# Sampling and finding ground states of spin glass models using parallel tempering, simulated annealing, and tensor networks

Kylie Le*, supervised by Stephen Zhang, Yong See Foo, and Michael Stumpf
University of Melbourne
* thuylinh1@student.unimelb.edu.au

## Introduction

- Consider a set $\Lambda$ of spins with state $\sigma = \{\sigma_k\}_{k\in\Lambda}$ where $\sigma_k \in \{-1, 1\}$. An interaction $J_{ij}$ exists between any two spins $i, j \in \Lambda$ and an external magnetic field $h_i$ interacts with any spin $i \in \Lambda$. The spin glass model is described by the Hamiltonian function

$$H(\sigma) = -\sum_{ij} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$$

- The significance arises from its NP-hard nature, closely tied to challenging combinatorial optimization tasks.
- To solve the sampling problem, we rely on Monte Carlo methods like single-spin flip and parallel tempering.
- Along with simulated annealing, we also explore tensor network - an exact algorithm for finding stationary states of lowest energy, or ground states.

## Single-spin flip Monte Carlo

- Design a Markov chain Monte Carlo (MCMC) random walker to sample according to Boltzmann distribution

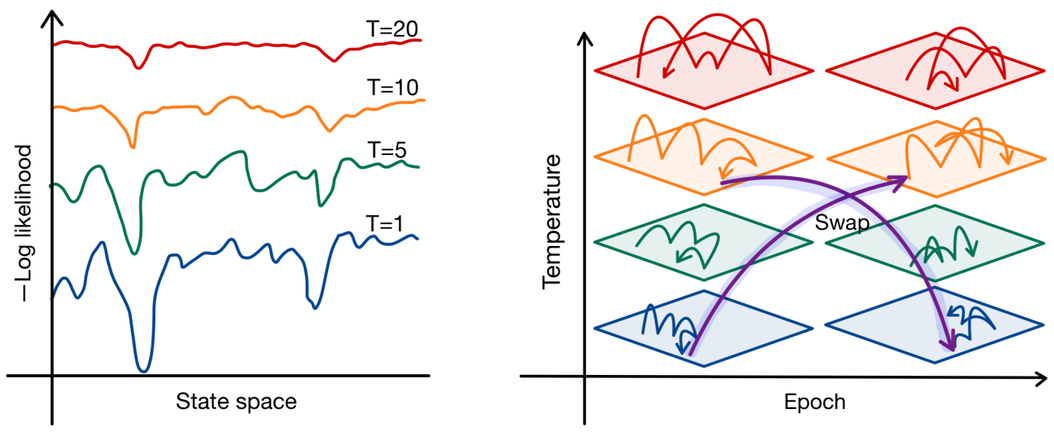
$$\pi(\sigma, T) = \frac{exp\left(-\frac{H(\sigma)}{T}\right)}{\sum_\sigma exp\left(-\frac{H(\sigma)}{T}\right)} \qquad \text{where } T = 1.$$

- Update using Metropolis algorithm: propose a new state $\hat{\sigma}$ by flipping a random spin of the current state $\sigma$ and accept $\hat{\sigma}$ with probability

$$p_{flip} = 1 \wedge \exp\left(-\frac{H(\hat{\sigma}) - H(\sigma)}{T}\right)$$

## Parallel Tempering

- Embed the actual PDF $\pi(\sigma, 1)$ to a higher dimensional sampling space $\pi(\sigma, T)$, where the prominence of local and global minima is controlled by temperature T.
- Distribute an ensemble of MCMC random walkers across all temperature levels $T_i$ (i = 1, . . . , n).




