# Optimizing Solution-Samplers for Combinatorial Problems: The Landscape of Policy-Gradient Methods

NeurIPS'23

Constantine Caramanis, Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, Christos Tzamos

Stanford CS/MS&E 331

# Gradient descent for combinatorial opt

- **Common pipeline** (e.g., several papers from this quarter):
  1. Train NN whose parameters define a **distribution** over solutions
  2. Optimize expected cost via **gradients**
- Works well, but **lacks theory** explaining *why*
- **Challenges:**
  - Naïve **exponential-size** simplexes guarantee convexity
  - But compact parameterizations often have **bad landscapes**
- Q: Design polynomial-size, well-behaved distribution families where gradient descent **provably** finds near-optimal solutions?
- This paper: **yes**, with applications to several CO problems

# Setup

- Instances $I \in \mathcal{I}$ with common solution space $S$
  - Prior distribution $\mathcal{R}$ over train/test inputs
- Cost function $L(\cdot\,;\,I)\colon S \to \mathbb{R}$ (assume efficient to evaluate)
  - E.g., for max cut: $L(s\,;\,I) = -(\text{cut weight})$
  - Results hold even for **blackbox/oracle access** to $L$
- Distribution $p(\cdot\,; I, \boldsymbol{w})$ over solutions with trainable $\boldsymbol{w} \in \mathcal{W}$
- Loss function $\mathcal{L}(\boldsymbol{w}) = \mathbb{E}_{I\sim\mathcal{R}, s\sim p(\cdot\,;I,\boldsymbol{w})}[L(s\,;I)]$
- **Goal:** find $\boldsymbol{w}$ so that $\mathcal{L}(\boldsymbol{w})$ is close to opt $= \mathbb{E}_{I\sim\mathcal{R}}\left[\min_{s\in S} L(s\,;I)\right]$

# Gradient descent dynamics

- **Algorithm is fixed:** gradient descent
  - Many other algorithms you could imagine (see, e.g., Remark 7)
  - Choose gradient descent because it's very common
- **At each iteration ($t$):**
  1. Sample an instance $I \sim R$ (or a minibatch)
  2. Sample solutions $s_1, \ldots, s_m \sim p(\cdot\,; I, \boldsymbol{w}_t)$
  3. Query costs $c_k = L(I, s_k)$
  4. Form a policy-gradient estimate $\boldsymbol{g}_t$ and update $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \eta_t \boldsymbol{g}_t$

**This paper:** Form of the distribution

$$\boldsymbol{g}_t \approx \frac{1}{m} \sum_{k=1}^{m} c_k \nabla_{\boldsymbol{w}} \log p(s_k\,; I, \boldsymbol{w}_t)$$

**This paper:** How to regularize loss

# Key desiderata

- Distribution over all $(s, I)$ needs exponentially many params
- Neural parameterizations are **compressed representations**
  - $[\mathcal{I}]$ = # bits to represent instance
- **Three desiderata:**
  - **Complete:** some $\bar{\boldsymbol{w}}$ achieves $\mathcal{L}(\bar{\boldsymbol{w}}) \leq \text{opt} + \epsilon$
  - **Compressed:** Description size of $\mathcal{W}$ is $[\mathcal{W}] = \text{poly}\left([\mathcal{I}], \frac{1}{\epsilon}\right)$
  - **Efficiency:** first-order methods reach $2\epsilon$-opt loss in $\text{poly}\left([\mathcal{W}], \frac{1}{\epsilon}\right)$ steps
- Does **not** imply P=NP: sampling may still be hard
  - In practice, **approximate samplers** often work well

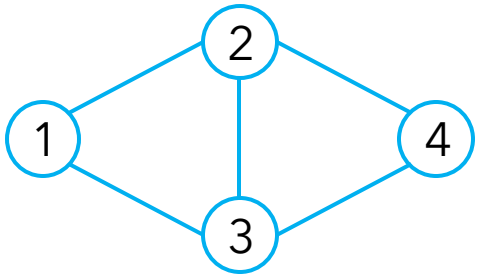
# Key assumptions

Feature mappings $\psi_S(s) \in \mathbb{R}^{n_X}, \psi_\mathcal{I}(I) \in \mathbb{R}^{n_Z}$

