaxation landscape is designed appropriately.

Thank You!

Code: `https://github.com/Phymath0Masics/MiP-CRIM`