



探索基于学习的数学优化算法与理论

文再文

北京大学

北京国际数学研究中心

国际机器学习研究中心

大数据分析与应用技术国家工程实验室

致谢：谢中林、李天佑、陈铨、邹海军、周瑞松、袁浩然、潘奕成

The Bitter Lesson (March 13, 2019)



Richard Sutton

2024年图灵奖

核心观点

The biggest lesson that can be read from 70 years of AI research is that **general methods** that **leverage computation** are ultimately the most effective, and by a large margin

历史经验

- 棋类游戏：规则与专家策略 → 被搜索与学习方法超越
- 语音/视觉：特征工程 → 端到端深度学习
- 控制/机器人：人类建模直觉 → 强化学习与通用优化

核心教训

One thing that should be learned from the bitter lesson is **the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great**. The two methods that seem to **scale arbitrarily** in this way are **search** and **learning**

人工智能驱动和优化算法

深度学习免模型方法

- Learn GD by GD
- 算法直观且易实现
- 泛化性无保证



核心诉求

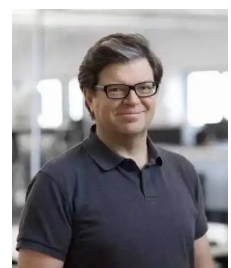
优化问题的表征与理解

适应不同规模与种类

复杂决策与推理能力

学习优化方法

- LISTA算法展开
- 参数的利用效率高
- 依赖经典优化算法



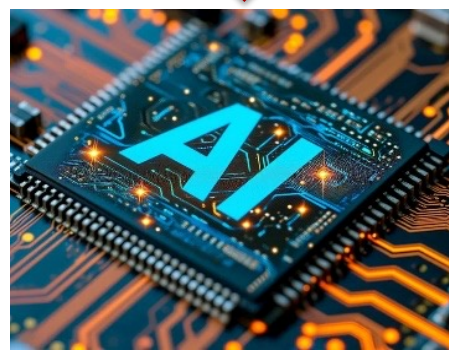
图灵奖得主
LeCun

端到端学习方法

- 指针网络求解组合优化
- 高效率、高精度
- 难以泛化到大规模问题



DeepMind首席科学家
Oriol Vinyals



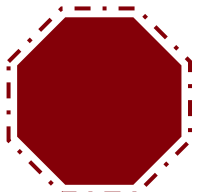
数学优化基座大模型

强化学习方法

- 整数规划学习分支方法
- 精确方法有理论保障
- 依赖奖励函数设计



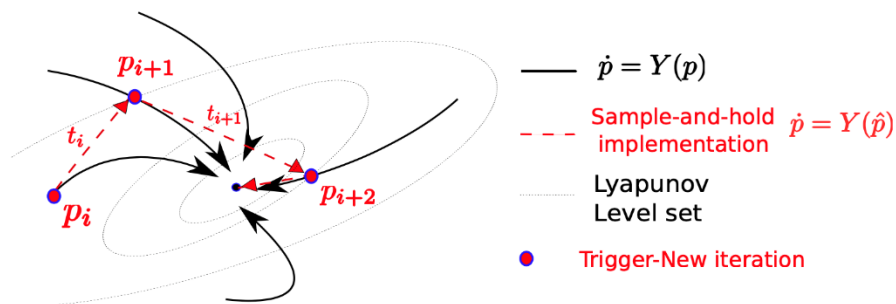
Farkas 奖得主
Andrea Lodi

 ODE+学习 → 优化算法



微分方程视角下学习优化：两个重要问题

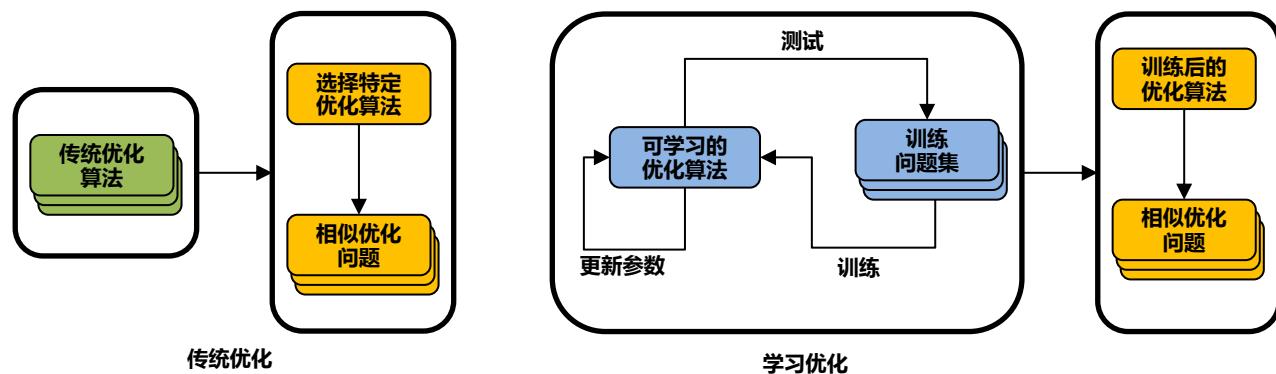
□ 微分方程快速收敛的轨迹如何**安全转化**为实用的优化算法？



稳定的离散需要控制轨迹的曲率

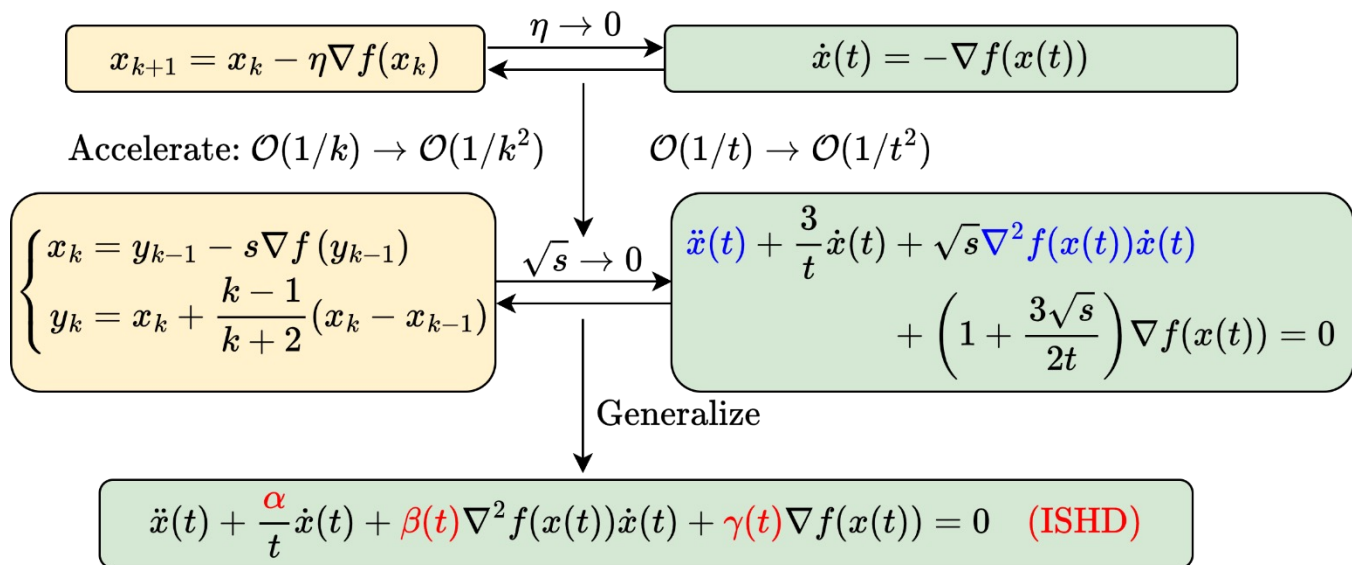
- 给出连续轨迹收敛的充分条件
- 稳定性条件控制全局离散误差
- **加速不丢失**：收敛速度可达 $1/k^2$

□ 在这一大类实用的优化算法中，怎么**针对特定问题**选择最好的算法？



- 提出**停止时刻**作为连续版复杂度
- 基于优化算法收敛的条件，构造带期望约束的 L2O 问题
- 给出此L2O问题收敛的理论保证

加速方法的连续时间视角 $\min_x f(x)$



考虑逻辑回归问题

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{N} \sum_{i=1}^N \log(1 + \exp(-b_i \langle a_i, w \rangle))$$

其中 $\{a, b_i\} \in \mathbb{R}^n \times \{0, 1\}, i \in [N]$

选取 $p = 5, \alpha = 2p + 1, \beta(t) \equiv 0$ 及 $\gamma(t) = p^2 t^{p-2}$

此时(ISHD)的轨迹必然收敛, 且收敛速率为

$$f(x(t)) - f_* \leq \mathcal{O}(1/t^p)$$

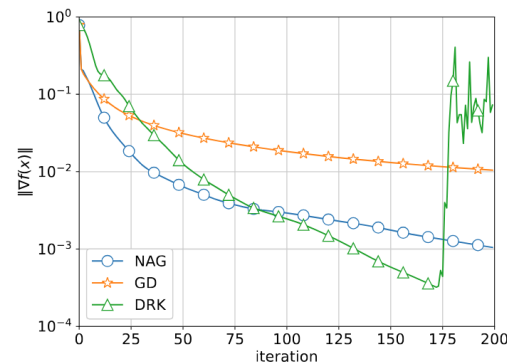
直接使用Runge-Kutta格式离散, 最终会发散

已有的收敛性结论:

令 $w(t) = \gamma(t) - \dot{\beta}(t) - \beta(t)/t$, 方程(ISHD)的轨迹收敛的条件为

$$\gamma(t) > \dot{\beta}(t) + \frac{\beta(t)}{t}, \quad tw(t) \leq (\alpha - 3)w(t), \quad \text{对于所有 } t \geq t_0$$

收敛速率为: $f(x(t)) - f_* = \mathcal{O}(1/(t^2 w(t)))$



理论结果：收敛性与稳定性条件

- 任取常数 $\kappa \in (0, 1], \lambda \in (0, \alpha - 1], f$ 二次可微函数

- 假设如下收敛性条件成立:

$$\delta(t) = t^2(\gamma(t) - \kappa\dot{\beta}(t) - \kappa\beta(t)/t) + (\kappa(\alpha - 1 - \lambda) - \lambda(1 - \kappa))t\beta(t),$$

$$w(t) = \gamma(t) - \dot{\beta}(t) - \beta(t)/t, \quad \delta(t) > 0, \quad \text{and} \quad \dot{\delta}(t) \leq \lambda tw(t),$$

- 其中 $\alpha \geq 3, t_0 > 0, \varepsilon > 0$ 且函数 β, γ 负连续可微, 则有如下收敛性质:

$$f(x(t)) - f_* \leq \mathcal{O}\left(\frac{1}{\delta(t)}\right), \quad \|\nabla f(x(t))\| \leq \mathcal{O}\left(\frac{1}{t\beta(t)}\right), \quad \|\dot{x}(t)\| \leq \mathcal{O}\left(\frac{1}{t}\right),$$

$$\int_{t_0}^{\infty} (\lambda tw(t) - \dot{\delta}(t))(f(x(t)) - f_*) dt \leq \infty, \quad \int_{t_0}^{\infty} t(\alpha - 1 - \lambda)\|\dot{x}(t)\|^2 dt \leq \infty,$$

$$\int_{t_0}^{\infty} t^2\beta(t)w(t)\|\nabla f(x)\|^2 dt \leq \infty, \quad \int_{t_0}^{\infty} t^2\beta(t)\langle \nabla^2 f(x(t))\dot{x}(t), \dot{x}(t) \rangle dt \leq \infty.$$

- 假设收敛性条件及其假设成立, 且存在三个非负实数满足

$$|\dot{\beta}(t)| \leq C_1\beta(t), \quad |\dot{\gamma}(t) - \ddot{\beta}(t)| \leq C_2(\gamma(t) - \dot{\beta}(t)), \quad \beta(t) \leq C_3w(t).$$

- 且存在可积函数满足

$$\Lambda(x, f) \geq \|\nabla^2 f(x)\| \quad \text{for all } x, \quad \text{and} \quad \sup_{t \geq t_0} \beta(t) \Lambda(x(t), f) < \infty.$$

- 若如下稳定性条件满足, 必然有 $f(x_k) - f_* \leq \mathcal{O}(1/k)$:

$$\frac{\alpha}{t}\beta(t) \leq \gamma(t) - \dot{\beta}(t), \quad \alpha' := \sup_{k \geq 0} ((\gamma(t_k) - \dot{\beta}(t_k))h - \beta(t_k)) t_k \Lambda_k < \alpha,$$

$$\beta(t_k)\sqrt{\Lambda_k} \leq \sqrt{\gamma(t_k) - \dot{\beta}(t_k)} + \sqrt{\gamma(t_k) - \dot{\beta}(t_k) - \frac{\alpha}{t_k}\beta(t_k)},$$

