# Understanding Curriculum Learning in Policy Optimization for Online Combinatorial Optimization

Runlong Zhou

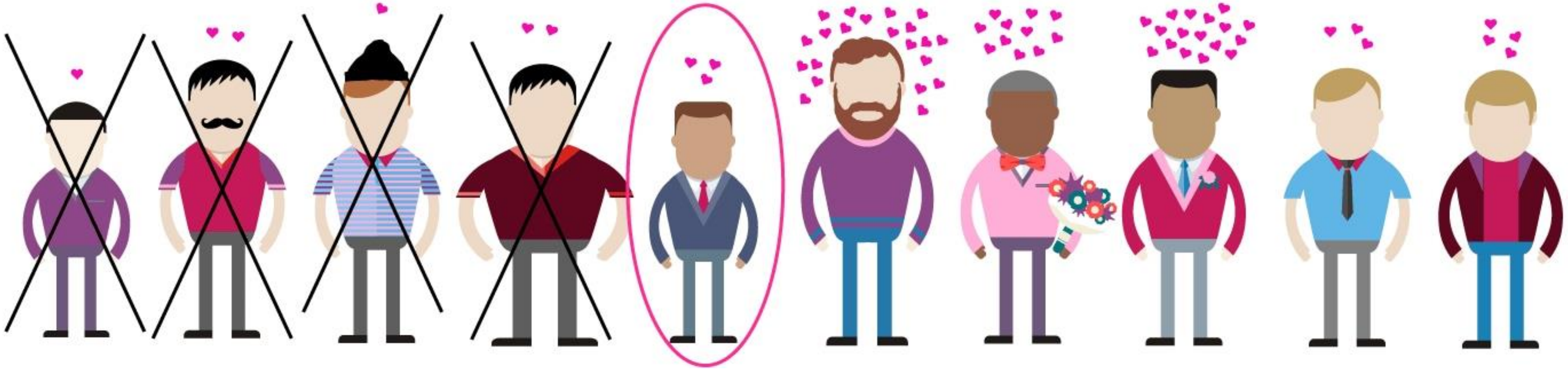
Joint work with Yuandong Tian, Yi Wu and Simon Du

# Motivation

- Machine Learning is good at Combinatorial Optimization problems
  - Is (any part of) this success explainable?
- **Online** CO matches the nature of Reinforcement Learning
  - Sequential decision-making
- Theoretical understanding of RL techniques on **online** CO
  - Curriculum Learning

# Example 1

- Secretary Problem (a.k.a. Marriage Problem, Sultan's Dowry Problem)
  - Hire one secretary among $n$ candidates, each with (different) ability score
  - Arrive sequentially, but the order is unknown
  - Once reject someone, cannot come back and re-hire
  - Once hire someone, ends
  - **Maximize the probability of hiring the best candidate**



https://www.m2woman.com/when-to-stop-dating-and-settle-down-according-to-math/

# Example 1 (cont'd)

- Secretary Problem
  - Isomorphic up to only the relative rank
  - $\boldsymbol{n}!$ Permutations (instances), each with equal probability
  - Find a policy working **averagely well** on the instance distribution

# Example 2

- Online Knapsack
  - $n$ items arrive sequentially
  - Each with value $v_i$ and size $s_i$ **revealed upon arrival**
  - Total budget $B$
  - Once reject something, cannot come back and re-pick
  - To pick something, requires total size $\leq B$
  - **Maximize total value of picked items**
  - Distribution of instances defined via each $(v_i, s_i) \sim F_v \times F_s$ i.i.d.

