Annealed Training for Combinatorial Optimization on Graphs

Haoran Sun Etash K. Guha Hanjun Dai
presenter: Shen Yuan

中国人民大學 RENMIN UNIVERSITY OF CHINA 高瓴人工智能学院 Gaoling School of Artificial Intelligence

- Introduction
- Annealed Training for Combinatorial Optimization
- Case Study
- Experiments
- Summary

Introduction

This paper proposed an annealed training framework for Combinatorial Optimization (CO) problems.

Introduction

This paper proposed an annealed training framework for Combinatorial Optimization (CO) problems.

1. The CO problems are transformed into unbiased energy-based models (EBMs).

Introduction

This paper proposed an annealed training framework for Combinatorial Optimization (CO) problems.

1. The CO problems are transformed into unbiased energy-based models (EBMs).
2. Then graph neural networks are trained to approximate the EBMs.

Introduction

This paper proposed an annealed training framework for Combinatorial Optimization (CO) problems.

1. The CO problems are transformed into unbiased energy-based models (EBMs).
2. Then graph neural networks are trained to approximate the EBMs.
3. To prevent the training from being stuck at local optima near the initialization, an annealed loss function are introduced.

- Introduction
- Annealed Training for Combinatorial Optimization
- Case Study
- Experiments
- Summary

Annealed Training for Combinatorial Optimization

1. The CO problems are transformed into unbiased energy-based models (EBMs).
2. Then graph neural networks are trained to approximate the EBMs.
3. To prevent the training from being stuck at local optima near the initialization, an annealed loss function are introduced.

What are the Combinatorial Optimization (CO) problems?

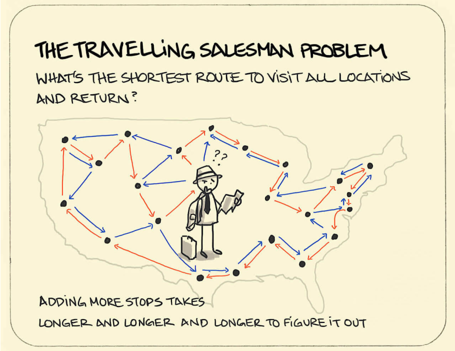


Figure 1: The Travelling salesman problem

What are the Combinatorial Optimization (CO) problems?

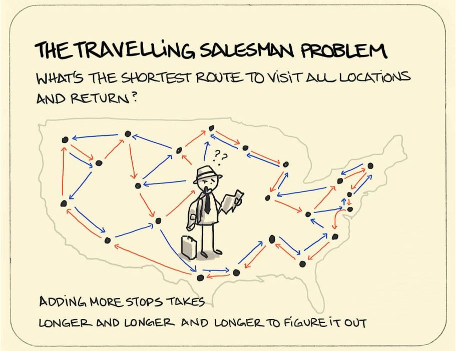


Figure 1: The Travelling salesman problem

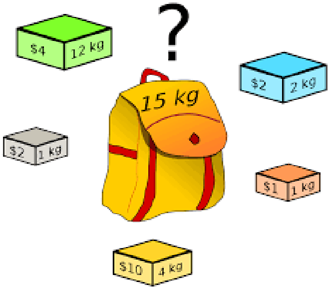


Figure 2: The Knapsack problem

What are the Combinatorial Optimization (CO) problems?

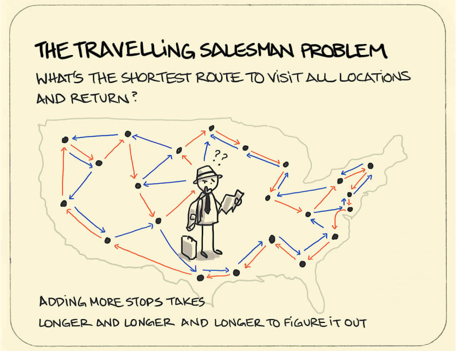


Figure 1: The Travelling salesman problem

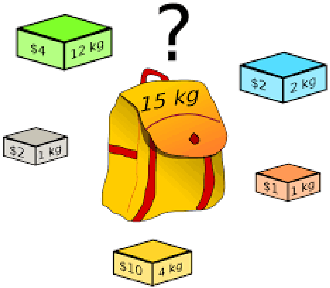


Figure 2: The Knapsack problem

Combinatorial Optimization is the process of searching for maxima (or minima) of an objective function F whose domain is a discrete but large configuration space.