收敛性条件限制(ISHD)在一个保证收敛的空间中

收敛性条件



(ISHD)的轨迹收敛



常数步长前向Euler离散:

$$\begin{cases} \frac{x_{k+1} - x_k}{h} = v_k - \beta(t_k)\nabla f(x_k), \\ \frac{v_{k+1} - v_k}{h} = -\frac{\alpha}{t}(v_k - \beta(t_k)\nabla f(x_k)) + (\dot{\beta}(t_k) - \gamma(t_k))\nabla f(x_k) \end{cases} \quad (\text{EIGAC})$$

需限制系数增长速度, 以保证(EIGAC)不发散



稳定性条件



(EIGAC)的轨迹收敛

停止时刻与学习优化框架

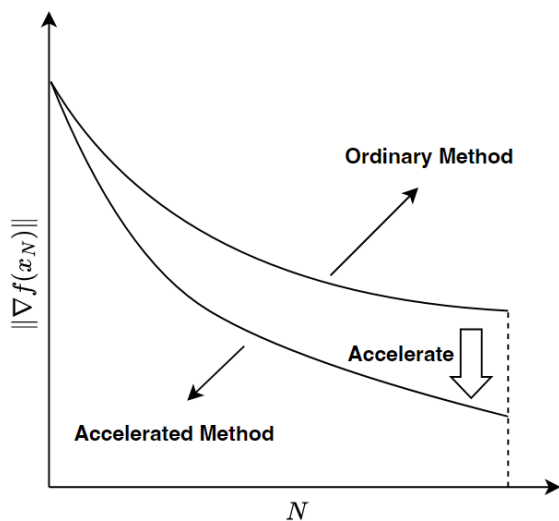
停止时刻:

记(ISHD)的轨迹为 $X(\Xi, t, f)$

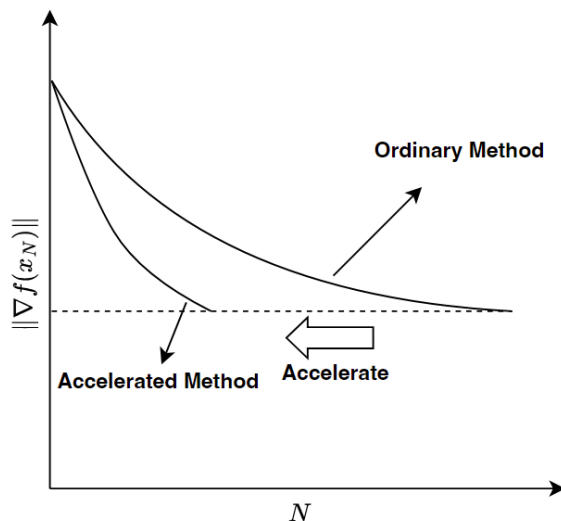
则针对给定的精度, 连续时间下的复杂度定义为

$$T(\Xi, f) = \inf\{t \mid \|\nabla f(X(\Xi, t, f))\| \leq \varepsilon, t \geq t_0\}$$

连续时间下的复杂度



基于最优性度量的加速范式



基于复杂度的加速范式

给定随机变量 $\xi \sim \mathbb{P}$

定义关于参数化函数类的分布为

$$\mathbb{E}_f[T(\Xi, f)] = \int_{\xi} T(\Xi, f(\cdot; \xi)) d\mathbb{P}(\xi) = \mathbb{E}_{\xi}[T(\Xi, f(\cdot; \xi))]$$

在收敛性和稳定性条件下, 最小化停止时间的期望

$$\min_{\Xi} \mathbb{E}_f[T(\Xi, f)],$$

$$\text{s.t. } \mathbb{E}_f[P(\Xi, f)] \leq 0, \quad \mathbb{E}_f[Q(\Xi, f)] \leq 0$$

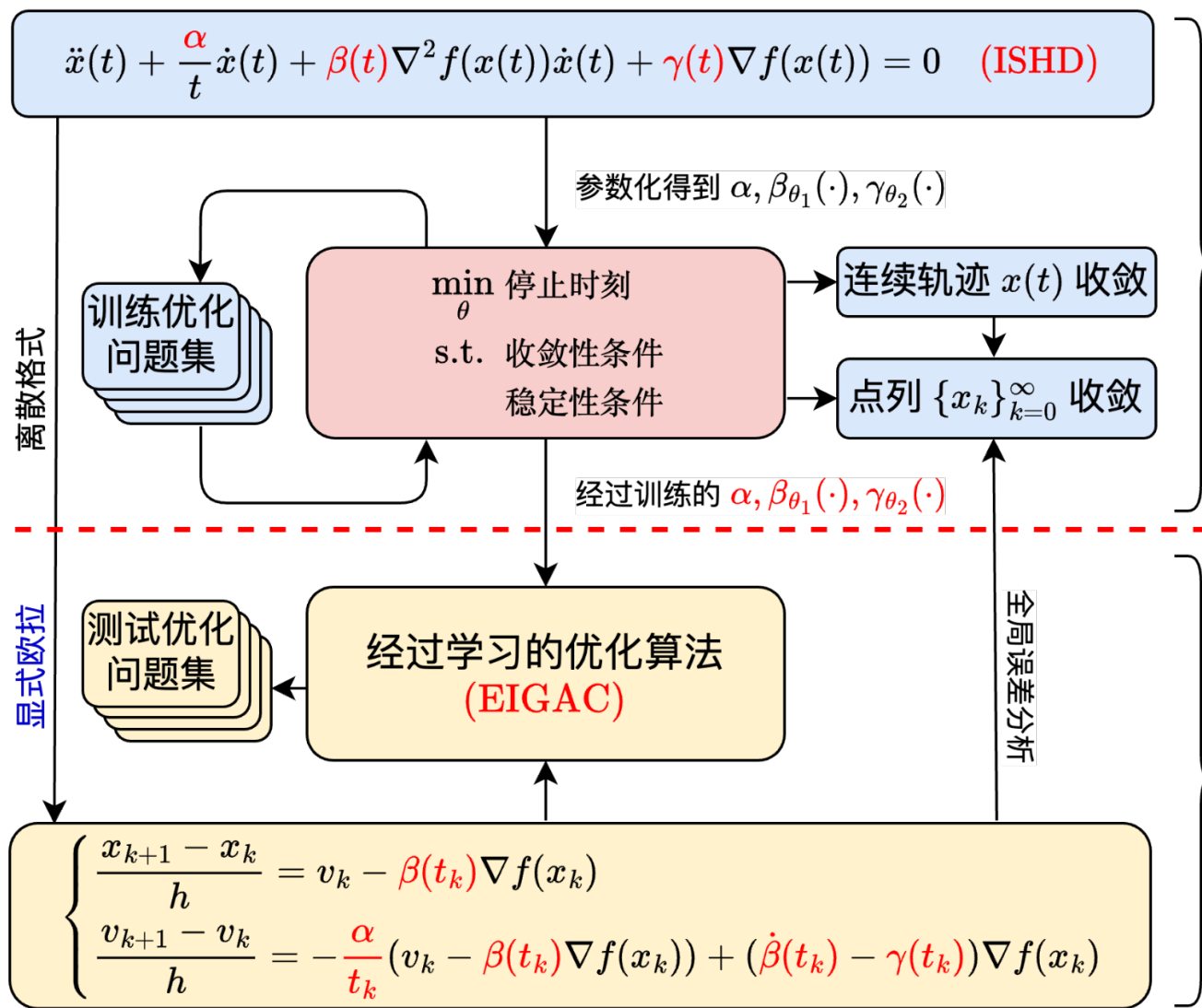
参数化求解

参数化 $\beta \rightarrow \beta_{\theta_1}, \gamma \rightarrow \gamma_{\theta_2}$ 并记 $\theta = (\alpha, \theta_1, \theta_2)$

使用精确罚函数转化为标准随机优化问题

$$\begin{aligned} \min_{\theta} \Upsilon(\theta) &= \mathbb{E}_f[T(\theta, f)] + \rho (\mathbb{E}_f[P(\theta, f)] + \mathbb{E}_f[Q(\theta, f)]) \\ &= \mathbb{E}_f [T(\theta, f) + \rho (P(\theta, f) + Q(\theta, f))] \end{aligned}$$

训练与测试框架



训练阶段

测试阶段



方法创新

微分方程统一算法形式

停止时刻度量算法效率

理论结果保证算法收敛



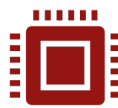
潜在影响

基于停止时刻的学习优化建模

保证学习优化算法收敛的理论

保持连续时间收敛性质的算法

问题类型



数值结果

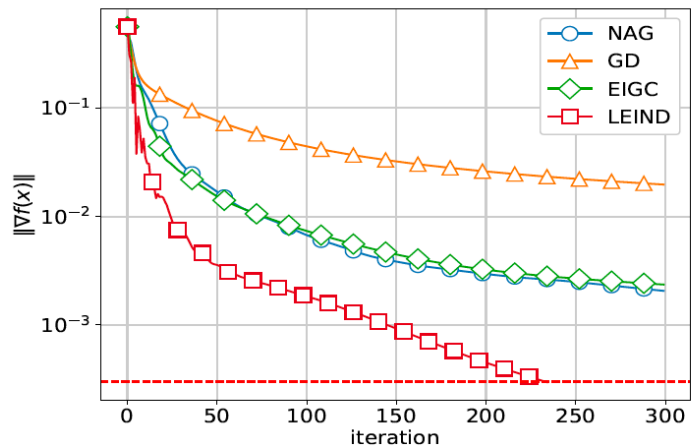
逻辑回归 $\min_{x \in \mathbb{R}^n} f_{\mathcal{D}}(x) = \frac{1}{|\mathcal{D}|} \sum_{(a_i, b_i) \in \mathcal{D}} \log(1 + \exp(-b_i \langle a_i, x \rangle))$

L^p 回归 $\min_{x \in \mathbb{R}^n} f_{\mathcal{D}}(x) = \frac{1}{|\mathcal{D}|} \sum_{(a_i, b_i) \in \mathcal{D}} \frac{1}{p} (\langle a_i, x \rangle - b_i)^p$