# Formulation

- Latent MDPs [Kwon et al.,2021]
  - A distribution of MDPs $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$
  - Each with **weight (probability)** $\boldsymbol{w_1}, \ldots, \boldsymbol{w_M}$
  - All MDPs share **states $\boldsymbol{S}$, actions $\boldsymbol{A}$, horizon $\boldsymbol{H}$**
  - But have their own **initial state distribution $\boldsymbol{\nu_m}$ reward function $\boldsymbol{r_m}$ (bounded by $[\mathbf{0}, \mathbf{1}]$), transition $\boldsymbol{P_m}$**
- Why use this setting?
  - Characterize **instances** and **working averagely well**

# Formulation (cont'd)

- Policy class
  - Stationary $\pi: S \to \Delta(A)$
  - Log-linear parameterization
    - Assume we have a fixed feature mapping $\phi: S \times A \to \mathbb{R}^d$
    - Learn parameter $\theta$, and policy

$$\pi(a|s) = \frac{\exp\left(\theta^\top \phi(s,a)\right)}{\sum_{a' \in A} \exp\left(\theta^\top \phi(s,a')\right)}$$

# Formulation (cont'd)

- Value functions and **entropy regularization**
  - $V_{i,h}^{\pi,\lambda}(s) = \mathbb{E}\left[\sum_{t=0}^{h-1} r_i(s_t, a_t) + \lambda \ln \frac{1}{\pi(a_t|s_t)} \mid \mathcal{M}_i, \pi, s_0 = s\right]$
  - $Q_{i,h}^{\pi,\lambda}(s) = \mathbb{E}\left[\sum_{t=0}^{h-1} r_i(s_t, a_t) + \lambda \ln \frac{1}{\pi(a_t|s_t)} \mid \mathcal{M}_i, \pi, (s_0, a_0) = (s, a)\right]$
  - $A_{i,h}^{\pi,\lambda}(s) = Q_{i,h}^{\pi,\lambda}(s) - V_{i,h}^{\pi,\lambda}(s)$
- Goal
  - Find $\pi_\lambda^\star = \mathrm{argmax}_\pi \sum_i w_i \sum_{s_0} \nu_i(s_0)\, V_{i,h}^{\pi,\lambda}(s_0)$

# Prerequisites

- State-action visitation distribution
  - $d_{i,h}^{\pi}(s) = \mathbb{P}(s_h = s \mid \mathcal{M}_i, \pi)$
  - $d_{i,h}^{\pi}(s,a) = d_{i,h}^{s_0,\pi}(s)\pi(a|s)$
- Function approximation loss

$$L(g;\theta,v) = \sum_{i=1}^{M} w_i \sum_{h=1}^{H} \mathop{\mathbb{E}}_{s,a\sim v_{i,H-h}} \left[ \left( A_{i,h}^{\pi_\theta,\lambda}(s,a) - g^\top \nabla \ln \pi_\theta(a|s) \right)^2 \right]$$

- Fisher information matrix

$$\Sigma_v^\theta = \sum_{i=1}^{M} w_i \sum_{h=1}^{H} \mathop{\mathbb{E}}_{s,a\sim v_{i,H-h}} \left[ \nabla_\theta \ln \pi_\theta(a|s) \left( \nabla_\theta \ln \pi_\theta(a|s) \right)^\top \right]$$

# Natural Policy Gradient

- Initial param: $\theta_0$
- Update:
  $$\theta_{t+1} = \theta_t + \eta g_t$$
- $g_t \approx \operatorname{argmin}_g L(g; \theta_t, d^t)$