Energy Based Model

We denote the set of combinatorial optimization problems as $\mathcal{I}$. An instance $I \in \mathcal{I}$ is

$$I = (c(\cdot), \{\psi_i\}_{i=1}^{m}) := \arg\min_{\boldsymbol{x} \in \{0,1\}^n} c(\boldsymbol{x}) \quad \text{s.t. } \psi_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, m \tag{1}$$

where $c(\cdot)$ is the objective function we want to minimize and $\psi_i \in \{0, 1\}$ indicates whether the i-th constraint is satisfied or not.

Energy Based Model

We denote the set of combinatorial optimization problems as $\mathcal{I}$. An instance $I \in \mathcal{I}$ is

$$I = (c(\cdot), \{\psi_i\}_{i=1}^{m}) := \arg\min_{\boldsymbol{x} \in \{0,1\}^n} c(\boldsymbol{x}) \quad \text{s.t. } \psi_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, m \tag{1}$$

where $c(\cdot)$ is the objective function we want to minimize and $\psi_i \in \{0, 1\}$ indicates whether the i-th constraint is satisfied or not.
We could rewrite the constrained problem into an equivalent unconstrained form via big M method:

$$\arg\min_{\boldsymbol{x} \in \{0,1\}^n} f^{(I)}(\boldsymbol{x}) := c(\boldsymbol{x}) + \sum_{i=1}^{m} \beta_i \psi_i(\boldsymbol{x}), \quad \beta_i \geq 0 \tag{2}$$

Energy Based Model

Using unbiased $f^{(I)}$ to measure the fitness of a solution x, we can define the unbiased energy-based models (EBMs):

$$P_{\tau}^{(I)}(\boldsymbol{x}) \propto e^{-f^{(I)}(\boldsymbol{x})/\tau} \tag{3}$$

Energy Based Model

Using unbiased $f^{(I)}$ to measure the fitness of a solution x, we can define the unbiased energy-based models (EBMs):

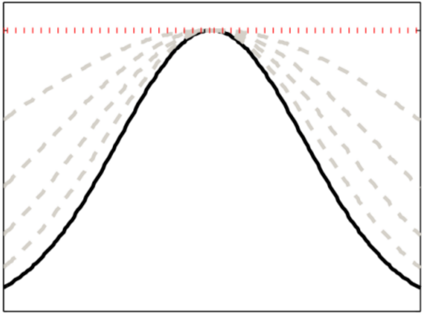
$$P_{\tau}^{(I)}(\boldsymbol{x}) \propto e^{-f^{(I)}(\boldsymbol{x})/\tau} \tag{3}$$

The temperature τ can be used to control the smoothness of the distribution.

Energy Based Model

$$P_\tau^{(I)}(\boldsymbol{x}) \propto e^{-f^{(I)}(\boldsymbol{x})/\tau}$$

- When $\tau \to +\inf$, the EBM P_τ converges to a uniform distribution over the whole state space $\{0, 1\}$;
- When $\tau \to 0$, the EBM P_τ converges to a uniform distribution over the optimal solutions.



Annealed Training for Combinatorial Optimization

1. The CO problems are transformed into unbiased energy-based models (EBMs).
2. Then graph neural networks are trained to approximate the EBMs.
3. To prevent the training from being stuck at local optima near the initialization, an annealed loss function are introduced.

Tempered Loss and Parameterization

Given an instance $I \in \mathcal{I}$, $G_\theta(I) = \phi$ generates a vector ϕ that determines a variational distribution Q_ϕ to approximate the target distribution $P_\tau^{(I)}$.

[1] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. "Combinatorial optimization with graph convolutional networks and guided tree search". In: *Advances in neural information processing systems* 31 (2018).

[2] Hanjun Dai et al. "A Framework For Differentiable Discovery Of Graph Algorithms". In: (2020).

Tempered Loss and Parameterization

Given an instance $I \in \mathcal{I}$, $G_\theta(I) = \phi$ generates a vector ϕ that determines a variational distribution Q_ϕ to approximate the target distribution $P_\tau^{(I)}$.
In this work, we consider the variational distribution as a product distribution:

$$Q_\phi(x) = \prod_{i=1}^{n} (1 - \phi_i)^{1-x_i} \phi_i^{x_i} \tag{4}$$

Such a form is a popular choice in learning graphical neural networks for combinatorial optimization[1][2]

[1] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. "Combinatorial optimization with graph convolutional networks and guided tree search". In: *Advances in neural information processing systems* 31 (2018).