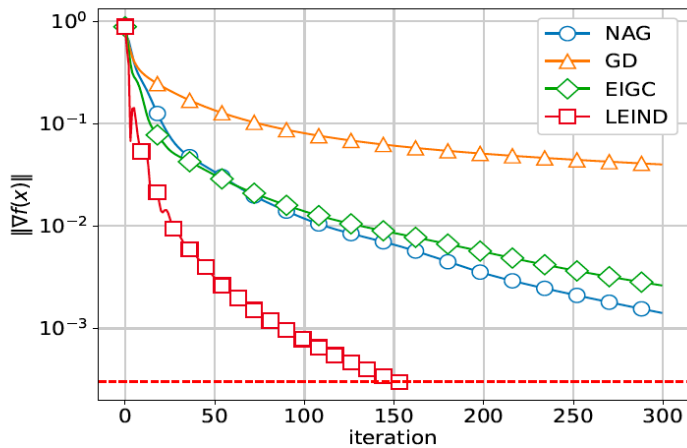
对比NAG求解时间平均减少 60%

未经学习的算法与NAG表现相当

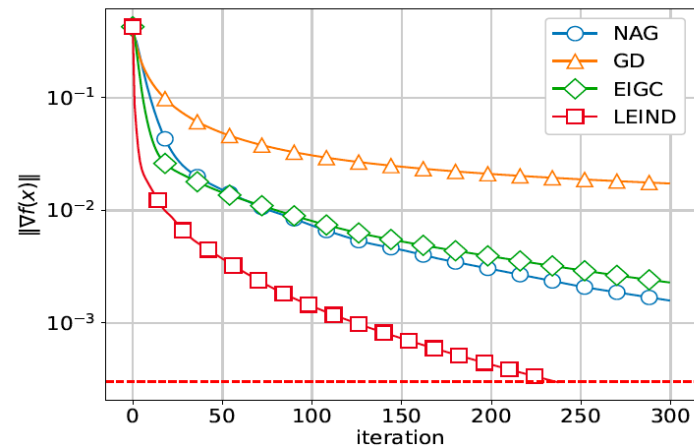
测试结果: 逻辑回归



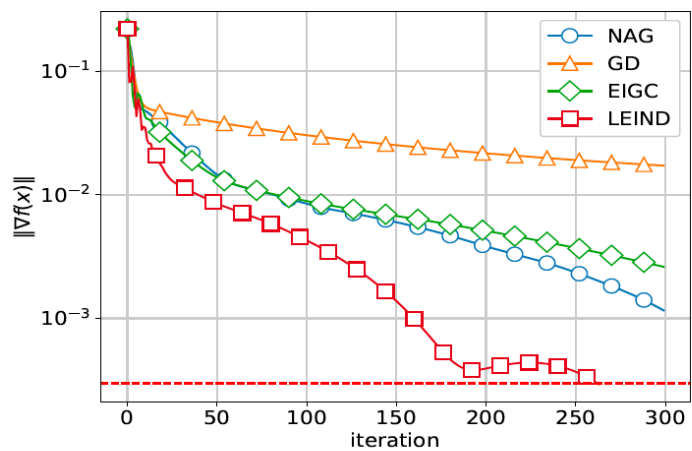
(c) a5a



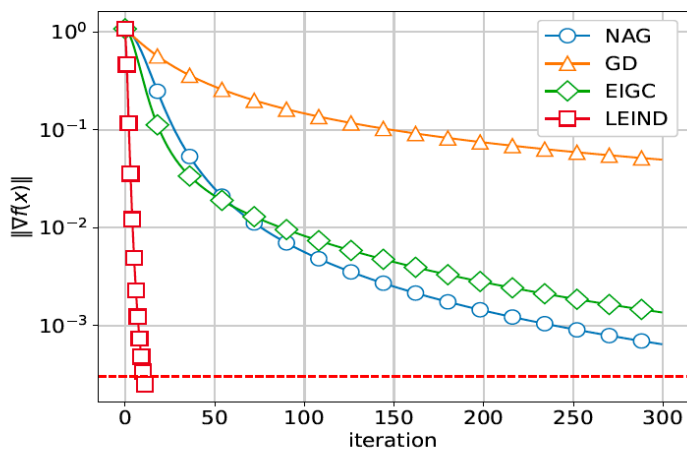
(d) mushrooms



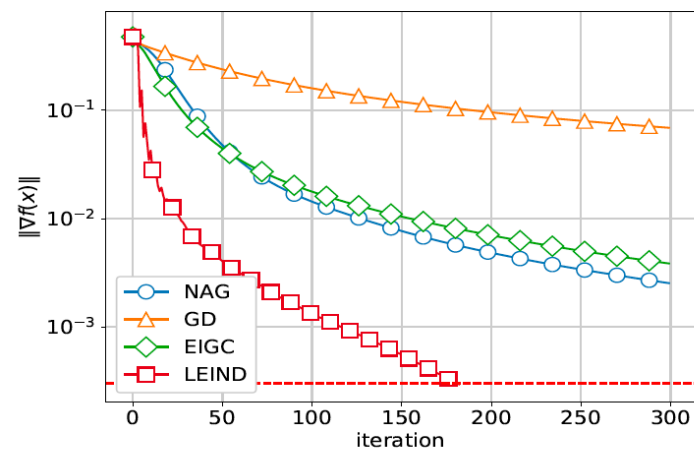
(e) w3a



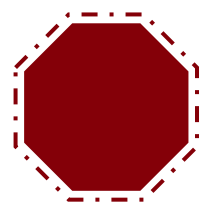
(f) covtype



(g) separable



(h) phishing



可微停止时刻加速优化



通过可微停止时刻加速优化

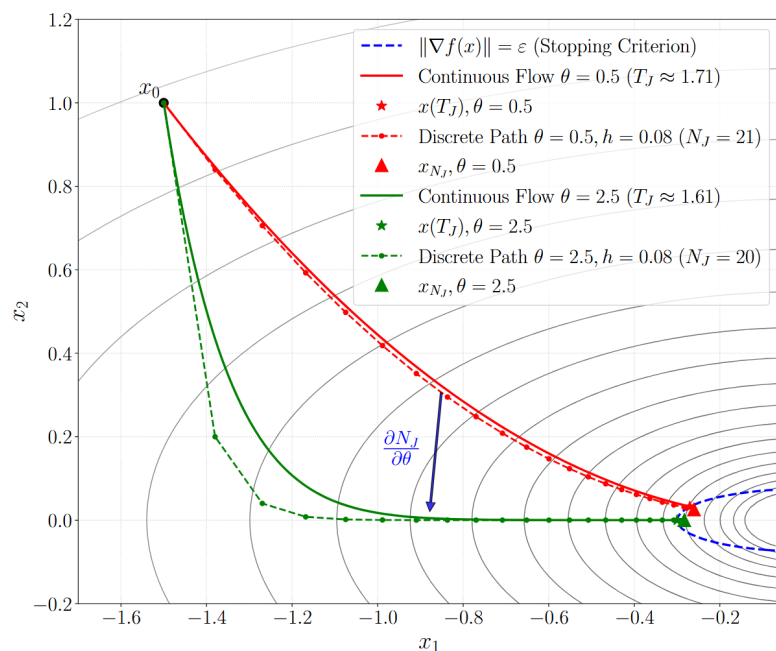
优化算法可视为对应ODE的离散

$$\dot{x}(t) = -\mathcal{A}(\theta, x(t), t), \quad x(t_0) = x_0$$

$$x_{k+1} = x_k - h\mathcal{A}(\theta, x_k, t_k)$$

考虑二维的rescaled gradient descent

$$f(x_1, x_2) = 0.5x_1^2 + 2x_2^2 \quad \mathcal{A}(x, \theta, t) = \text{diag}(1, \theta)\nabla f(x)$$



实线：不同优化算法的对应的ODE连续轨迹

虚线：针对ODE使用大步长离散得到的点列

离散点列和连续轨迹高度吻合，可借助点列对停时T求导，摆脱ODE依赖

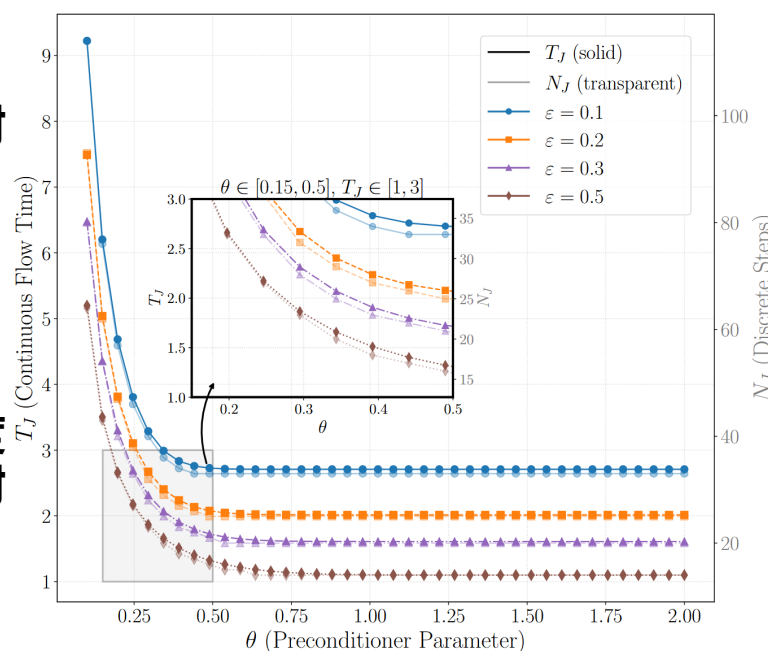
连续情形下的停止时刻可证明可微性

$$T_J(\theta, x_0, \varepsilon) = \inf_t \{t \mid J(x(t)) \leq \varepsilon\}$$

离散情形下停止时刻不可微

$$N_J(\theta, x_0, \varepsilon) = \min_n \{n \mid J(x_n) \leq \varepsilon\}$$

取停止准则 $J(x)$ 为 $\|\nabla f(x)\|^2$



连续情形下停止时刻导数依赖求解伴随灵敏度方程

基于离散点列反向传播的导数非常接近连续情形，无需求解方程

停止时刻灵敏度的定义、理论性质与高效计算

核心思想:

- 离散点列与连续轨迹的充分接近
- 类比连续情形, 停时处准则J对参数的“导数”近似为0

$$0 \approx \nabla J(x_N)^\top \left(\frac{\partial x_N}{\partial N} \frac{\partial N}{\partial \theta} + \frac{\partial x_N}{\partial \theta} \right) \approx \frac{J(x_N) - J(x_{N-1})}{h} \frac{\partial N}{\partial \theta} + \nabla J(x_N)^\top \frac{\partial x_N}{\partial \theta}$$

基于隐函数定理形式上定义的灵敏度是导数的美好逼近

定理 5 (连续停止时间的导数)

若 \mathcal{A} 与 J 连续可微, 且 $\nabla J(x(T))^\top \dot{x}(T) \neq 0$, 则 $T_J(\theta, x_0, \varepsilon)$ 关于 θ 可微, 且满足:

$$\frac{\partial T_J}{\partial \theta} = - \frac{\nabla J(x(T))^\top \partial x(T) / \partial \theta}{\nabla J(x(T))^\top \dot{x}(T)}$$

定义 6 (离散停止时间的灵敏度)

设 $N = N_J(\theta, x_0, \varepsilon)$ 表示离散停止时间, 则 N 关于 θ 和 x_0 的灵敏度定义为

$$\frac{\partial N}{\partial \theta} := - \frac{h \nabla J(x_N)^\top \partial x_N / \partial \theta}{J(x_N) - J(x_{N-1})}$$

定理 7 (近似误差)

在 \mathcal{A} 与 J 满足适当正则性条件下, 对于充分小的 h , 有:

$$\|\nabla_\theta T_J(\theta, x_0, \varepsilon) - \nabla_\theta N_J(\theta, x_0, \varepsilon)\| = \mathcal{O}(h)$$

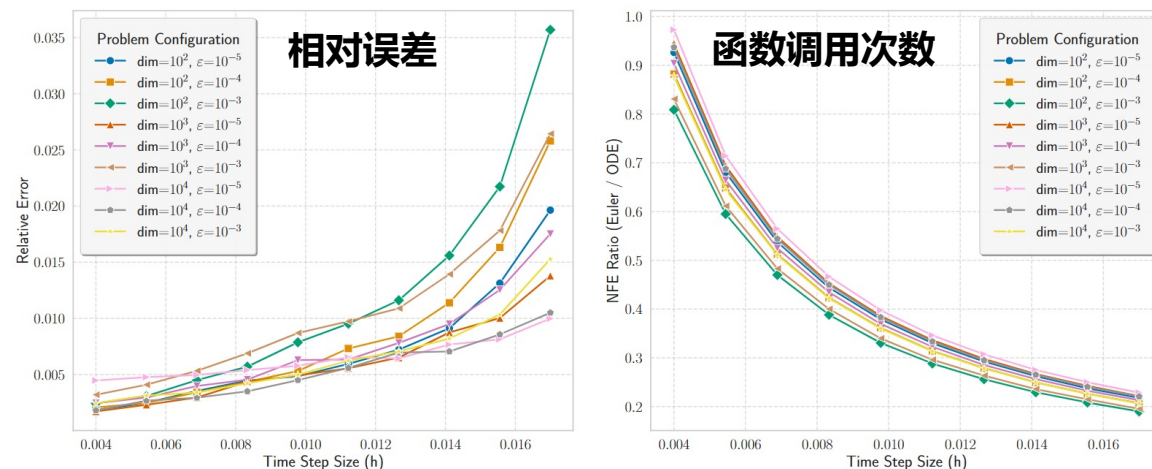
离散伴随灵敏度方法:

- 内存占用 $\mathcal{O}(1)$, 若直接使用自动微分则内存占用高
- 可以反向传播上千层, 直接使用自动微分不稳定

Algorithm 1 Discrete Adjoint Method for Sensitivity Components

- Input:** Forward trajectory $\{x_k\}_{k=0}^N$, parameters θ , $J(x_N)$, time step h , initial time t_0 .
- Output:** $S_\theta = \nabla J(x_N)^\top (\partial x_N / \partial \theta)$ and $S_{x_0} = \nabla J(x_N)^\top (\partial x_N / \partial x_0)$.
- $\lambda \leftarrow \nabla J(x_N)$. ▷ Initialize adjoint vector
- $S_\theta \leftarrow \mathbf{0}$ (vector of same size as θ). ▷ Initialize sensitivity component for θ
- for** $k = N - 1$ **downto** 0 **do**
- $t_k \leftarrow t_0 + kh$.
- $S_\theta \leftarrow S_\theta - h \left(\frac{\partial \mathcal{A}(\theta, x_k, t_k)}{\partial \theta} \right)^\top \lambda$. ▷ Accumulate contribution to S_θ
- $\lambda \leftarrow \left(I - h \frac{\partial \mathcal{A}(\theta, x_k, t_k)}{\partial x_k} \right)^\top \lambda$. ▷ Propagate adjoint vector backward
- end for**
- $S_{x_0} \leftarrow \lambda$. ▷ After the loop, λ represents $\nabla J(x_N)^\top (\partial x_N / \partial x_0)$
- return** S_θ, S_{x_0} .

