Unbiased generation of metastable states for 2D Ising spin glasses

Petri Savola

Laboratory for Theoretical Computer Science, TKK

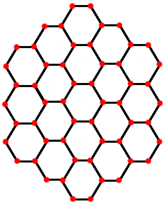
12.10.2007

Overview

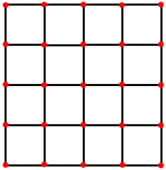
- Basics of spin glass lattices
- Examples
- Generating the metastable states
- Estimating the number of local minima
- Sampling the energy distribution

Basics

Two example 2D lattices



(a) Hex lattice



(b) Square lattice

- Nodes are spins: $\sigma_i \in \{-1, 1\}$
- Edges are interactions: $J_{ij} \in \{-1, 0, 1\}$
- Hamiltonian: $H(\sigma) = -\frac{1}{2} \sum_{ij}^{N} J_{ij}\sigma_i\sigma_j$

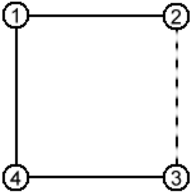
Note: $J_{ij} \geq 0$ for a ferromagnetic lattice

Basics

$$J = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

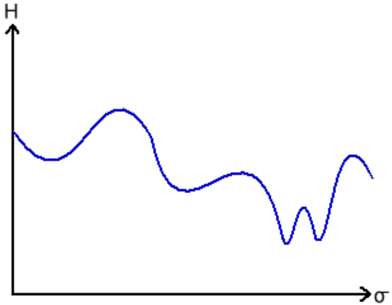
Note: $H =$ (# of UNSAT edges) − (# of SAT edges)

Frustration: cycle with odd number of edges with weight -1 $\Rightarrow$ odd number of UNSAT edges



Basics

- Local minimum: no energy decrease by a single spin flip
- Usually wanted: ground state, $min(H(\sigma))$
  - Hard
- Greedy algorithms: get stuck in local minima
- Ground state(s) hard to prove



Example

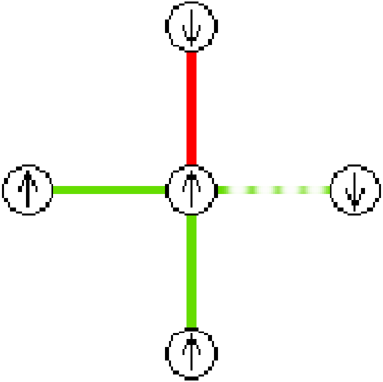


Figure: Is this a metastable state?

Example

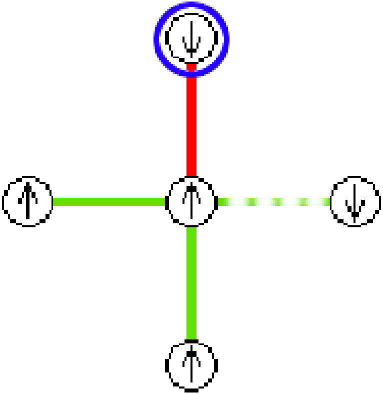


Figure: No, because this spin could be flipped.

Example

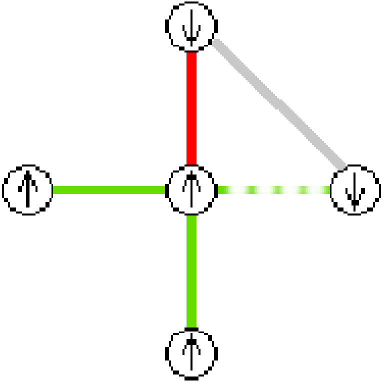


Figure: Would this new edge be satisfied?

Example

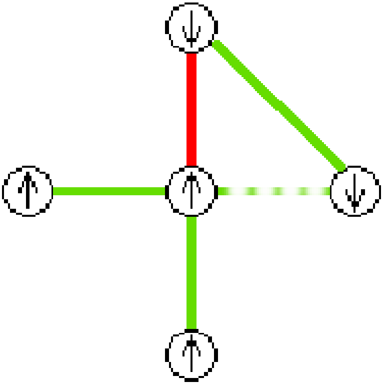


Figure: Yes, spins are the same in both ends.