[2] Hanjun Dai et al. "A Framework For Differentiable Discovery Of Graph Algorithms". In: (2020).

Tempered Loss and Parameterization

We want to minimize the KL-divergence:

$$\begin{aligned} D_{\mathrm{KL}}(Q_\phi \| P_\tau^{(I)}) &= \int Q_\phi(\boldsymbol{x}) \left(\log Q_\phi(\boldsymbol{x}) - \log \frac{e^{-f^{(I)}(\boldsymbol{x})/\tau}}{\sum_{\boldsymbol{x} \in \{0,1\}^n} e^{-f^{(I)}(\boldsymbol{x})/\tau}} \right) \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{\tau} \mathbb{E}_{\boldsymbol{x} \sim Q_\phi(\cdot)} [f^{(I)}(\boldsymbol{x})] - H(Q_\phi) + \log \sum_{\boldsymbol{x}\{0,1\}^n} e^{-f^{(I)}(\boldsymbol{x})/\tau} \end{aligned} \tag{5}$$

Tempered Loss and Parameterization

We want to minimize the KL-divergence:

$$
\begin{aligned}
D_{\mathrm{KL}}(Q_\phi \| P_\tau^{(I)}) &= \int Q_\phi(\boldsymbol{x}) \left(\log Q_\phi(\boldsymbol{x}) - \log \frac{e^{-f^{(I)}(\boldsymbol{x})/\tau}}{\sum_{\boldsymbol{x}\in\{0,1\}^n} e^{-f^{(I)}(\boldsymbol{x})/\tau}} \right) \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{\tau} \mathbb{E}_{\boldsymbol{x}\sim Q_\phi(\cdot)}[f^{(I)}(\boldsymbol{x})] - H(Q_\phi) + \log \sum_{\boldsymbol{x}\{0,1\}^n} e^{-f^{(I)}(\boldsymbol{x})/\tau}
\end{aligned} \tag{5}
$$

Remove the terms not involving ϕ and multiply the constant τ, the annealed loss functions for ϕ and τ can be defined as:

$$
\begin{aligned}
L_\tau(\phi, I) &= \mathbb{E}_{\boldsymbol{x}\sim Q_\phi(\cdot)}[f^{(I)}(\boldsymbol{x})] - \tau H(Q_\phi) \\
L_\tau(\theta) &= \mathbb{E}_{I\sim\mathcal{I}} \left[\mathbb{E}_{\boldsymbol{x}\sim Q_{G_\theta}(\cdot)}[f^{(I)}(\boldsymbol{x})] - \tau H(Q_{G_\theta(I)}) \right]
\end{aligned} \tag{6}
$$

Annealed Training for Combinatorial Optimization

1. The CO problems are transformed into unbiased energy-based models (EBMs).
2. Then graph neural networks are trained to approximate the EBMs.
3. To prevent the training from being stuck at local optima near the initialization, an annealed loss function are introduced.

What is Simulated Annealing?

Simulated Annealing is a stochastic global search optimization algorithm.

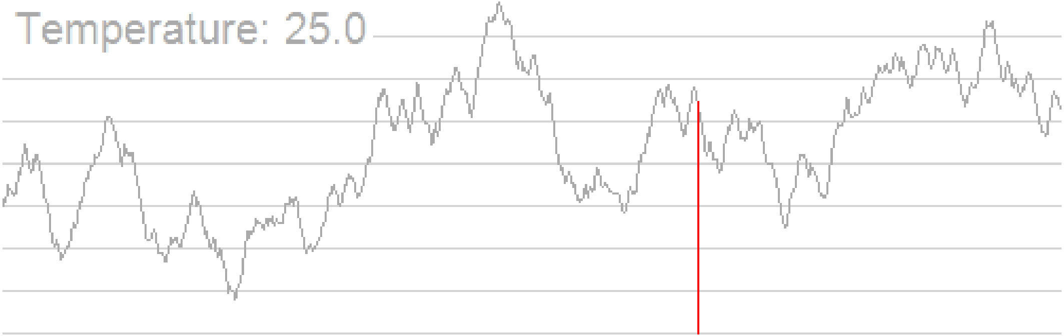


Figure 3: The Simulated Annealing process

Annealed Training

To address the non-convexity in training, we employ an annealed training. In particular, we use a large initial temperature τ_0 to smooth the loss function and reduce τ_t gradually to zero during training.

$$\tau_k = \frac{\tau_0}{1 + \alpha k} \tag{7}$$

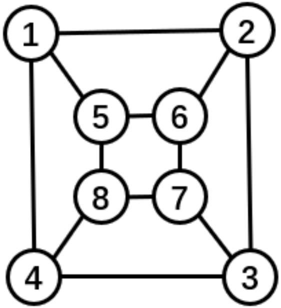
- Introduction
- Annealed Training for Combinatorial Optimization
- Case Study
- Experiments
- Summary

Maximum Independent Set

An independent set is a subset of the vertices $S \subseteq V$, such that for arbitrary $i, j \in S, (i, j) \notin E$.