**Algorithm 3** `NPG`: Sample-based NPG (full version).

**Input:** Environment $E$; learning rate $\eta$; episode number $T$; batch size $N$; initialization $\theta_0$; sampler $\pi_s$; regularization coefficient $\lambda$; entropy clip bound $U$; optimization domain $\mathcal{G}$.
**for** $t \leftarrow 0, 1, \ldots, T-1$ **do**
  Initialize $\widehat{F}_t \leftarrow 0^{d \times d}$, $\widehat{\nabla}_t \leftarrow 0^d$.
  **for** $n \leftarrow 0, 1, \ldots, N-1$ **do**
    **for** $h \leftarrow 0, 1, \ldots, H-1$ **do**
      **if** $\pi_s$ is not None **then**
        $s_h, a_h, \widehat{A}_{H-h}(s_h, a_h) \leftarrow$ `Sample` $(E, \pi_s, \text{True}, \pi_t, h, \lambda, U)$ (see Alg. 4).
        // $s, a \sim \widetilde{d}^{\pi_s}_{m,h}$, estimate $A^{t,\lambda}_{m,H-h}(s,a)$.
      **else**
        $s_h, a_h, \widehat{A}_{H-h}(s_h, a_h) \leftarrow$ `Sample` $(E, \pi_t, \text{False}, \pi_t, h, \lambda, U)$.
        // $s, a \sim d^{\theta_t}_{m,h}$, estimate $A^{t,\lambda}_{m,H-h}(s,a)$.
      **end if**
    **end for**
    Update:

$$\widehat{F}_t \leftarrow \widehat{F}_t + \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_{\theta_t}(a_h|s_h) \left(\nabla_\theta \ln \pi_{\theta_t}(a_h|s_h)\right)^\top,$$

$$\widehat{\nabla}_t \leftarrow \widehat{\nabla}_t + \sum_{h=0}^{H-1} \widehat{A}_{H-h}(s_h, a_h) \nabla_\theta \ln \pi_{\theta_t}(a_h|s_h).$$

  **end for**
  Call any solver to get $\widehat{g}_t \leftarrow \arg\min_{g \in \mathcal{G}} g^\top \widehat{F}_t g - 2 g^\top \widehat{\nabla}_t$.
  Update $\theta_{t+1} \leftarrow \theta_t + \eta \widehat{g}_t$.
**end for**
**Return:** $\theta_T$.

# Analysis

- Assumption
  - $g_t^\star$ is the true minimizer of $L$ at time $t$
  - $d^\star$ is the visitation distribution of the optimal policy
  - $||\phi(s,a)||_2 \leq B$

**Definition 4.** *Define for* $0 \leq t \leq T$*:*

- *(Excess risk)* $\epsilon_{\text{stat}} := \max_t \mathbb{E}[L(g_t; \theta_t, d^t) - L(g_t^\star; \theta_t, d^t)]$*;*
- *(Transfer error)* $\epsilon_{\text{bias}} := \max_t \mathbb{E}[L(g_t^\star; \theta_t, d^\star)]$*;*
- *(Relative condition number)* $\kappa := \max_t \mathbb{E}\left[\sup_{x \in \mathbb{R}^d} \frac{x^\top \Sigma_{d^\star}^{\theta_t} x}{x^\top \Sigma_t x}\right]$*. Note that term inside the expectation is a random quantity as* $\theta_t$ *is random.*

*The expectation is with respect to the randomness in the sequence of weights* $g_0, g_1, \ldots, g_T$*.*

# Analysis (cont'd)

- Main result

**Theorem 6.** *With Def. 4, 5 and 8, our algorithm enjoys the following performance bound:*

$$\mathbb{E}\left[\min_{0\le t\le T} V^{\star,\lambda} - V^{t,\lambda}\right] \le \frac{\lambda(1-\eta\lambda)^{T+1}\Phi(\pi_0)}{1-(1-\eta\lambda)^{T+1}} + \eta\frac{B^2G^2}{2} + \sqrt{H\epsilon_{\text{bias}}} + \sqrt{H\kappa\epsilon_{\text{stat}}},$$

*where $\Phi(\pi_0)$ is the Lyapunov potential function which is only relevant to the initialization.*

- Interpretation
  - **Linear convergence** ($\lambda = 0$ gets $1/\sqrt{T}$ rate)
  - $\epsilon_{\text{bias}}$ depends on the **design of feature mapping**, and is hardly removable
  - $\epsilon_{\text{stat}}$ depends on the closeness between $g_t$ and $g_t^\star$, can be reduced with a bigger batch-size
  - $\kappa$ shows the closeness between $d^\star$ and $d^t$, and **is of our special interest**

