Reinforcement Learning for Ising Model

Yicheng Lu[1]*, **Xiao-Yang Liu**[1]
[1]Columbia University
yichen98@mail.ustc.edu.cn, XL2427@columbia.edu

Abstract

The Ising spin glasses model [2], a fundamental model in statistical mechanics and condensed matter physics, providing insights into the behavior of interacting spins within a physical system, has been studied for many decades [11, 5, 13], while also being able to formulate many NP-hard problems [10]. However, solving the Ising model for large and complex systems is computationally demanding and often infeasible using traditional methods. In this paper, we present the deterministic REINFORCE algorithm tailored for the Ising model, enabling state-of-the-art performance through learned state transition policies. In our work, we first formulate the Ising model with MaxCut problem as a case study. Secondly, we propose a novel deterministic REINFORCE algorithm incorporating the Local Search approach. Finally, we evaluate our algorithm on well-known datasets and demonstrate state-of-the-art performance.

1 Introduction

Motivation. The Ising spin glasses model [2, 11, 13], a fundamental model in statistical mechanics, provides valuable insights into the behavior of interacting spins within physical systems. It has been shown that many NP-hard problems, including all of Karp's 21 NP-complete problems [7] (such as the MaxCut problem, 3-SAT problem and the graph coloring problem), have Ising formulations [10]. Leveraging these insights, we adapt related methodologies to address Ising Model challenges, offering a promising avenue for more efficient solutions.

Challenges. Prior methods, while valuable, face hurdles in handling the vast search space of large-scale systems. Balancing accuracy with efficiency is a persistent issue. This underscores the importance of leveraging reinforcement learning's robust space-searching capabilities. This dual exploitation-exploration approach holds great potential for addressing the complexities of the Ising Model effectively.

In this paper, we propose a novel scheme for solving the Ising Model—the deterministic REINFORCE algorithm, achieving state-of-the-art performance. Our contributions are threefold: Firstly, We have eliminated the dynamic aspects from the environment. Secondly, we introduce the deterministic REINFORCE algorithm, integrating the Local Search trick, thereby addressing the vast search space of large-scale systems. Finally, we have employed large-scale parallel sampling, conduct comprehensive evaluations on well-established datasets, consistently demonstrating state-of-the-art performance and validating the efficacy of our proposed scheme.

*Yichen Lu completed this work as a research assistant at Columbia University.

Machine Learning and the Physical Sciences Workshop at 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Problem Formulation

2.1 Ising Spin Glasses Model

The Ising model provides a framework for understanding a system of spin glasses, where each spin, denoted as σ_i, can be in one of two states: $\sigma_i \in \{+1, -1\}$. These spins interact with their neighbors, and this interaction is quantified by the coupling coefficient J_{ij}. It's important to note that spins that are spatially distant have negligible interaction, allowing us to consider J_{ij} as effectively zero for non-neighboring spins. In this model, each spin σ_i is also influenced by an external magnetic field h_i. The energy of a spin configuration $s = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ is defined by the Hamiltonian function:

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

The goal in the Ising model is typically to find the ground state, or the configuration of spins that minimizes the Hamiltonian.

2.2 Graph MaxCut Problem

Given an undirected graph $G = (V, E)$ with n vertices, the goal of MaxCut Problem is to partition the vertices into two sets S_1 and S_2, such that the number of edges with one endpoint in S_1 and the other in S_2 is maximized. Fig. 1 illustrates a graph consisting of five vertices and six edges. A possible maximum cut is indicated by the dashed line, which divides the vertices into two subsets: set S_1 includes the filled vertices, while set S_2 comprises the unfilled vertices. The edges that form the maximum cut are highlighted in blue, totaling a maximum cut-size of five. Note that this graph may have alternative configurations that also result in a maximum cut.

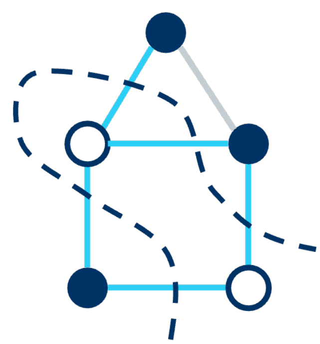

Figure 1: A Graph MaxCut Example

Equivalently, if we associate each node V_i with a binary value $\sigma_i \in \{-1, +1\}$, assigning $+1$ to nodes in set S_1, and -1 to nodes in set S_2 as follows:

$$\sigma_i = \begin{cases} +1 & \text{if } V_i \in S_1, \\ -1 & \text{if } V_i \in S_2. \end{cases} \tag{2}$$