Maximum Independent Set

An independent set is a subset of the vertices $S \subseteq V$, such that for arbitrary $i, j \in S, (i, j) \notin E$.



Maximum Independent Set

An independent set is a subset of the vertices $S \subseteq V$, such that for arbitrary $i, j \in S, (i, j) \notin E$.
The maximum independent set problem is to find an independent set S having the largest weight. If we denote $x_i = 1$ to indicate $i \in S$ and $x_i = 0$ to indicate $i \notin S$, the problem can be formulated as:

$$\mathop{\arg\min}_{x \in \{0,1\}^n} c(x) := -\sum_{i=1}^{n} w_i x_i, \quad \text{s.t. } x_i x_j = 0, \forall (i,j) \in E \tag{8}$$

We define the corresponding energy function:

$$f(x) := -\sum_{i=1}^{n} w_i x_i + \sum_{(i,j) \in E} \beta_{ij} x_i x_j \tag{9}$$

Proof - Maximum Independent Set

$$f(x) := -\sum_{i=1}^{n} w_i x_i + \sum_{(i,j)\in E} \beta_{ij} x_i x_j \tag{10}$$

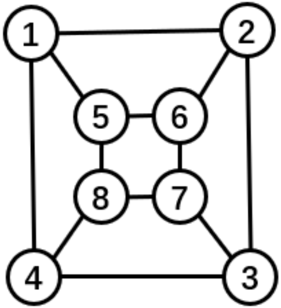
Proposition A.1. If $\beta_{ij} \geq \min\{w_i, w_j\}$ for all $(i,j) \in E$, then for any $x \in \{0,1\}^n$, there exists a $x' \in \{0,1\}^n$ that satisfies the constraints in Eqn.8 and has lower energy: $f(x') \leq f(x)$.

Maximum Clique

A clique is a subset of the vertices $S \subseteq V$, such that every two distinct $i, j \in S$ are adjacent:$(i, j) \in E$.

Maximum Clique

A clique is a subset of the vertices $S \subseteq V$, such that every two distinct $i, j \in S$ are adjacent: $(i, j) \in E$.



Maximum Clique

A clique is a subset of the vertices $S \subseteq V$, such that every two distinct $i, j \in S$ are adjacent:$(i,j) \in E$.
The maximum clique problem is to find a clique S having the largest weight. If we denote $x_i = 1$ to indicate $i \in S$ and $x_i = 0$ to indicate $i \notin S$, the problem can be formulated as:

$$\arg\min_{x \in \{0,1\}^n} c(x) := -\sum_{i=1}^{n} w_i x_i, \quad \text{s.t. } x_i x_j = 0, \forall (i,j) \in E^c \tag{11}$$

where $E^c = \{(i,j) \in V \times V : i \neq j, (i,j) \notin E\}$ is the set of complement edges on graph G.
We define the corresponding energy function:

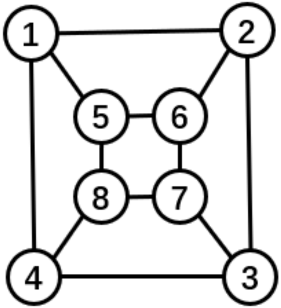
$$f(x) := -\sum_{i=1}^{n} w_i x_i + \sum_{(i,j) \in E^c} \beta_{ij} x_i x_j \tag{12}$$

Minimum Dominate Set

A dominate set is a subset of the vertices $S \subseteq V$, where for any $v \in V$, there exists $u \in S$ such that $(u, v) \in E$ or $u = v$.

Minimum Dominate Set

A dominate set is a subset of the vertices $S \subseteq V$, where for any $v \in V$, there exists $u \in S$ such that $(u, v) \in E$ or $u = v$.