# Curriculum Learning [Bengio et al.,2009]

- First learn small scale problems, then use this model to initialize large scale learning
- Mentioned as “Bootstrapping” in [Kong et al.,2019]
    - They suggested a series of curricula: $10, 20, \ldots, 200$
    - No need to do this, because **reducing $\boldsymbol{\kappa}$ is sufficient**
- Even works for changing distribution
    - Uniform over $10!$ instances
    - Arbitrary complex for $200!$ Instances
    - As long as the optimal policies are **“Similar”**, it transfers well

# Curriculum Learning for SP

- Distribution modeling
  - Suppose for each candidate $i$, it has probability $P_i$ to be the so-far best
  - For classical SP, $P_i = 1/i$
- Optimal policy
  - Always a $\boldsymbol{p}$**-threshold policy**: accept if and only if $i/n > p$ and is so-far best
  - For classical SP, $p = 1/\mathrm{e}$

# Curriculum Learning for SP (cont'd)

- Comparing $\kappa$ under with/without curriculum learning

**Theorem 7.** *Assume that each candidate is independent of others and the i-th candidate has a probability $P_i$ of being the best so far (Sec. 4.1). Assume the optimal policy is a p-threshold policy and the sampling policy is a q-threshold policy. There exists a policy parameterization such that:*

$$\kappa_{\text{curl}} = \Theta\left(\begin{cases} \prod_{j=\lfloor nq \rfloor+1}^{\lfloor np \rfloor} \frac{1}{1-P_j}, & q \le p, \\ 1, & q > p, \end{cases}\right),$$
$$\kappa_{\text{naïve}} = \Theta\left(2^{\lfloor np \rfloor} \max\left\{1, \max_{i \ge \lfloor np \rfloor + 2} \prod_{j=\lfloor np \rfloor+1}^{i-1} 2(1-P_j)\right\}\right), \quad (1)$$

*where $\kappa_{\text{curl}}$ and $\kappa_{\text{naïve}}$ are $\kappa$ of the sampling policy and the naïve random policy, respectively.*

# Curriculum Learning for SP (cont'd)

- Classical case
  - The target problem is the classical SP

$$\kappa_{\text{curl}} = \begin{cases} \frac{\lfloor n/\mathrm{e} \rfloor}{\lfloor nq \rfloor}, & q \le \frac{1}{\mathrm{e}}, \\ 1, & q > \frac{1}{\mathrm{e}}, \end{cases} \qquad \kappa_{\text{naïve}} = 2^{n-1} \frac{\lfloor n/\mathrm{e} \rfloor}{n-1}.$$

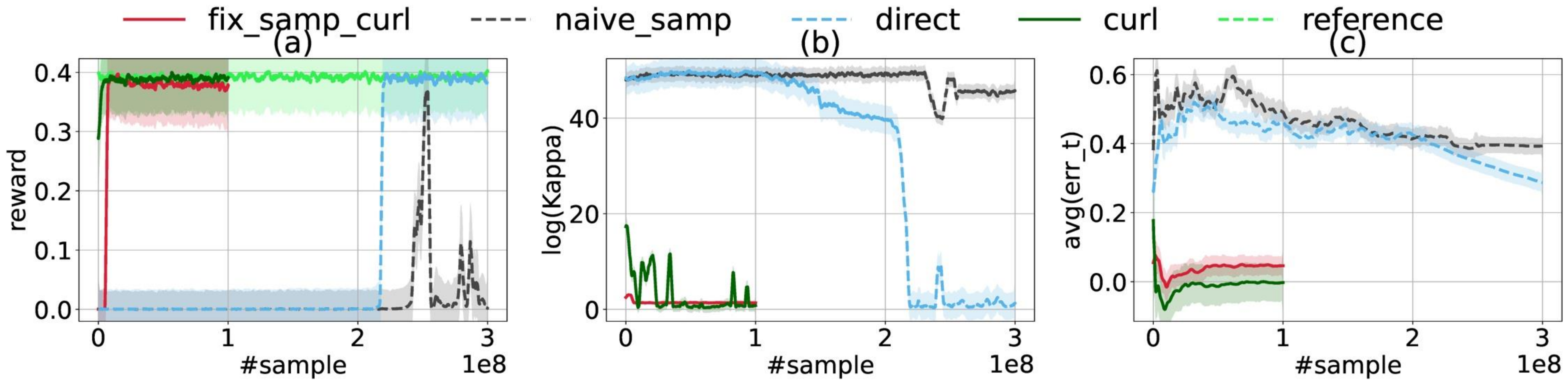
- General case
  - The target problem satisfies $P_i \le 1/2$