To represent the entire partition of vertices in the MaxCut problem, we use a binary vector of length n, $s = \{\sigma_1, \sigma_2, ..., \sigma_n\}$. This vector allows us to express the cut-size of the graph Cut(s) as:

$$\text{Cut}(s) = \frac{1}{2}\sum_{(i,j)\in E}(1 - \sigma_i\sigma_j). \tag{3}$$

We can express the Hamiltonian of the Ising model in a way that mirrors the objective of the MaxCut problem. the objective is to find the ground state that minimizes the Hamiltonian $H(s) = -\text{Cut}(s)$, which is equivalent to maximize the cut-size of the graph. To simplify the Hamiltonian for computational convenience and align it more closely with the typical Ising model format, we can remove any redundant coefficients and constant terms. The simplified Hamiltonian becomes:

$$H(s) = \sum_{(i,j)\in E} \sigma_i\sigma_j. \tag{4}$$

In this form, the Hamiltonian directly reflects the objective of the MaxCut problem within the Ising model framework. Finding the ground state that minimizes $H(s)$ effectively solves the MaxCut problem by maximizing the number of cut edges

3 Reinforcement Learning Algorithm

3.1 Deterministic REINFORCE Algorithm

REINFORCE is a policy gradient algorithm in reinforcement learning. It directly adjusts policy parameters to maximize expected rewards. It samples trajectories, computes gradients from rewards, and updates the policy via stochastic gradient ascent. It enables on-policy learning without the need for a value function. The original REINFORCE algorithm [14] is considering the case of stochastic and parameterized policy π_θ. We denote the policy with the parameter θ by $\pi_\theta(\cdot|s_t)$. The log-probability of a trajectory $\tau = \{s_0, a_0, r_1, s_1, a_1, ..., s_{T-1}, a_{T-1}, r_T, s_T\}$ is:

$$\log P(\tau|\theta) = \log \rho_0(s_0) + \sum_{t=0}^{T} \left[\log P(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t)\right]. \tag{5}$$

Here, we remove the dynamics of the environment, in another word, $P(s_{t+1}|s_t, a_t) = 1$. The state transition is a deterministic process when the decision is made. the Log-Probability of a Trajectory is:

$$\log P(\tau|\theta) = \log \rho_0(s_0) + \sum_{t=0}^{T} \log \pi_\theta(a_t|s_t). \tag{6}$$

since the action is deterministic, we use the policy network to generate the transition probability to the next state s_{t+1} directly from the the current state s_t, that is, $\pi_\theta(s_{t+1}|s_t)$

$$\log P(\tau|\theta) = \log \rho_0(s_0) + \sum_{t=0}^{T} \log \pi_\theta(s_{t+1}|s_t). \tag{7}$$

In this context, we consider the distribution on each single variable and assume independence, akin to the mean field (MF) approximation in statistical physics. By overlooking certain interdependencies between random variables, the parameterized joint transition probability $\pi_\theta(s|\cdot)$ is simplified, with each component of s treated as independent:

$$\log \pi_\theta(s|\cdot) = \sum_{k=1}^{n} \log p_\theta(\sigma_k|\cdot), \tag{8}$$

we would like to optimize the policy π_θ by gradient ascent $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$, where $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}[R(\tau)]$ is the expected return, and we take $R(\tau)$ to give the finite-horizon undiscounted return $R(\tau) = \sum_{t=1}^{T} r_t$. In order to get a effective evaluation of the transition, we firstly implement the local search over the state $\widehat{s} = \mathbf{LS}(s)$, and use the decrease of Hamiltonian of the state after local search $r_{t+1} = H(\widehat{s}_t) - H(\widehat{s}_{t+1})$ as the Reward. The policy π_θ introduces a trajectory distribution, the gradient term works out to:

$$\nabla_\theta J(\pi_\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{t=0}^{T} \left(H(\widehat{s}_t) - H(\widehat{s}_{t+1})\right) \cdot \sum_{t=0}^{T} \nabla_\theta \left(\sum_{k=1}^{n} \log p_\theta(\sigma^i_{k,t+1}|s^i_t) \right) \right]. \tag{9}$$

Alg. 1 gives the pseudocode of the proposed algorithm.

3.2 Implementation Tricks

Trick 1: Massively parallel simulation on GPUs. We utilized the Markov Chain Monte Carlo (MCMC) method to achieve massively parallel simulation on GPUs. This approach significantly enhances computational efficiency by leveraging the parallel processing capabilities of modern GPU architectures.

Trick 2: Local search for computing advantage function. We integrate a filter scheme that incorporates a local search technique [3], expanding the scope of our function landscape. This method systematically explores the vicinity of the current solution, selecting the most promising neighboring option. This refinement improves binary solutions and addresses challenges in discrete spaces.