- Within-chain steps: Metropolis updates.
- Between-chain steps: exchange swaps between random pairs $T_i$ and $T_j$ with acceptance probability

$$p_{swap} = 1 \wedge \exp\left((H(\sigma_i) - H(\sigma_j))\left(\frac{1}{T_i} - \frac{1}{T_j}\right)\right)$$

## Sampling Result

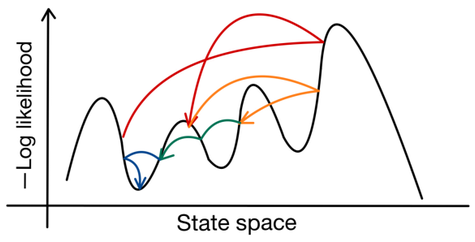
- Generate 50 sets of random J, h, and initial state $\sigma$.
- Run single-spin flip with $10^5$ steps and parallel tempering with $5x10^3$ steps over 20 chains $(T = 1, \ldots, 20)$, both with burn-in and sample interval of 100 steps.

Table 1: Total variation distance

| | # spins | | | | |
|---|---|---|---|---|---|
| Algorithm | 5 | 10 | 15 | 20 | 25 |
| Single-spin flip | 0.450 | 0.551 | 0.450 | 0.661 | 0.740 |
| Parallel tempering | 0.012 | 0.012 | 0.008 | 0.018 | 0.028 |

## Simulated Annealing

- Independent MCMC random walkers start at a high T (exploratory search) and slowly move to lower T (localized search) according to the cooling schedule $T(t) = T_0\alpha^t$ where $0 < \alpha < 1$.
- Perform Metropolis updates at each T to keep the system close to equilibrium.

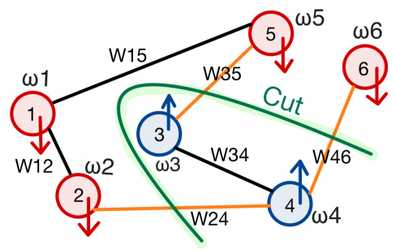



## Tensor Networks

- Reduce to the cutting problem with graph $G = (V, E)$ and energy model

$$H_c(G) = \sum_{(i,j)\in E} W_{ij}\big((1-n_i)n_j + (1-n_j)n_i\big) - \sum_{i\in V}\omega_i n_i$$

where $n_i = \frac{1-\sigma_i}{2}$ is the partition index in the cutting problem, $\omega_i = -2h_i$ is vertex weight, and $W_{ij} = 2J_{ij}$ is edge weight.




- Solve the cutting problem using tensor network contraction.

## Ground States Result

- Generate 60 sets of random J and h.
- Run simulated annealing with $T_0 = 20$ and $\alpha = 0.9$ over 20 chains, record the proportion of chains that correctly identified ground energy given by tensor network.

Table 2: Proportion of accurate chains

| | # spins | | | | | |
|---|---|---|---|---|---|---|
| Algorithm | 5 | 10 | 15 | 20 | 25 | 30 |
| Simulated annealing | 1.00 | 0.98 | 0.79 | 0.61 | 0.57 | 0.37 |

Table 3: Running time (in seconds)

| | # spins | | | | | |
|---|---|---|---|---|---|---|
| Algorithm | 5 | 10 | 15 | 20 | 25 | 30 |
| Simulated annealing | 0.06 | 0.08 | 0.10 | 0.11 | 0.17 | 0.21 |
| Tensor network | 0.00 | 0.04 | 0.12 | 0.48 | 2.46 | 30.01 |

## Further Extension

Use approximate tensor network contractions to propose collective (all-spins) updates, which mitigate the locality problem in Metropolis (single-spin) updates.

## References

1. Malcolm Sambridge, A Parallel Tempering algorithm for probabilistic sampling and multimodal optimization, Geophysical Journal International, Volume 196, Issue 1, January 2014, Pages 357–374, https://doi.org/10.1093/gji/ggt342
2. Liu, J.-G., Gao, X., Cain, M., Lukin, M.D. and Wang, S.-T. (2023). Computing Solution Space Properties of Combinatorial Optimization Problems Via Generic Tensor Networks. SIAM Journal on Scientific Computing, 45(3), pp.A1239–A1270. doi:https://doi.org/10.1137/22m1501787.