Minimum Dominate Set

A dominate set is a subset of the vertices $S \subseteq V$, where for any $v \in V$, there exists $u \in S$ such that $(u, v) \in E$ or $u = v$.
The minimum dominate set problem is to find a dominate set S having the minimum weight. If we denote $x_i = 1$ to indicate $i \in S$ and $x_i = 0$ to indicate $i \notin S$, the problem can be formulated as:

$$\arg\min_{x \in \{0,1\}^n} c(x) := \sum_{i=1}^{n} w_i x_i, \quad \text{s.t. } (1 - x_i) \prod_{j \in N(i)} (1 - x_j) = 0, \forall i \in V \tag{13}$$

We define the corresponding energy function:

$$f(x) := \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \beta_i (1 - x_i) \prod_{j \in N(i)} (1 - x_j) \tag{14}$$

Proof - Minimum Dominate Set

$$f(x) := \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \beta_i (1 - x_i) \prod_{j \in N(i)} (1 - x_j) \tag{15}$$

Proposition A.3. If $\beta_i \geq \min\{w_k : k \in N(i) \; or \; k = i\}$, then for any $x \in \{0, 1\}^n$, there exists a $x' \in \{0, 1\}^n$ that satisfies the constraints in Eqn.13 and has lower energy: $f(x') \leq f(x)$

Experiments

Table 1: Evaluation of Maximum Independent Set

Size	small		large		Collab		Twitter	
Method	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)
Erdos	0.805 ± 0.052	0.156	0.781 ± 0.644	2.158	0.986 ± 0.056	0.010	0.975 ± 0.033	0.020
Our's	$\mathbf{0.898} \pm 0.030$	0.165	$\mathbf{0.848} \pm 0.529$	2.045	$\mathbf{0.997} \pm 0.020$	0.010	$\mathbf{0.986} \pm 0.012$	0.020
Greedy	0.761 ± 0.058	0.002	0.720 ± 0.046	0.009	0.996 ± 0.017	0.001	0.957 ± 0.037	0.006
MFA	0.784 ± 0.058	0.042	0.747 ± 0.056	0.637	0.998 ± 0.007	0.002	0.994 ± 0.010	0.003
RUNCSP	0.823 ± 0.145	1.936	0.587 ± 0.312	7.282	0.912 ± 0.101	0.254	0.845 ± 0.184	4.429
G(0.5s)	0.864 ± 0.169	0.723	0.632 ± 0.176	1.199	1.000 ± 0.000	0.029	0.950 ± 0.191	0.441
G(1.0s)	0.972 ± 0.065	1.063	0.635 ± 0.176	1.686	1.000 ± 0.000	0.029	1.000 ± 0.000	0.462

Table 2: Evaluation of Maximum Clique

Size	small		large		Collab		Twitter	
Method	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)
Erdos	0.813 ± 0.067	0.279	0.735 ± 0.084	0.622	0.960 ± 0.019	0.139	0.822 ± 0.085	0.222
Our's	$\mathbf{0.901} \pm 0.055$	0.262	$\mathbf{0.831} \pm 0.078$	0.594	$\mathbf{0.988} \pm 0.011$	0.143	$\mathbf{0.920} \pm 0.083$	0.213
Greedy	0.764 ± 0.064	0.002	0.727 ± 0.038	0.014	0.999 ± 0.002	0.001	0.959 ± 0.034	0.001
MFA	0.804 ± 0.064	0.144	0.710 ± 0.045	0.147	1.000 ± 0.000	0.005	0.994 ± 0.010	0.010
RUNCSP	0.821 ± 0.131	2.045	0.574 ± 0.299	7.332	0.887 ± 0.134	0.164	0.832 ± 0.153	4.373
G(0.5s)	0.948 ± 0.076	0.599	0.812 ± 0.087	0.617	0.997 ± 0.035	0.061	0.976 ± 0.065	0.382
G(1.0s)	0.984 ± 0.042	0.705	0.847 ± 0.101	1.077	0.999 ± 0.015	0.062	0.997 ± 0.029	0.464

Experiments

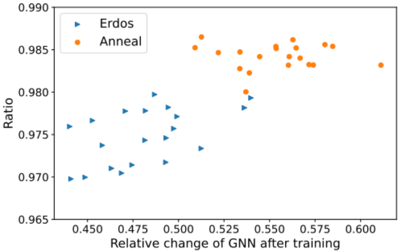
Table 3: Evaluation of Minimum Dominate Set