Example

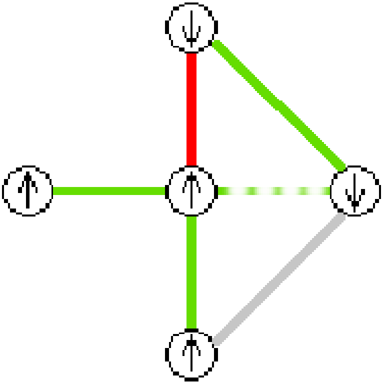


Figure: What about this edge?

Example

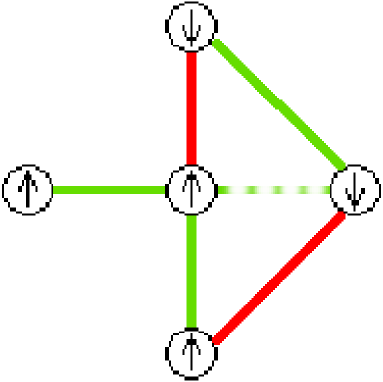
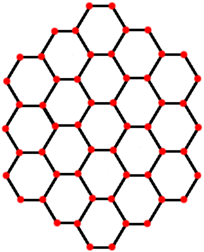
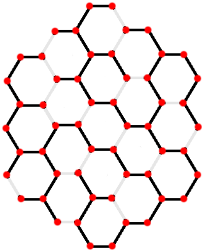


Figure: No, spins are not the same.

Spanning tree example



(a) Graph



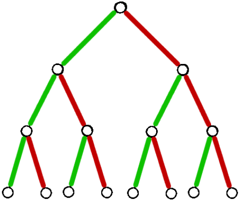
(b) A spanning tree

- Every node connected
- No cycles

Requirements

A way to find local minima:

- Choose a spanning tree
- Enumerate the spanning tree edges and give them color
  - Binary tree
- Frustration ⇒ rest of the edge colors are determined

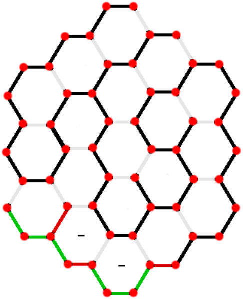


Coloring is valid for a metastable state if:

- σ is a valid state
- More satisfied than unsatisfied edges leaving from all spins

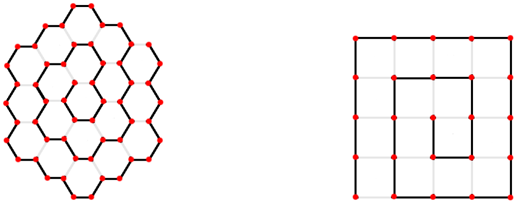
Generation of the spanning path

Choosing just any spanning tree causes problems:



Generation of the spanning path

But spiral-like spanning path seems to work:



Generation of the search tree

- Cut off non-interesting branches
- Each leaf represents an unique local minimum
- $H(\sigma)$ can be calculated

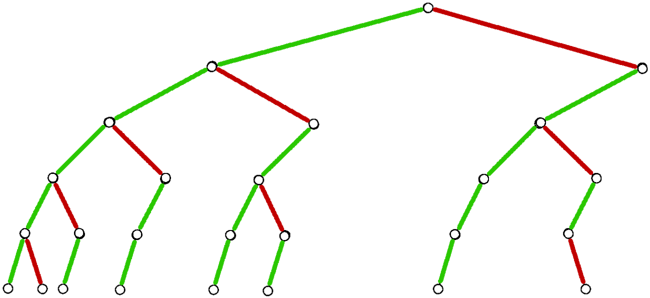


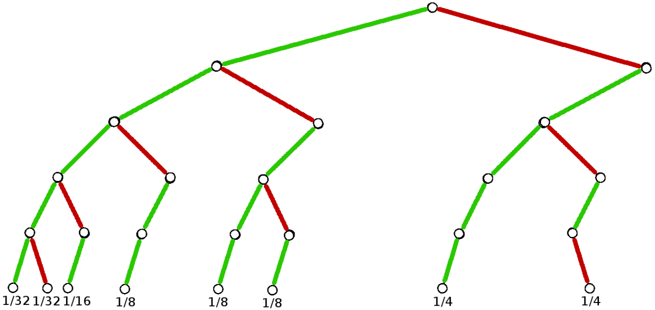
Figure: Search tree for a graph with one hex and an odd number of negative edges