针对不同维数和次优程度高效近似: 20%代价 -> 97%精度



应用：加速学习优化收敛与学习率在线更新

$$\min_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{f, x_0} \left[\sum_{k=0}^{K_{\max}} w_k f(x_k) + \lambda N_J(\theta, x_0, \varepsilon) \right]$$

L2O-RNNprop-Time vs L2O-RNNprop

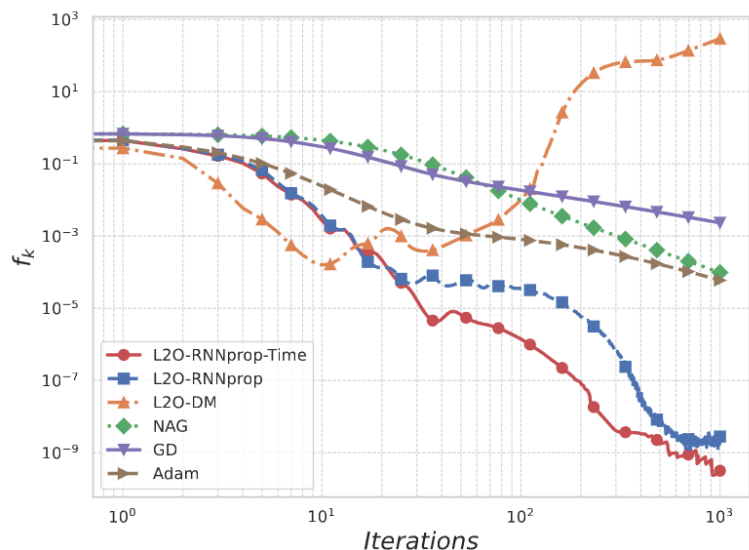
- 基于合成数据构造**逻辑回归**问题并应用L2O
- 加速达到**特定次优点**的时间
- 在**规模大4倍**的测试例子上可保持领先

$$S_k = \frac{\nabla f(x_{k+1})^\top d_k}{f(x_{k+1}) - f(x_k)}$$

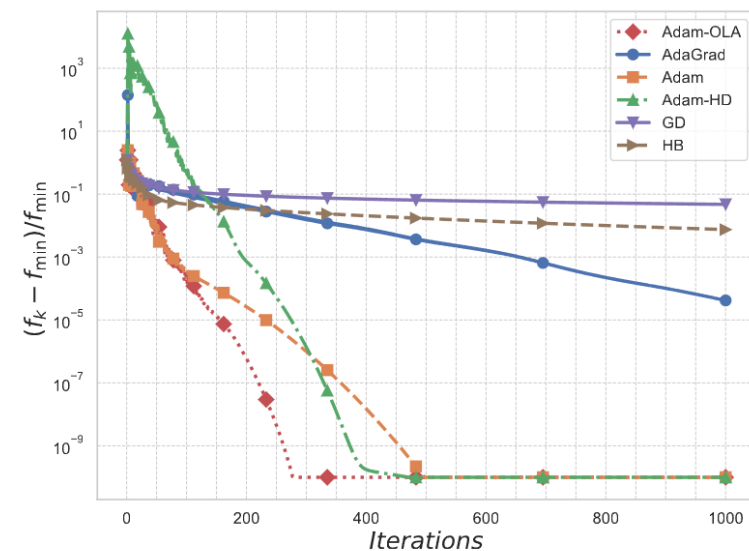
$$\alpha_{k+1} = \alpha_k - \eta_{\text{adapt}} S_k$$

Adam-OLA在Smooth SVM问题中加速了收敛

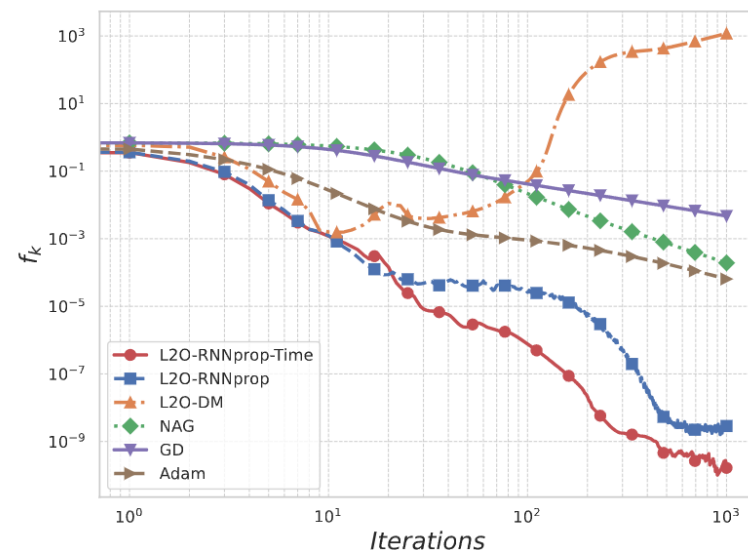
- 计算函数值**下降特定量所需步数**
- 将步数视为学习率的函数，求导并**更新学习率**
- 重复即得**Online Learning Rate Adaptation**



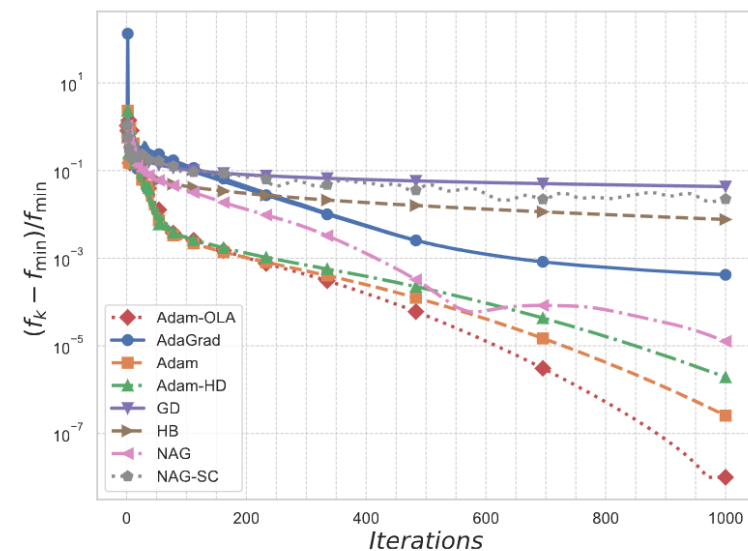
(a) Train and test on the same size problems.



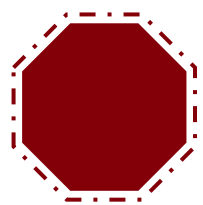
(a) a1a



(b) Test on 4x larger problems than training.



(b) a2a



二元整数规划MCPG



二元整数规划

二元整数规划问题:

$$\begin{aligned} \min f(x) \\ \text{s.t. } x \in \{-1, 1\}^n \end{aligned}$$

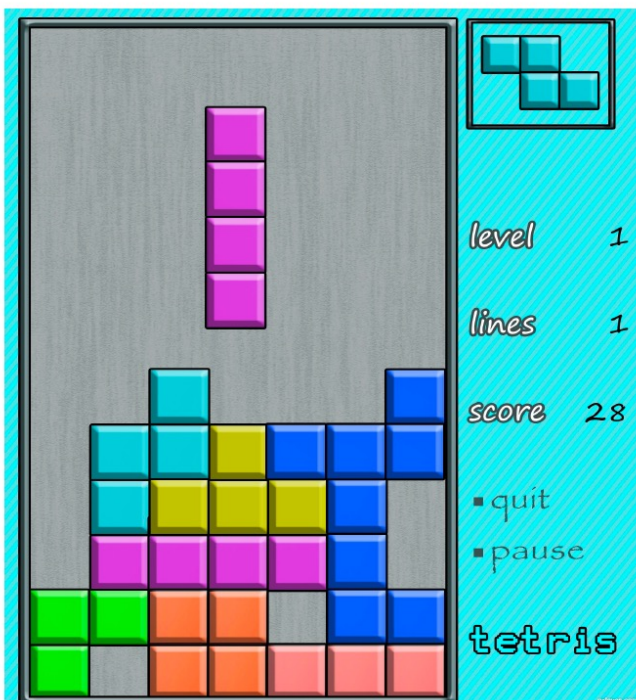
求解概率模型:

$$\min_{\theta} \mathbb{E}_{x \sim p_{\theta}} [f(x)]$$

NP难的整数规划基本问题

现有求解器难以通用求解

工业界求解大规模问题的算法需求



$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \text{ with } s_0 \sim \rho_0, a_t \sim \pi(\cdot | s_t), s_{t+1} \sim P(\cdot | s_t, a_t).$$

概率表示降低复杂度

精确罚函数处理额外约束

概率模型

• 构建参数化模型

MCMC采样

• 利用GPU快速采集大规模样本

计算目标函数

• 将计算过程拆解为矩阵算子

策略梯度优化

解空间的高效大规模探索

局部搜索提升解的质量

概率方法

概率表示整数规划最优解

目标分布

$$q^*(x) = \frac{1}{|\mathcal{X}^*|} \mathbf{1}_{\mathcal{X}^*}(x) = \begin{cases} \frac{1}{|\mathcal{X}^*|}, & x \in \mathcal{X}^*, \\ 0, & x \notin \mathcal{X}^*. \end{cases}$$

Gibbs分布

$$q_\lambda(x) = \frac{1}{Z_\lambda} \exp\left(-\frac{f(x)}{\lambda}\right) \text{ 其中 } Z_\lambda = \sum_{x \in \mathcal{B}_n} \exp\left(-\frac{f(x)}{\lambda}\right)$$

参数化分布

$p_\theta(x)$ 可以用以 θ 为参数的神经网络表示的分布 ...

神经网络训练

损失函数的梯度

$$\nabla_\theta L_\lambda(\theta) = \mathbb{E}_{p_\theta} [A_\lambda(x; \theta, c) \nabla_\theta \log p_\theta(x)]$$

类似于强化学习中的策略梯度!

其中 A_λ 是优势函数, 定义为

$$A_\lambda(x; \theta, c) := f(x) + \lambda \log p_\theta(x) - c$$

c 是常数, 可取为 $c = \mathbb{E}_{p_\theta}[f(x)]$

Gibbs分布
逼近
目标分布

参数化分布
逼近
Gibbs分布

通过分布逼近降低复杂度

当 $\lambda \rightarrow 0$ 时,

$$q_\lambda(x) = \frac{1}{Z_\lambda} \exp\left(-\frac{f(x)}{\lambda}\right) \rightarrow \frac{1}{|\mathcal{X}^*|} \mathbf{1}_{\mathcal{X}^*}(x) = q^*(x), \quad x \in \mathcal{B}_n$$

引入 KL 散度, 通过优化参数 θ 来极小化分布之间的差异

$$\begin{aligned} \text{KL}(p_\theta \| q_\lambda) &= \frac{1}{\lambda} \sum_{x \in \mathcal{B}_n} p_\theta(x) f(x) + \sum_{x \in \mathcal{B}_n} p_\theta(x) \log p_\theta(x) + \log Z_\lambda \\ &= \frac{1}{\lambda} (\mathbb{E}_{p_\theta} [f(x)] + \lambda \mathbb{E}_{p_\theta} [\log p_\theta(x)]) + \log Z_\lambda. \end{aligned}$$

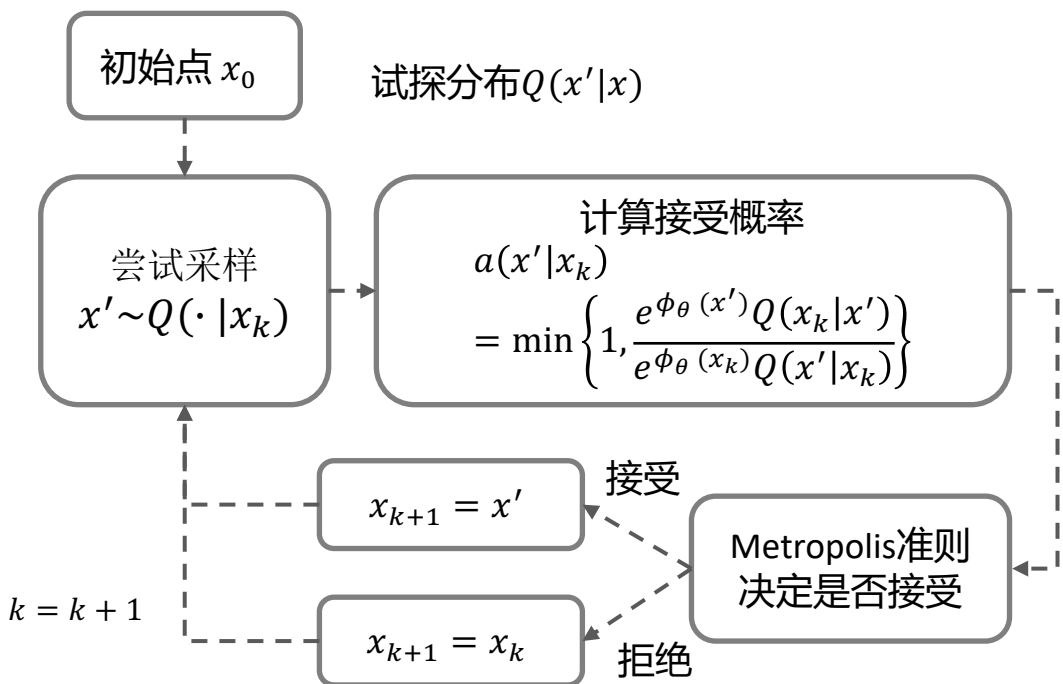
Z_λ 是常数

概率模型的损失函数

$$\min_{\theta} L_\lambda(\theta) = \mathbb{E}_{p_\theta} [f(x)] + \lambda \mathbb{E}_{p_\theta} [\log p_\theta(x)]$$