Size	small		large		Collab		Twitter	
Method	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)
Erdos	0.909 ± 0.037	0.121	0.889 ± 0.017	0.449	0.982 ± 0.070	0.007	0.924 ± 0.098	0.015
Our's	$\mathbf{0.954} \pm 0.006$	0.120	$\mathbf{0.931} \pm 0.015$	0.453	$\mathbf{0.993} \pm 0.062$	0.006	$\mathbf{0.952} \pm 0.074$	0.016
Greedy	0.743 ± 0.053	0.254	0.735 ± 0.026	3.130	0.661 ± 0.406	0.028	0.741 ± 0.142	0.079
MFA	0.926 ± 0.032	0.213	0.910 ± 0.016	3.520	0.895 ± 0.210	0.030	0.952 ± 0.076	0.099
G(0.5s)	0.993 ± 0.014	0.381	0.994 ± 0.013	0.384	1.000 ± 0.000	0.042	1.000 ± 0.000	0.084
G(1.0s)	0.999 ± 0.005	0.538	0.999 ± 0.005	0.563	1.000 ± 0.000	0.042	0.839 ± 0.000	0.084

Table 4: Evaluation of Minimum Cut

Size	SF-295		Facebook		Twitter	
Method	*ratio*	time (s)	*ratio*	time (s)	*ratio*	time (s)
Erdos	$\mathbf{0.124} \pm 0.001$	0.22	0.156 ± 0.026	289.3	0.292 ± 0.009	6.17
Our's	0.135 ± 0.011	0.23	$\mathbf{0.151} \pm 0.045$	290.5	$\mathbf{0.201} \pm 0.007$	6.16
L1 GNN	0.188 ± 0.045	0.02	0.571 ± 0.191	13.83	0.318 ± 0.077	0.53
L2 GNN	0.149 ± 0.038	0.01	0.305 ± 0.082	13.83	0.388 ± 0.074	0.53
Pagerank-Nibble	0.375 ± 0.001	1.48	N/A	N/A	0.603 ± 0.005	20.62
CRD	0.364 ± 0.001	0.03	0.301 ± 0.097	596.46	0.502 ± 0.020	20.35
MQI	0.659 ± 0.000	0.03	0.935 ± 0.024	408.52	0.887 ± 0.007	0.71
Simple-Local	0.650 ± 0.024	0.05	0.961 ± 0.019	1787.79	0.895 ± 0.006	0.84
G(10s)	0.105 ± 0.000	0.16	0.961 ± 0.010	1787.79	0.535 ± 0.006	52.98

Parameter Change Distance



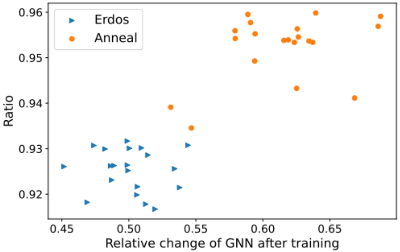


Figure 2: Distance in MIS

Figure 3: Distance in MDS

The relative change is calculated as $\frac{\|u-v\|_2}{\|v\|_2}$, where v and u are vectors flattened from the parameters of GNN before and after training.

Ablation Study

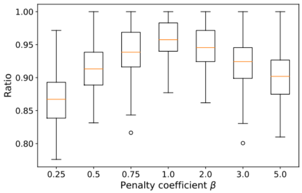


Figure 4: Ablation for β

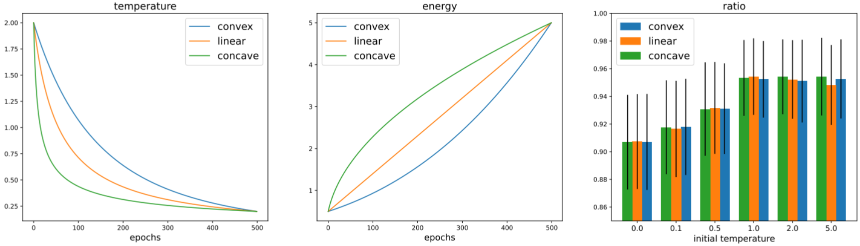


Figure 5: Ablation for annealing schedule

- Introduction
- Annealed Training for Combinatorial Optimization
- Case Study
- Experiments
- Summary

Summary

- This paper proposed an annealed training framework for Combinatorial Optimization problems.

Summary

- This paper proposed an annealed training framework for Combinatorial Optimization problems.
- The temperature τ needs to be set manually, maybe it can be calculated automatically with some mechanism, or added noise.

Summary

- This paper proposed an annealed training framework for Combinatorial Optimization problems.
- The temperature τ needs to be set manually, maybe it can be calculated automatically with some mechanism, or added noise.
- This paper is poorly written...