$$\kappa_{\text{curl}} \le \begin{cases} 2^{\lfloor np \rfloor - \lfloor nq \rfloor}, & q \le p, \\ 1, & q > p, \end{cases} \qquad \kappa_{\text{naïve}} \ge 2^{\lfloor np \rfloor}.$$

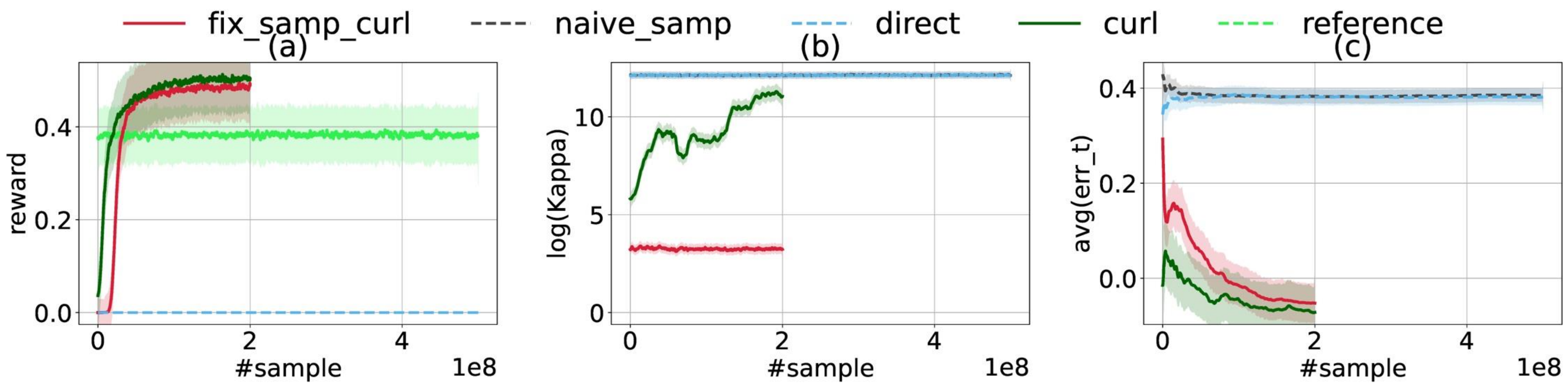
- Failure case
  - Best candidate always come as the last one

$\kappa_{\text{curl}} = \infty$, larger than $\kappa_{\text{naïve}} = 2^{n-1}$.

# Experiments – Secretary Problem



# Experiments – Online Knapsack



# References


- Semih Cayci, Niao He, and R. Srikant. Linear convergence of entropy-regularized natural policygradient with linear function approximation, 2021.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. InProceedings of the 26th Annual International Conference on Machine Learning, ICML ’09, page41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161.doi: 10.1145/1553374.1553380. URLhttps://doi.org/10.1145/1553374.1553380.
- Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policygradient methods: Optimality, approximation, and distribution shift, 2020.
- Weiwei Kong, Christopher Liaw, Aranyak Mehta, and D. Sivakumar. A new dog learns old tricks: Rlfinds classic optimization algorithms. InICLR, 2019.
- Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Rl for latent mdps:Regret guarantees and a lower bound, 2021.