采样与局部搜索

马尔可夫蒙特卡洛采样



平衡“探索”与“利用”

只需非归一化的概率分布

GPU上大规模并行

平均场近似提高采样效率

滤波函数局部搜索策略

- 设 x 有邻域 $\mathcal{N}(x)$, 定义滤波函数

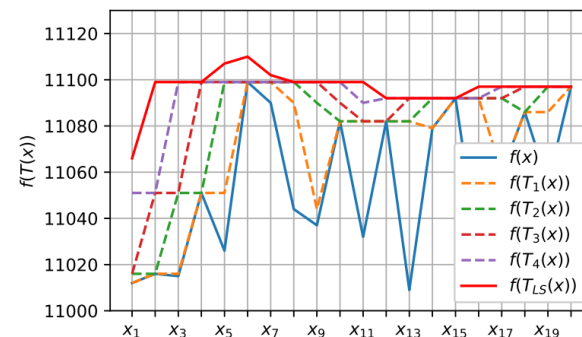
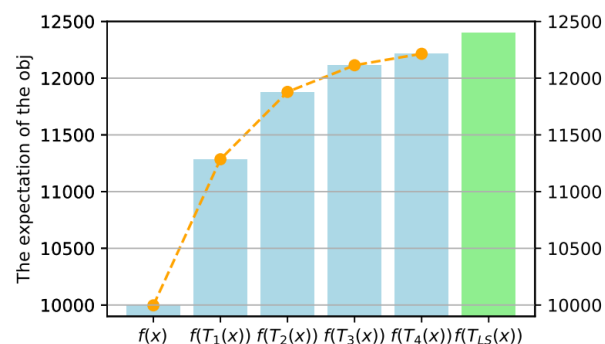
$$T(x) \in \arg \min_{\hat{x} \in \mathcal{N}(x)} f(\hat{x})$$

- 例1: 自然邻域中的最优解

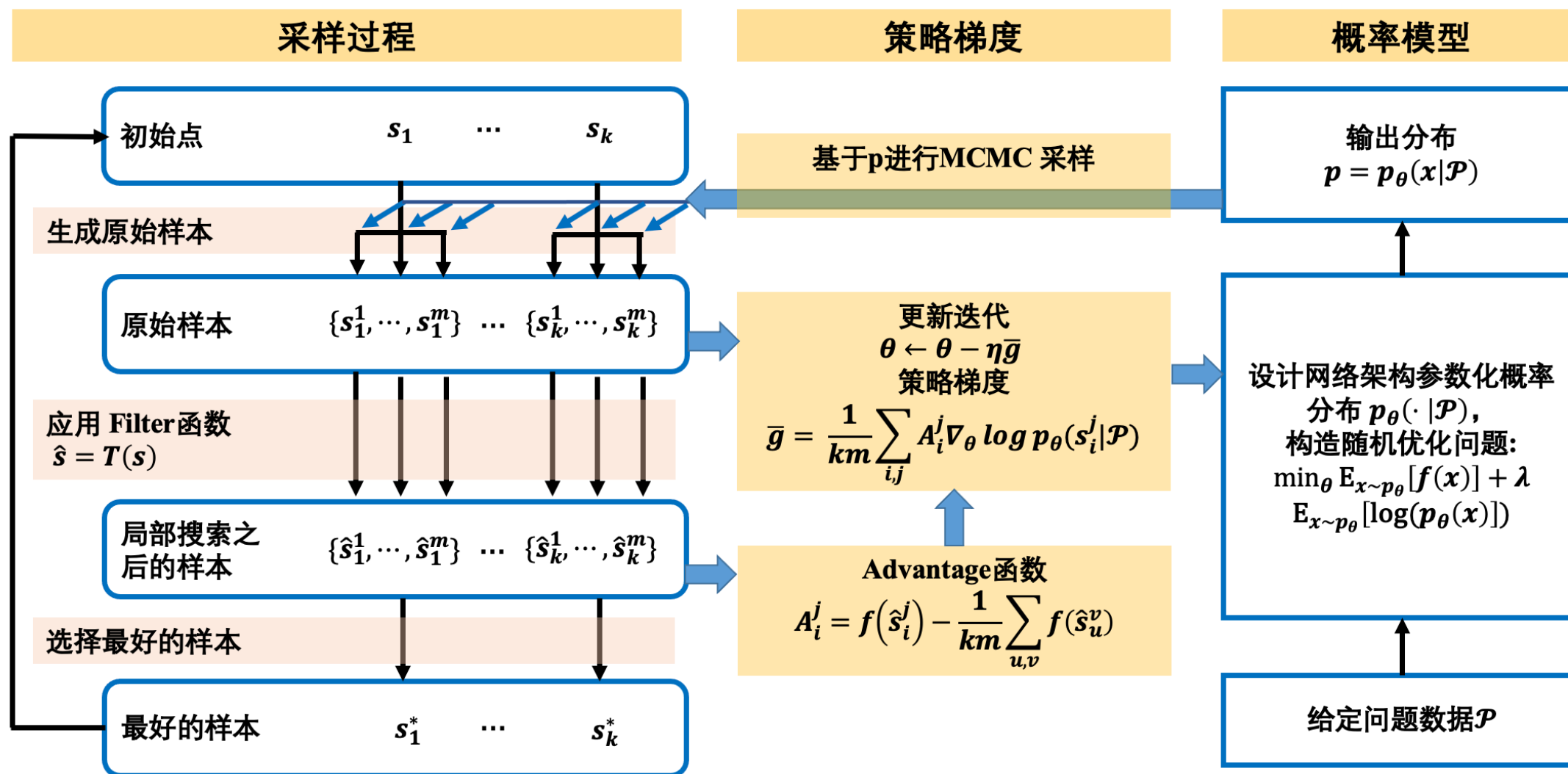
$$T_k(x) = \arg \min_{\|\hat{x} - x\|_1 \leq 2k} f(\hat{x}), \quad \mathcal{N}(x) = \{\hat{x} \mid \|\hat{x} - x\|_1 \leq 2k\}$$

- 例2: 算法作为滤波函数 $T_{LS}(x) = \text{LocalSearch}_f(x)$

- 滤波函数 T 将 x 投影到邻域内的更优解



二元整数规划: MCPG架构



算法收敛性

假设1: 策略正则条件

- 设 $\phi(x; \theta) = \log p_{\theta}(x|P)$, 存在常数 $M_1, M_2, M_3 > 0$, 使得对于任意 x , 下列条件成立
 - 1 $\sup_{\theta \in \mathbb{R}^n} |\phi(x; \theta)| \leq M_1$,
 - 2 $\sup_{\theta \in \mathbb{R}^n} \|\nabla_{\theta} \phi(x; \theta)\| \leq M_2$,
 - 3 $\|\nabla_{\theta_1} \phi(x; \theta) - \nabla_{\theta_2} \phi(x; \theta)\| \leq M_3 \|\theta_1 - \theta_2\|, \forall \theta_1, \theta_2 \in \mathbb{R}^n$

假设2: 谱间隙

- 设 P 为有限状态时间齐次马尔可夫链的转移矩阵, 则 P 的谱间隙定义为

$$\gamma(P) := \frac{1 - \max\{|\lambda_2(P)|, |\lambda_{2^n}(P)|\}}{2} \in (0, 1]$$

- 其中 $\lambda_i(P)$ 是矩阵 P 的第 i 个最大的特征值。
- 假设对于任意 θ , 存在正的下界 $\gamma(P_{\theta}) \geq \gamma > 0$

Theorem (一阶收敛性)

假设以上假设成立, 且 $\{\theta^t\}$ 由 MCPG 生成。如果步长满足 $\eta^t \leq \frac{1}{2L}$, 则对于任意 τ , 有

$$\min_{1 \leq t \leq \tau} \mathbb{E} \left[\|\nabla_{\theta} L_{\lambda}(\theta^t)\|^2 \right] \leq O \left(\frac{1}{\sum_{t=1}^{\tau} \eta^t} + \frac{\sum_{t=1}^{\tau} (\eta^t)^2}{mk \sum_{t=1}^{\tau} \eta^t} + \frac{1}{m^2} \right).$$

特别地, 如果步长选择为 $\eta^t = \frac{c\sqrt{mk}}{\sqrt{t}}$, 其中 $c \leq \frac{1}{2l}$, 则有

$$\min_{1 \leq t \leq \tau} \mathbb{E} \left[\|\nabla_{\theta} L_{\lambda}(\theta^t)\|^2 \right] \leq O \left(\frac{\log \tau}{\sqrt{mk\tau}} + \frac{1}{m^2} \right).$$

最大割问题

最大割问题的二元整数规划形式:

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} w_{ij}(1 - x_i x_j), \\ \text{s.t.} \quad & x \in \{-1, 1\}^n. \end{aligned}$$

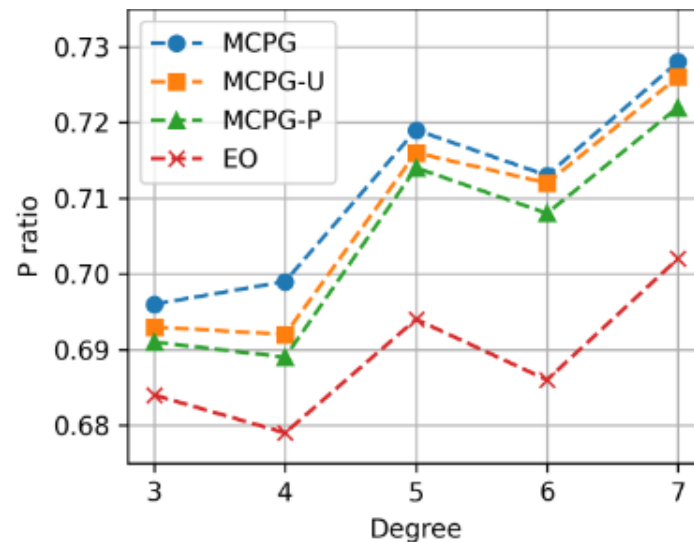
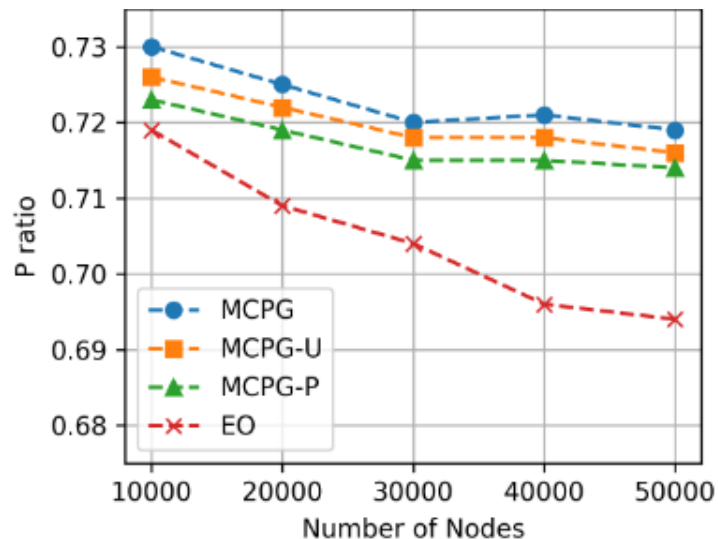
MCPG在大规模问题上有稳定的优秀表现

- MCPG在规模为50000的随机图上的表现与10000规模仍十分接近, 没有显著衰减

MCPG在Gset数据集上取得最佳结果

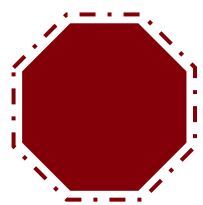
- 在Gset实例中, MCPG找到了所有已知最佳结果, 且在G55和G70上得到优于所有已知结果的解。
- 短时间内, MCPG能获得十分接近最优解的结果。
- 通过设计更复杂的Filter函数, MCPG的算法性能能获得进一步提升 (见右图MCPG - E)