1. *Boundedness:* $\|\psi_S(s)\|_2 \leq D_S, \|\psi_\mathcal{I}(I)\|_2 \leq D_\mathcal{I}$ for all $s, I$
2. *Bilinear cost* structure $L(s, I) = \psi_S(s)^\top M \psi_\mathcal{I}(I)$ with $\|M\|_F \leq C$
   - Example in future slides
   - $M$ is unknown (results hold even for blackbox/oracle access to $L$)
3. *Variance preserves features* when $s \sim \mathrm{Unif}(S)$
   - For all $\boldsymbol{v} \in \mathbb{R}^{n_X}$, $\mathrm{Var}_{s \sim \mathrm{Unif}(S)}[\boldsymbol{v} \cdot \psi_S(s)] \geq \alpha \|\boldsymbol{v}\|_2$
   - Ensures gradients of optimization objective are not vanishing
4. *Polynomial scaling* $n_X, n_Z, D_S, D_\mathcal{I}, C \leq \mathrm{poly}([\mathcal{I}]), \alpha \geq 1/\mathrm{poly}([\mathcal{I}])$

# Example: Max-cut

Diagonal degree matrix: $D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$




Adjacency matrix: $A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

Laplacian: $L_G = D - A = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}$

# Example: Max-cut

- For $n$-node graph, max-cut equivalent to: $\max_{\boldsymbol{s}\in\{\pm 1\}^n} \frac{1}{4}\boldsymbol{s}^\top L_G \boldsymbol{s}$
- For any $\boldsymbol{s} \in \{\pm 1\}^n$, $\psi_S(\boldsymbol{s}) = (\boldsymbol{s}\boldsymbol{s}^\top)^\flat \in \mathbb{R}^{n^2}$

Flatten to $\mathbb{R}^{n^2}$

- $\psi_{\mathcal{I}}(G) = {L_G}^\flat \in \mathbb{R}^{n^2}$

1. *Boundedness*: $\|\psi_S(\boldsymbol{s})\|_2, \|\psi_{\mathcal{I}}(G)\|_2 \leq \mathrm{poly}([\mathcal{I}]) = \mathrm{poly}(n)$
2. *Bilinear cost:* Define $M = \frac{-1}{4} \cdot I_{n^2}$

$$L(s, I) = \psi_{\mathcal{I}}(G)^\top M \psi_S(\boldsymbol{s}) = -\frac{1}{4}({L_G}^\flat)^\top (\boldsymbol{s}\boldsymbol{s}^\top)^\flat$$

3. *Variance preservation:* $\mathrm{Var}_{\boldsymbol{s}\sim\mathrm{Unif}(\{\pm 1\}^n)}[\boldsymbol{v}\cdot\boldsymbol{s}] = \|\boldsymbol{v}\|_2^2$

# Solution sampler: Obstacle 1

- **Goal in mind:** sample $\propto \exp\left(-\tau \cdot \psi_S(s)^\top M \psi_I(I)\right)$
  - $\tau$: temperature; as $\tau \to \infty$, samples solutions with small loss
  - But $M$ is **unknown** (results hold even for blackbox/oracle access to $L$)
- **Natural candidate:** $p(s\,; I, W) \propto \exp\left(\psi_S(s)^\top W \psi_I(I)\right)$
- In GD, is $W_t$ moving **in the direction** of $\bar{W} = -\tau M$ with $\tau \to \infty$?
  - Yes: $\nabla_W \mathcal{L}(W) \cdot M \geq 0$, so moving in the direction of $-M$ decreases loss
- But **no finite optimizer**; infimum only reached as $\|W\| \to \infty$
- Conflicts with "efficiency" desideratum:

  "First-order methods reach $2\epsilon$-opt loss in $\mathrm{poly}\left([\mathcal{W}], \frac{1}{\epsilon}\right)$ steps"

# Solution 1: Add entropy regularization

- Natural candidate: $p(s\,; I, W) \propto \exp\left(\psi_S(s)^\top W \psi_\mathcal{I}(I)\right)$
- Make loss landscape **more benign** by adding regularizer
  - **Goal:** make the landscape "quasar-convex"
- $f$ is $\gamma$-*quasar-convex* wrt minimizer $x^*$ on domain $D$ if, $\forall x \in D$

$$\nabla f(x) \cdot (x - x^*) \geq \gamma\left(f(x) - f(x^*)\right), \qquad \gamma \in [0,1]$$

- Gradient always points somewhat **toward** $x^*$
- Role of $\gamma$:
  - Measures how strongly the gradient "leans" toward the optimum
  - **Larger $\gamma$ ⇒ faster progress** for gradient descent

