Deterministic Policy Gradient for RL-Based MaxCut on Sparse Ising Graphs

COMP 579

Sarah Ameur Fadi Younes Habiba Abdelrehim

School of Computer Science, McGill University

Outline

Motivation

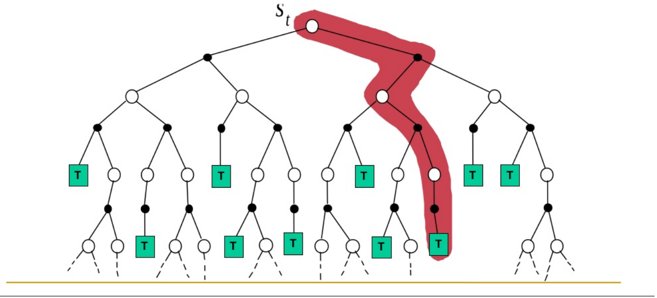
- **MaxCut:** partition vertices of $G = (V, E)$ to maximize crossing edges.
- Ising model: find spin configuration minimizing energy H.
- NP-hard: 2^n node configs
- Sparse graphs ($|E| \ll n^2$) yield shallow minima—random flips give little feedback.
- RL: Policy NN can learn a spin flipping. Avoids searching.

$$H(\boldsymbol{\sigma}) = -\sum_{(i,j)\in E} J_{ij}\,\sigma_i\,\sigma_j,$$

$$\mathrm{Cut}(\boldsymbol{\sigma}) = \frac{1}{2}\sum_{(i,j)\in E}\Big(1 - \sigma_i\sigma_j\Big).$$

Deterministic REINFORCE

- Policy: MLP with n inputs, two ReLU hidden layers of size n, n sigmoid outputs.
- Deterministic actions: threshold probabilities at 0.5.
- Reward $r_t = H(\sigma_t) - H(\sigma_{t+1})$. Gradient ascent on expected return.
- **New:** Metropolis–Hastings burn-in at each trajectory start to escape local minima.



Deterministic REINFORCE

Algorithm 1 Adapted deterministic REINFORCE

Input: epochs M, learning rate α, chains N, trajectory length T, spin count n

Initialize policy network π_θ with input dimension n, two hidden ReLU layers of size n, and Sigmoid output of size n

for epoch $1 \to M$ **do**

  Sample initial configurations $\{\sigma_0^i\}_{i=1}^N \subset \{\pm 1\}^n$

  Perform Metropolis burn-in: spin flips accepted with probability $\propto \exp(-\Delta E/T)$

  for $t = 0 \to T-1$ and $i = 1 \to N$ **do**

    Compute $p_\theta(\cdot \mid \sigma_t^i)$ via forward pass

    Determine σ_{t+1}^i by thresholding p_θ

    Apply greedy local search $LS(\sigma_{t+1}^i)$ to reduce energy

    Compute reward $r_t^i = H(\sigma_t^i) - H(\sigma_{t+1}^i)$

  end for

  Compute returns $R_t^i = \sum_{t'=t}^{T-1} r_{t'}^i$

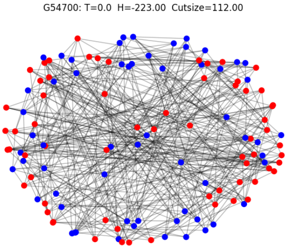
  Update $\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_{i=1}^N \sum_{t=0}^{T-1} R_t^i \nabla_\theta \log \pi_\theta(\sigma_{t+1}^i \mid \sigma_t^i)$

end for

Output: best spin configuration

Benchmarks

- Compare adapted REINFORCE to:
    - Classical Metropolis–Hastings (local stochastic flips).
    - Pure greedy local search (flip if energy decreases).



- 10 dense graphs ($n = 125, |E| = 750$)
- Achieved 89–96% of Lu & Liu [2023] in 10–30 min (vs. 60 min)

- 7 sparse graphs ($n = 800$–10^4, $|E| \approx 0.1\%$–1.5%).
- Achieved 77–89% of original MaxCuts ($+$ limited runtime)
- Lowest relative performance: G70 (10'000 nodes).

Conclusion & Future Work

- **Future:** deterministic actor–critic
    - Value network + Policy network, to reduce gradient variance
    - Helps guide policy

References

Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*.

Khalil, E. B., Dai, H., Zhang, Y., Dilkina, B., and Song, L. (2017). Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pages 6348–6358.

Lu, Y. and Liu, X.-Y. (2023). Reinforcement Learning for Ising Model. In *Machine Learning and the Physical Sciences Workshop at NeurIPS 2023*. https://ml4physicalsciences.github.io/2023/files/NeurIPS_ML4PS_2023_248.pdf.

GRAFO Research Group (2025). Optsicom Benchmark Instances. University Rey Juan Carlos and University of Valencia. https://grafo.etsii.urjc.es/optsicom/#instances. Accessed: 2025-03-02.

Ye, Y. (2025). GSet Graph Collection. Stanford University. https://web.stanford.edu/~yyye/yyye/Gset/. Accessed: 2025-03-03.