大规模随机生成图结果



Gset实例结果

图			BLS	MCPG	DSDP	MCPG (限时)		MCPG-E (限时)	
名称	节点	边				差距	时间	差距	时间
G14	800	4694	3064	3064	2922	0.02%	28	0.00%	93
G15	800	4661	3050	3050	2938	0.01%	28	0.00%	96
G22	2000	19990	13359	13359	12960	0.10%	55	0.05%	205
G49	3000	6000	6000	6000	6000	0.00%	107	0.00%	119
G50	3000	6000	5880	5880	5880	0.03%	107	0.00%	119
G55	5000	12468	10294	10296	9960	0.47%	145	0.34%	294
G70	10000	9999	9541	9595	9456	0.86%	274	0.44%	480



路径规划LMask



约束路径规划的学习优化方法: LMask

NP难的整数规划基本问题

困难约束导致解可行性难以保证

工业界快速求解路径规划的算法需求

路径规划问题:

$$\begin{aligned} \min_{\pi \in \Pi} \quad & f(\pi; \mathcal{P}) \\ \text{s.t.} \quad & c(\pi; \mathcal{P}) \leq 0, \\ & d(\pi; \mathcal{P}) = 0, \end{aligned}$$

求解概率模型:

$$L(\theta; \mathcal{P}) := \mathbb{E}_{p_\theta(\cdot; \mathcal{P})}[f(\pi; \mathcal{P})] + \lambda \mathbb{E}_{p_\theta(\cdot; \mathcal{P})}[\log p_\theta(\pi; \mathcal{P})]$$

$$p_\theta(\pi_t | \pi_{1:t-1}; \mathcal{P}) = \frac{e^{\phi_\theta(\pi_t | \pi_{1:t-1}; \mathcal{P})} M(\pi_t | \pi_{1:t-1}; \mathcal{P})}{\sum_{k=0}^n e^{\phi_\theta(k | \pi_{1:t-1}; \mathcal{P})} M(k | \pi_{1:t-1}; \mathcal{P})}$$

- 整数规划建模形式复杂度高, 求解效率低
- 已有学习优化方法的掩码机制求解困难约束的路径问题

不能保证解的可行性

带时间窗的旅行商问题 (TSPTW)

$$\begin{aligned} \min \quad & f(\pi; \mathcal{P}) = \sum_{i=1}^n \|x_{\pi_i} - x_{\pi_{i+1}}\| \\ \text{s.t.} \quad & c_i(\pi; \mathcal{P}) = \sum_{t=1}^i t_{\pi_t, \pi_{t+1}} - l_i \leq 0, \quad i = 1, \dots, n, \\ & c_{n+i}(\pi; \mathcal{P}) = c_i - \sum_{t=1}^i t_{\pi_t, \pi_{t+1}} \leq 0, \quad i = 1, \dots, n, \\ & d_i(\pi; \mathcal{P}) = \sum_{t=1}^n \mathbb{1}_{\pi_t=i} - 1 = 0, \quad i = 1, \dots, n. \end{aligned}$$

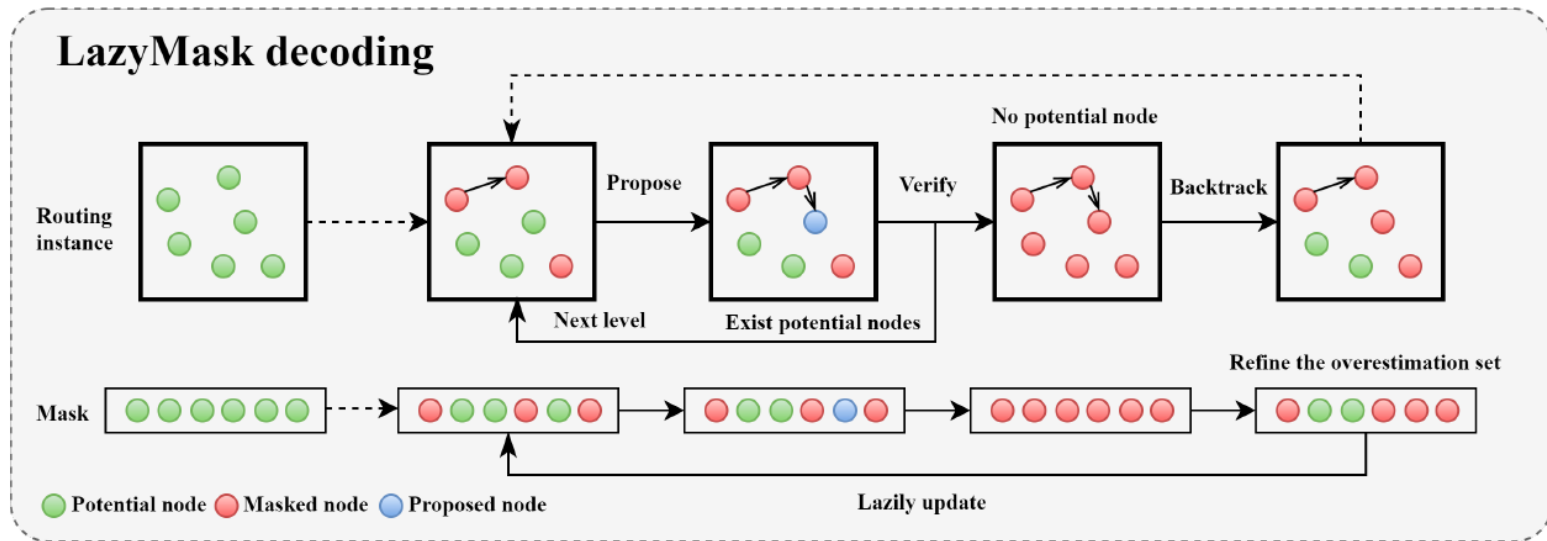
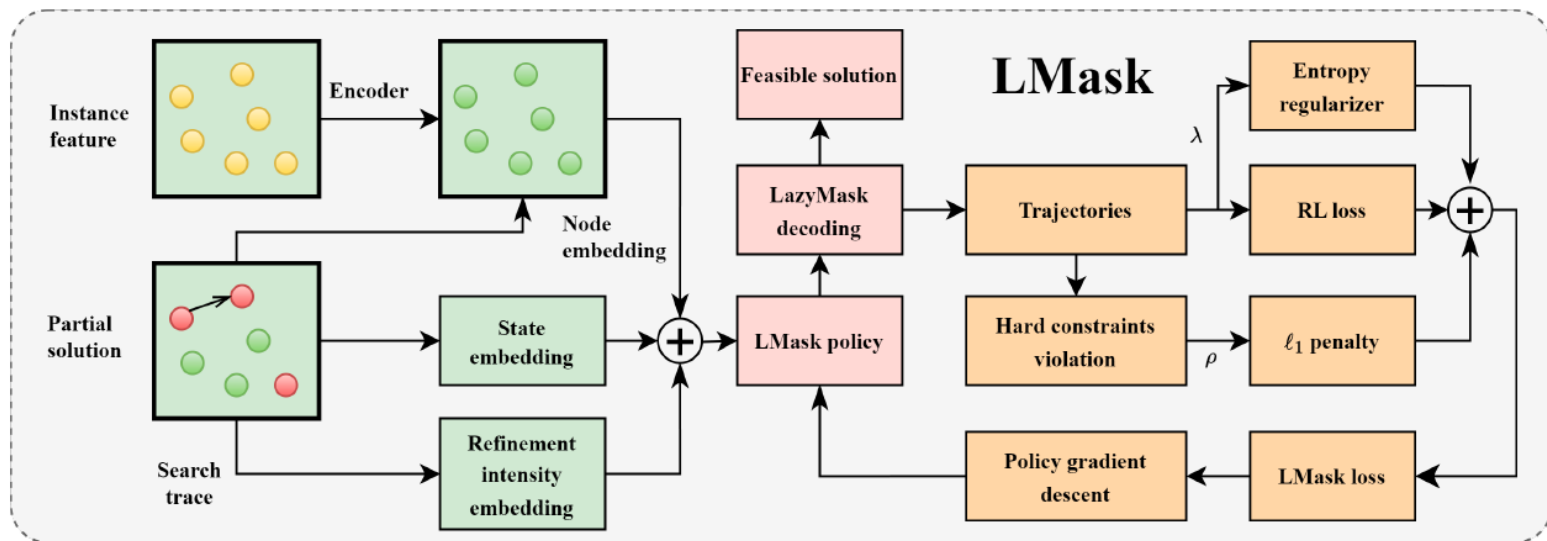
带吃水限制的旅行商问题 (TSPDL)

$$\begin{aligned} \min \quad & f(\pi; \mathcal{P}) = \sum_{i=1}^n \|x_{\pi_i} - x_{\pi_{i+1}}\| \\ \text{s.t.} \quad & c_i(\pi; \mathcal{P}) = \sum_{t=1}^i q_{\pi_t} - D_i \leq 0, \quad i = 1, \dots, n, \\ & d_i(\pi; \mathcal{P}) = \sum_{t=1}^n \mathbb{1}_{\pi_t=i} - 1 = 0, \quad i = 1, \dots, n. \end{aligned}$$

路径问题掩码示例



约束路径规划的学习优化方法: LMask



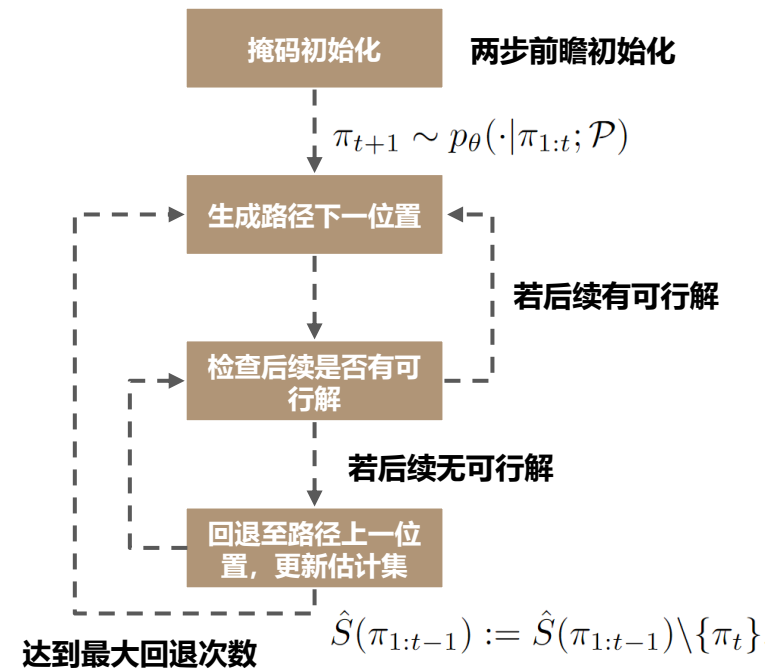
概率模型的损失函数

损失函数由以下三项组成:

- 路径成本损失
- 熵正则项
- 约束违反惩罚

$$\min \mathbb{E}_{\pi \sim p_{\theta}(\cdot; \mathcal{P})} \left[f(\pi; \mathcal{P}) + \rho \sum_{j=1}^J \max(c_j(\pi; \mathcal{P}), 0) + \lambda \log p_{\theta}(\pi; \mathcal{P}) \right]$$