# Solution 1: Add entropy regularization

- Natural candidate: $p(s\,;I,W) \propto \exp\left(\psi_S(s)^\top W \psi_\mathcal{I}(I)\right)$
- Make loss landscape **more benign** by adding regularizer
- Negative entropy: $H(W) = \mathbb{E}_{I\sim\mathcal{R},s\sim p(\cdot\,;I,W)}[\log p(s\,;I,W)]$
- **Regularized objective:** $\mathcal{L}_\lambda(W) = \mathcal{L}(W) + \lambda H(W)$
- Paper shows:
  - $\bar{W} = -\frac{M}{\lambda}$ is a minimizer of $\mathcal{L}_\lambda$
  - Sampling from $p(s\,;I,\bar{W})$ yields **$2\epsilon$-opt loss** for $\lambda = \mathrm{poly}\left(\epsilon, \frac{1}{D_S}, \frac{1}{D_\mathcal{I}}, \ldots\right)$
- So **GD will eventually converge** to a $2\epsilon$-opt loss
  - But quasar-convexity parameter $\gamma$ may be very small ⇒ **slow rates**

# Obstacle 2: Vanishing gradients

For quasar-convexity, we want

$$\nabla \mathcal{L}_\lambda(W) \cdot (W - \bar{W}) \geq \gamma\big(\mathcal{L}_\lambda(W) - \mathcal{L}_\lambda(\bar{W})\big)$$

Paper proves:

$$\nabla \mathcal{L}_\lambda(W) \cdot (W - \bar{W}) = \mathrm{Var}[\psi_S(s)^\top (W - \bar{W})\psi_\mathcal{I}(I)] = \mathrm{Var}(Y)$$

Problem: variance term can be **tiny** near $\bar{W}$, so $\gamma$ may be small

- Sampler is a softmax over scores: $p(s\,; I, W) \propto \exp\big(\psi_S(s)^\top W \psi_\mathcal{I}(I)\big)$
- As $\|W\|$ grows, softmax **concentrates on argmax** of $\psi_S(s)^\top W \psi_\mathcal{I}(I)$
- So RV $Y$ becomes **almost deterministic**, so $\mathrm{Var}(Y) \to 0$

# Solution 2: Fast/slow mixture generators

**New generator family:** mix two exponential-family samplers

$$\mathcal{P} = \{(1-\beta^*)p(\cdot\,; I, W) + \beta^* p(\cdot\,; I, \rho^* W)\}$$

**Fast component:** $p(\cdot\,; I, W)$

- Drives convergence toward minimizer $\left(\bar{W} = -\frac{M}{\lambda}\right)$

**Slow component:** $p(\cdot\,; I, \rho^* W)$ with small $\rho^*$

- Stays close to the **uniform** distribution over solutions
- Key use of assumption: variance preservation under uniform dist
  - Guarantees a lower bound on variance

# Gradient descent dynamics

**At each iteration ($t$):**

1. Sample an instance $I \sim R$ (or a minibatch)
2. Sample solutions $s_1, \ldots, s_m \sim p(\cdot\,; I, \boldsymbol{w}_t)$
3. Query costs $c_k = L(I, s_k)$
4. Form a policy-gradient estimate and update $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \eta_t \boldsymbol{g}_t$

**This paper:** Exponential mixture

$$\boldsymbol{g}_t \approx \frac{1}{m} \sum_{k=1}^{m} c_k \nabla_{\boldsymbol{w}} \log p(s_k\,; I, \boldsymbol{w}_t)$$

**This paper:** Replace with entropy-regularized loss

✅ **Complete:** some $\bar{\boldsymbol{w}}$ achieves $\mathcal{L}(\bar{\boldsymbol{w}}) \leq \mathrm{opt} + \epsilon$

✅ **Compressed:** Description size of $\mathcal{W}$ is $[\mathcal{W}] = \mathrm{poly}([\mathcal{I}], 1/\epsilon)$

✅ **Efficiency:** GD reaches $2\epsilon$-opt loss in $\mathrm{poly}([\mathcal{W}], 1/\epsilon)$ steps

# Overview

- (Under assumptions) exist solution samplers that are:
  - **Complete** (near-optimal solutions exist)
  - **Compressed** (poly-sized parameterization)
  - **Efficiently optimizable** (GD finds $\epsilon$-opt in poly steps)
- Two key landscape **obstacles**:
  - Minimizers at infinity
  - Vanishing gradients near good solutions
- Two corresponding **fixes**:
  - Entropy-regularized loss → finite, quasar-convex minimizer
  - Fast/slow exponential mixture → non-vanishing gradients
- Results apply to several canonical CO problems