4 Performance Evaluations

4.1 Results for Ising Spin Glasses Model

Table 1 provides a comprehensive overview of the results obtained for the Ising Spin Glasses Model across various instances. The table includes the CPLEX Best Bound (the best solution found by

Algorithm 1 Deterministic REINFORCE

Input : Number of epochs τ, learning rate α, number of trajectories N, trajectory length T, number of nodes n

Training: policy network π_θ
for $j = 1, 2, ..., J$ **do**
  for $i = 1, 2, ..., N$ **do**
    Generate a random starting binary vector $s_0^i \in \{0, 1\}^n$
    for $t = 1, 2, ..., T$ **do**
      Obtain n probabilities $\{p_\theta(\sigma_{1,t+1}^i | s_t^i), p_\theta(\sigma_{2,t+1}^i | s_t^i), ..., p_\theta(\sigma_{n,t+1}^i | s_t^i)\}$ by performing a forward pass over the policy network $\pi_\theta(s_{t+1}^i | s_t^i)$
      Sample the next state s_{t+1}^i based on the obtained probabilities
      Perform local search $\widehat{s}_{t+1}^i = \mathbf{LS}(s_{t+1}^i)$
      Compute the reward $r_{t+1}^i = H(\widehat{s}_t^i) - H(\widehat{s}_{t+1}^i)$
    end
    Compute the accumulative return $R_t^i = \sum_{t'=t}^{T} r_{t'+1}^i$
  end
  Compute policy gradient: $\nabla_\theta J(\theta) = \frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{t=0}^{T} R_t^i \nabla_\theta \log \pi_\theta(s_{t+1} | s_t)$
  Update policy parameters: $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end

Output : the best found solution s^*, and corresponding Hamiltonian $H(s^*)$

Table 1: Results for Ising spin glasses model.

Instances	CPLEX Best Bound	Deterministic REINFORCE (**Ours**)	S2V+DQN [8]	MaxcutApprox [9]	SDP [6]
G54100	110	**110**	108	80	54
G54200	112	**112**	108	90	58
G54300	106	**106**	104	86	60
G54400	114	**114**	108	96	56
G54500	112	**112**	112	94	56
G54600	110	**110**	110	88	66
G54700	112	**112**	108	88	60
G54800	108	**108**	108	76	54
G54900	110	**110**	108	88	68
G541000	112	**112**	108	80	54

the CPLEX 12.6.1 optimizer within a 1-hour time limit), as well as the performance of different algorithms, including Deterministic REINFORCE (marked as **Ours**), S2V+DQN [8], MaxcutApprox [9], and SDP [6]. It is evident that our Deterministic REINFORCE algorithm consistently matches the optimal solutions for all instances, demonstrating its effectiveness and reliability. This performance indicates that our approach is highly competitive and capable of handling complex instances of the Ising spin glasses model.

4.2 Results for Graph MaxCut

In Table 2, we present the MaxCut results obtained using different algorithms, including Deterministic REINFORCE (Ours), BLS [1], DSDP [4], and PI-GNN [12]. Notably, our approach consistently outperforms the comparison algorithms across all instances. Specifically, for instances G55 and G70, Deterministic REINFORCE achieves the best results, surpassing the performance of BLS, DSDP and PI-GNN.

Table 2: Results for MaxCut.

Instances	Nodes	Edges	Deterministic REINFORCE (**Ours**)	BLS [1]	DSDP [4]	PI-GNN [12]
G14	800	4694	**3064**	**3064**	2922	3026
G15	800	4661	**3050**	**3050**	2938	2990
G22	2000	19990	**13359**	**13359**	12960	13181
G49	3000	6000	**6000**	**6000**	**6000**	5918
G50	3000	6000	**5880**	**5880**	**5880**	5820
G55	5000	12468	**10298**	10294	9960	10138
G70	10000	9999	**9583**	9541	9456	9421

5 Conclusions

In this study, we provide a new insight of reinforcement learning to solve the Ising Spin Glasses and combinatorial optimization problem. To overcome the challenge of a vast search space, we introduce a deterministic variant of the REINFORCE algorithm, integrating learned state transition policies with local search strategies. Our case study on Ising Spin Glasses and the MaxCut Problem consistently demonstrates state-of-the-art performance.

References

[1] Una Benlic and Jin-Kao Hao. Breakout local search for the max-cut problem. *Engineering Applications of Artificial Intelligence*, 26(3):1162–1173, 2013.

[2] Kurt Binder and A Peter Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics*, 58(4):801, 1986.

[3] Cheng Chen, Ruitao Chen, Tianyou Li, Ruichen Ao, and Zaiwen Wen. Monte carlo policy gradient method for binary optimization. *arXiv preprint arXiv:2307.00783*, 2023.

[4] Changhui Choi and Yinyu Ye. Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver. *Manuscript, Department of Management Sciences, University of Iowa, Iowa City, IA*, 52242, 2000.

[5] Arthur E Ferdinand and Michael E Fisher. Bounded and inhomogeneous ising models. i. specific-heat anomaly of a finite lattice. *Physical Review*, 185(2):832, 1969.

[6] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[7] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

[8] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

[9] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

[10] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

[11] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

[12] Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4(4):367–377, 2022.

[13] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792, 1975.

[14] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.