LazyMask解码算法



约束路径规划的学习优化方法: LMask

- 传统路径规划求解器

PyVRP, LKH, ORTOOL

- 经典启发式贪婪算法

Greedy-L, Greedy-C

- 学习优化算法

PIP, PIP-D, **LMask**

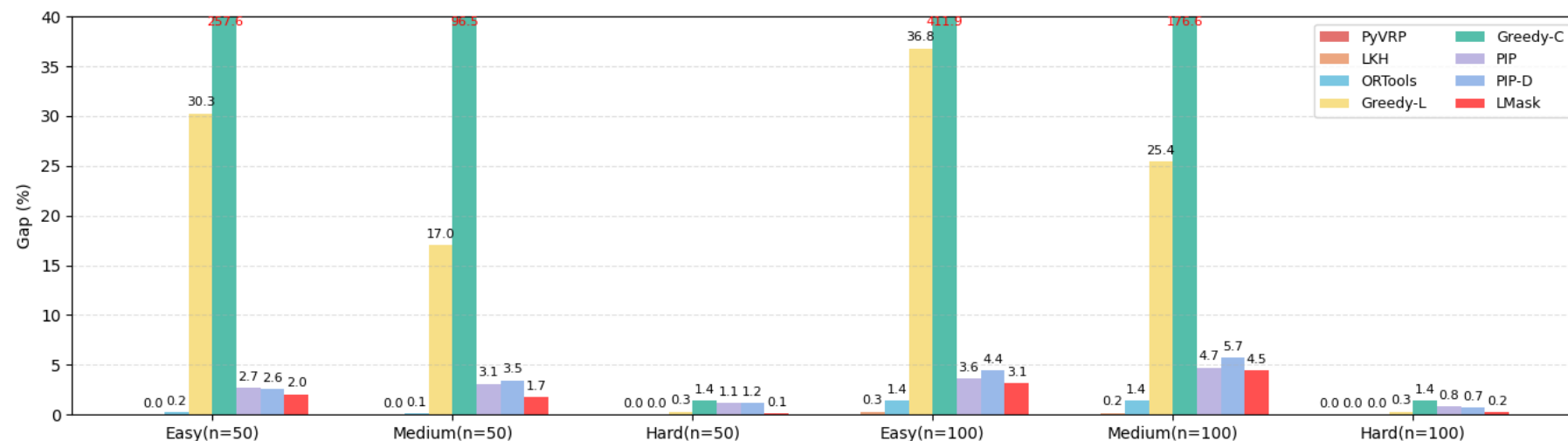
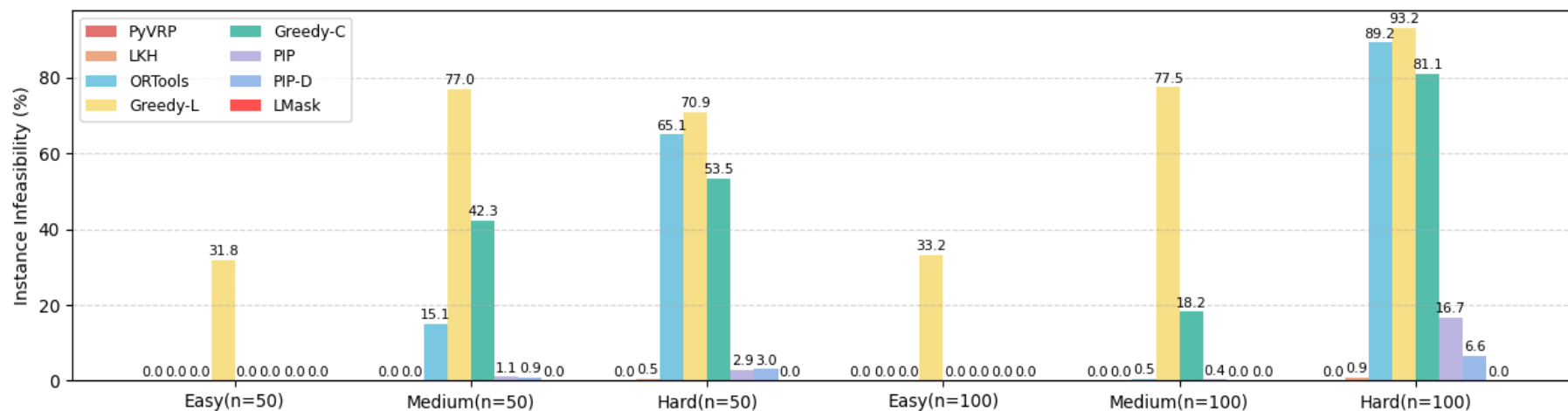
实现TSPTW算例基本可行

- LMask**不可行率接近**0.0%**, 大大超越已有学习优化算法

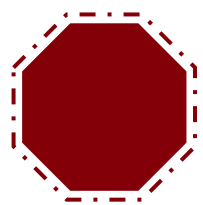
学习优化算法中最优间隙最低

- LMask**最优间隙相比其他学习优化算法显著降低

时间相比传统求解器大大缩短



- 传统求解器PyVRP和ORTools求解10000个100规模的TSPTW实例需要**4小时**以上
- 学习优化方法**LMask**只需要**20秒**可以更好求解10000个实例, 其他学习方法需要**30秒**以上



异构调度WeCAN



面向DAG异构调度问题的学习优化方法

DAG异构调度问题序列建模

$$\min_{x=(s,c)} f(x) := \max_{v \in V} [s(v) + t(v)/K_{acc}(v, c(v))],$$

$$\text{s.t. } s(v) + t(v)/K_{acc}(v, c(v)) \leq s(w), \forall (v, w) \in E,$$

$$\sum_{v \in F(t,c)} \rho(v) \leq \lambda(c), \forall t \geq 0, c \in C,$$

$$K_{acc}(v, c(v)) > 0, \forall v \in V,$$

$$s(v) \geq 0, \forall v \in V.$$

$$F(t, c) := \{v \in V | s(v) \leq t; s(v) + t(v)/K_{acc}(v, c(v)) > t; c(v) = c\}.$$

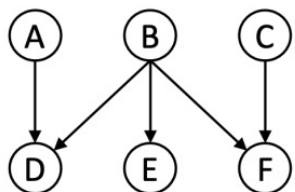
依赖约束

资源约束

匹配约束

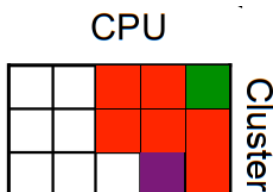
依赖约束

任务运行顺序满足有向图依赖关系



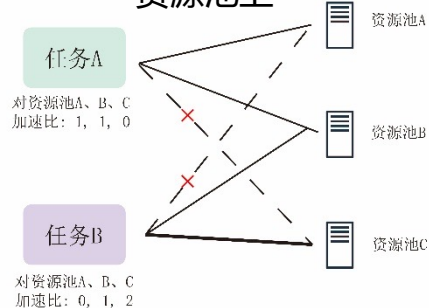
资源约束

任一时间任何资源池上运行中的任务消耗资源量总和不超过其资源总量



匹配约束

任务均分配到匹配的资源池上



问题规模 (TPC-H-100, 3 pool)

	变量		约束
序列建模	2.0×10^3	$5.0 \times 10^3 + E $	$\leq 1.0 \times 10^6$
MILP 建模	6.0×10^6		1.2×10^7

混合整数规划(MILP)建模

$$\min t_{max},$$

$$\text{s.t. } s_i + t_i \sum_k v_i^k a_{ik} \leq t_{max}, \forall i \in V,$$

$$s_i + t_i \sum_{k \in C} v_i^k a_{ik} \leq s_j, \forall (i, j) \in E,$$

$$s_i < s_j + C_1^{up} u_{ij}, \forall i, j \in V,$$

$$s_j \leq s_i + C_1^{up} (1 - u_{ij}), \forall i, j \in V,$$

$$s_i < s_j + t_j \sum_{k \in C} v_j^k a_{jk} + C_1^{up} (1 - w_{ij}), \forall i, j \in V,$$

$$s_j + t_j \sum_{k \in C} v_j^k a_{jk} \leq s_i + C_1^{up} w_{ij}, \forall i, j \in V,$$

$$u_{ij} + w_{ij} \leq 1 + x_{ij}, \forall i, j \in V,$$

$$u_{ij} + w_{ij} \geq 2x_{ij}, \forall i, j \in V,$$

$$x_{ij} + v_j^k \geq 2y_{ijk}, \forall i, j \in V, k \in C,$$

$$x_{ij} + v_j^k \leq 1 + y_{ijk}, \forall i, j \in V, k \in C,$$

$$\sum_k v_i^k = 1, \forall i \in V, k \in C,$$

$$1 - v_i^k + C_0^{up} - a_{ik} > 0, \forall i \in V, k \in C,$$

$$\rho_l^i + \sum_{j \neq i} y_{ijk} \rho_l^j \leq C_1^{up} (1 - v_i^k) + \lambda_l^k, \forall i \in V, k \in C, l \in \{1, \dots, r\},$$

$$s_i \geq 0, t_{max} \geq 0, u_{ij}, w_{ij}, x_{ij}, y_{ijk}, v_i^k \in \{0, 1\}, \forall i, j \in V, k \in C.$$

依赖约束

刻画顺序
整数变量

匹配约束

资源约束

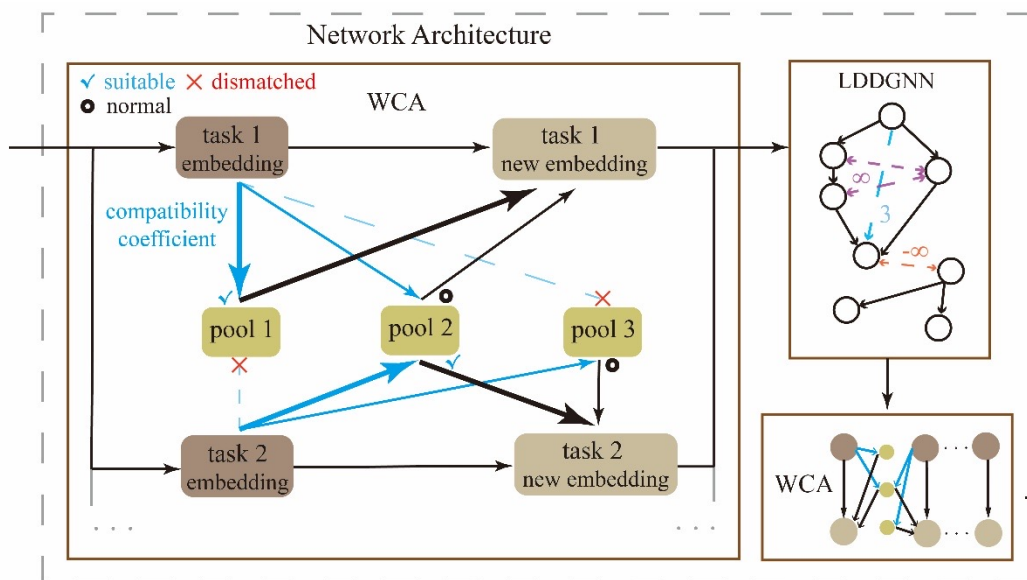
问题求解难点

难以直接生成问题的可行解的实数变量

有向图约束复杂且形式多样

变量和约束规模极端庞大

面向DAG异构调度问题的学习优化方法

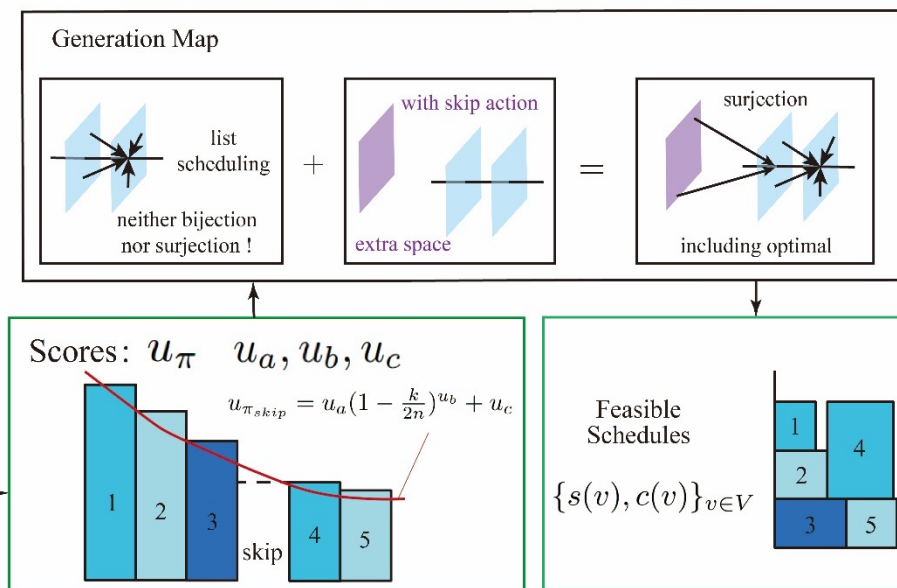


神经网络架构模块

调度问题 $(G, V, C, \rho, t, \lambda, K_{acc})$

任务-资源池对分数 u_π 与跳过系数 u_a, u_b, u_c

- 加权交叉注意力模块将所有兼容系数嵌入注意力偏置，对信息**无损编码**，同时保证单个网络对各类资源环境（不同池子数等）均可直接使用的**可扩展性**
- 基于最长有向距离的图神经网络对依赖关系高效编码，保证信息长距离传递并保留偏序关系



加入跳过决策的生成映射

分数与跳过系数 \rightarrow 完整解 $\{s(v), c(v)\}_{v \in V}$

生成映射：基于静态的任务-资源池分数与动态计算的跳过分数，选择未违反约束的分数最高的决策。若选择任务-资源池对 (v, c) ，赋予所选任务的开始时间 $s(v)=t$ ，分配资源池 $c(v)=c$ 。若选择跳过，更新 t 为最早的下个任务完成时间

- 引入的跳过决策修补了列表调度**可能遗漏最优解的缺陷**，在重型任务等**特定场景上大幅提升求解精度**
- 跳过分数的扩展方式将“坏解”聚拢，降低强化学习难度

面向DAG异构调度问题的学习优化方法

TPC-H工业数据集

Table 1: Experimental results on TPC-H datasets with standard deviation among random seed.