Search tree size approximation

Goal: calculate search tree size

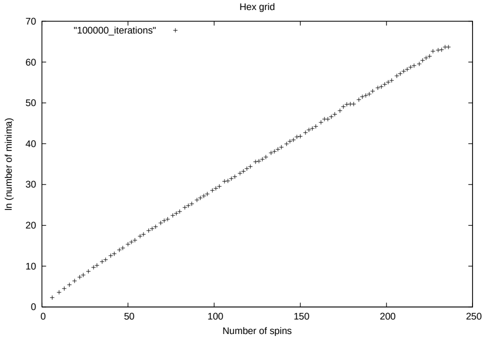
Knuth's method: Tree size is $a_1 a_2 ... a_n$, where a_i is the branching factor of node at level i (for binary tree $a_i \in \{1, 2\}$) and route from the root to a leaf $(1, 2, ..., n)$ is chosen randomly.

- Unbiased estimate
- Search tree somewhat balanced
  - Knuth's method works well



Results

Hypothesis: $N = e^{\alpha|V|}$, where N is the number of local minima, α is a constant which depends on the system and $|V|$ is the number of spins/nodes/vertices in the lattice.



- Honeycomb lattice: $\alpha \approx 0.231$ (2000 spins)
- Ferromagnetic honeycomb lattice: $\alpha \approx 0.226$ (2000 spins)

Energy distribution sampling

- Recursive use of Knuth's method
- Size of right branch: s(R)
- Size of left branch: s(L)
- Probability of choosing right branch: $\frac{s(R)}{s(R)+s(L)}$
- Each leaf is obtained with equal probability

Energy distribution results

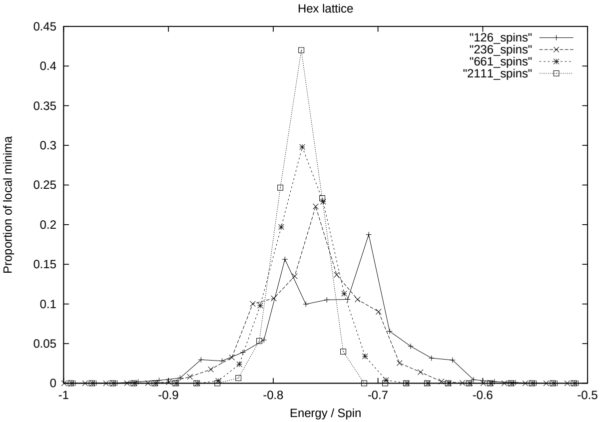


Figure: Ferromagnetic hex lattice

Energy distribution results

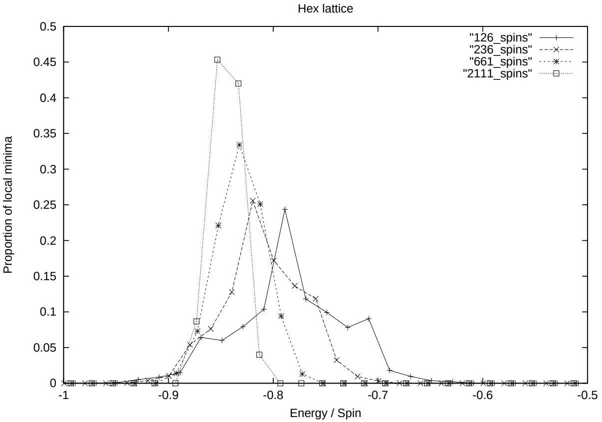


Figure: Spin glass hex lattice

Summary

- Spanning path method is an interesting new idea which probably could be useful
- There is an exponential number of local minima so finding a global minimum must be hard by using simple local heuristics or greedy search
- Typical energy of a local minimum is not very close to the energy value of the global minimum