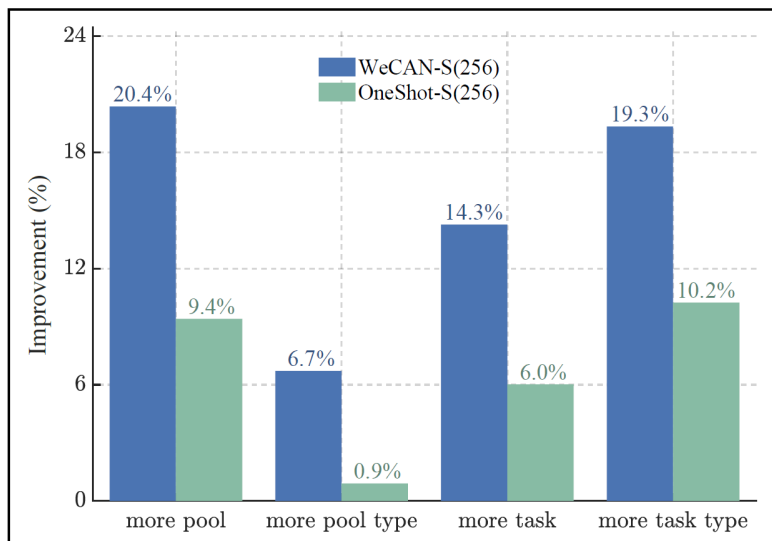
	TPC-H-30, 3 pools		TPC-H-50, 3 pools		TPC-H-100, 3 pools	
	MakeSpan	Time	MakeSpan	Time	MakeSpan	Time
SFT	27404	0.23	49172	0.78	84986	3.08
MOPNR	25052	0.30	43545	0.99	77362	3.34
CP	23869	0.29	41597	0.90	74364	3.35
HEFT	23177	0.18	39315	0.54	70137	1.86
Tetris	23170	0.21	38654	0.62	71296	2.13
PPO-BiHyb	21941	20.48	36333	55.74	67695	179.19
One-Shot-S(256)	20399 ± 181	2.26	35561 ± 108	4.16	66173 ± 180	9.85
WeCAN-Greedy	19578	0.15	33428	0.50	62587	1.72
WeCAN-S(64)	19053 ± 28	1.54	32912 ± 40	2.86	61662 ± 118	5.26
WeCAN-S(256)	18964 ± 10	2.43	32814 ± 47	4.39	61373 ± 28	10.43

计算图合成数据集

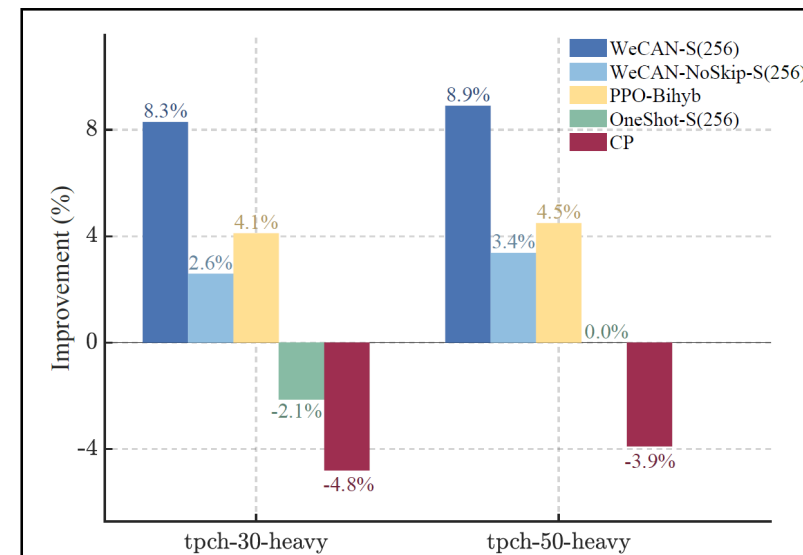
Table 2: Experimental results on Computation Graphs datasets with 500 tasks.

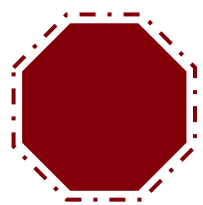
	Erdős-Rényi		Layer Graphs		Stochastic Block	
	MakeSpan	Time	MakeSpan	Time	MakeSpan	Time
SFT	13316.8	0.81	16158.4	0.34	14407.9	0.53
MOPNR	12771	1.07	14714	0.38	13148	0.68
Tetris	13084	0.52	14271	0.44	13666	0.64
CP	12457	1.08	14797	0.40	13388	0.74
HEFT	11098	0.55	12428	0.75	11260	0.57
PPO-BiHyb	10795	65.51	11883	73.7	10885	73.7
One-Shot-S(256)	11144 ± 29	4.45	12277 ± 49	3.83	11457 ± 64	4.00
WeCAN-Greedy	10270	0.57	11173	0.26	10539	0.41
WeCAN-S(64)	10115 ± 10	3.21	10862 ± 29	3.07	10074 ± 18	3.06
WeCAN-S(256)	10083 ± 13	4.94	10752 ± 27	4.30	10019 ± 12	4.58

良好的泛化性



跳过决策 保证最优可达





二次指派PLMA



面向二次指派问题的学习优化方法：PLMA

二次指派问题置换形式：

$$\min_{\pi \in \Pi_n} f(\pi; \mathcal{P}) := \sum_{i=1}^n \sum_{j=1}^n F_{ij} D_{\pi(i)\pi(j)}.$$

二次指派问题矩阵形式：

$$\min_{X \in \{0,1\}^{n \times n}} \langle F, XDX^T \rangle, \quad \text{s.t.} \quad X\mathbf{1} = \mathbf{1}, X^T\mathbf{1} = \mathbf{1}.$$

问题特点

置换空间复杂度高
常规模型采样低效

矩阵形式问题信息
D, F 特征提取困难

置换空间上的能量模型：

$$p_{\theta}(\pi | \mathcal{P}) \propto \exp(\Phi_{\theta}(\pi)), \quad \Phi_{\theta}(\pi) = \sum_{i=1}^n \phi_{i, \pi(i)}.$$

$O(1)$ 时间邻域搜索：

$$\exp(\Phi_{\theta}(\pi') - \Phi_{\theta}(\pi)) = \exp(\phi_{a, \pi(b)} + \phi_{b, \pi(a)} - \phi_{a, \pi(a)} - \phi_{b, \pi(b)})$$

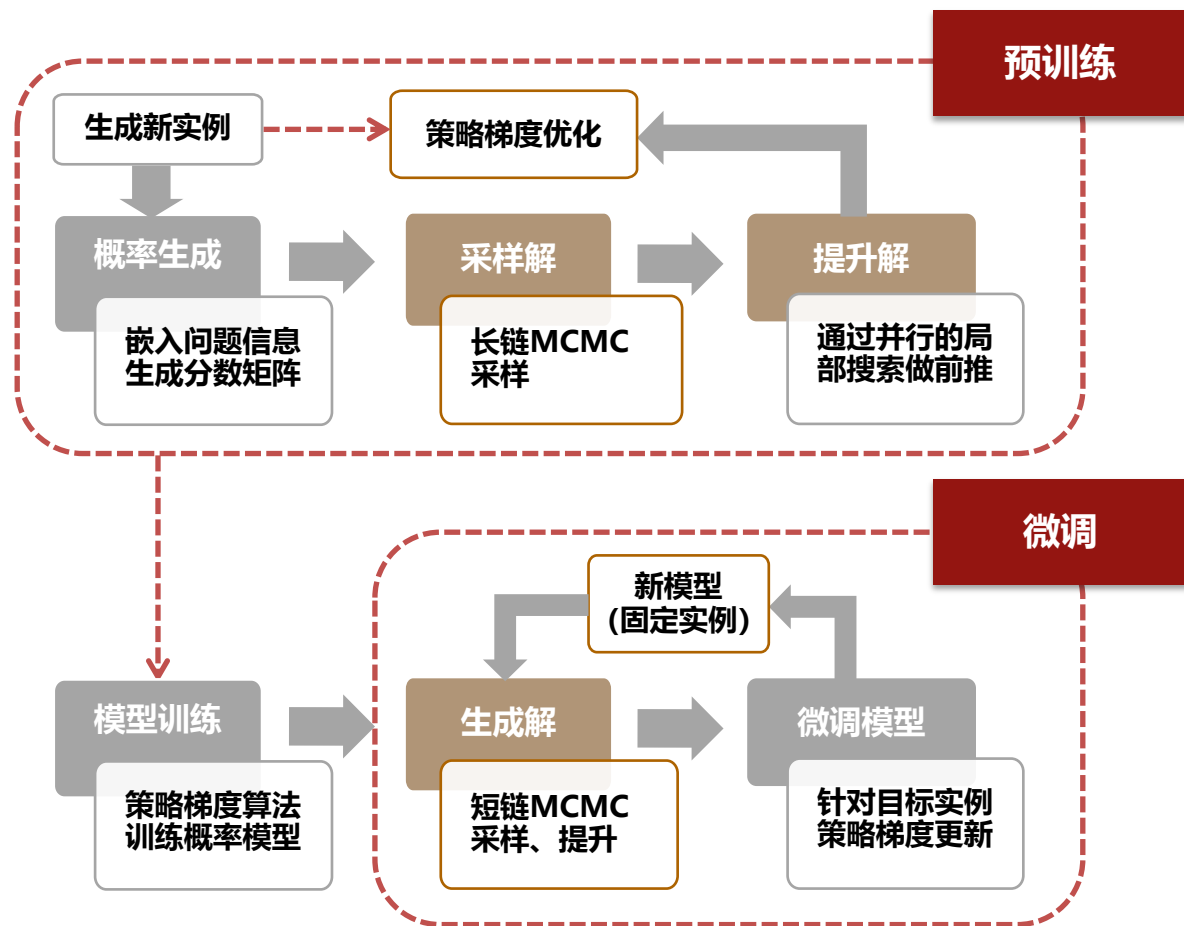
高效梯度估计：

$$\nabla_{\theta} \mathbb{E}_{\pi \sim p_{\theta}} [g(\pi)] \approx \frac{1}{N-1} \sum_{i=1}^N \left(g(\pi_i) - \frac{1}{N} \sum_{i=1}^N g(\pi_i) \right) \nabla_{\theta} \Phi_{\theta}(\pi_i).$$

前推训练目标：

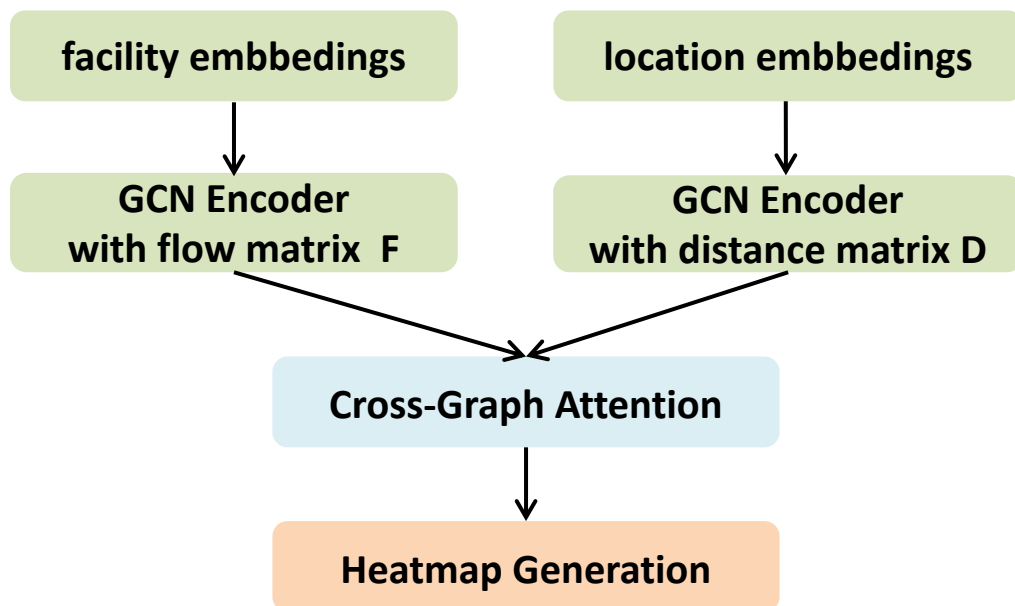
$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) := \mathbb{E}_{\mathcal{P} \sim \Gamma} \mathbb{E}_{\sigma \sim T_{\#p_{\theta}}(\cdot | \mathcal{P})} [f(\sigma; \mathcal{P})] = \mathbb{E}_{\mathcal{P} \sim \Gamma} \mathbb{E}_{\pi \sim p_{\theta}(\cdot | \mathcal{P})} [f(T(\pi); \mathcal{P})].$$

预训练、微调两阶段结合：



面向二次指派问题的学习优化方法：PLMA

网络架构

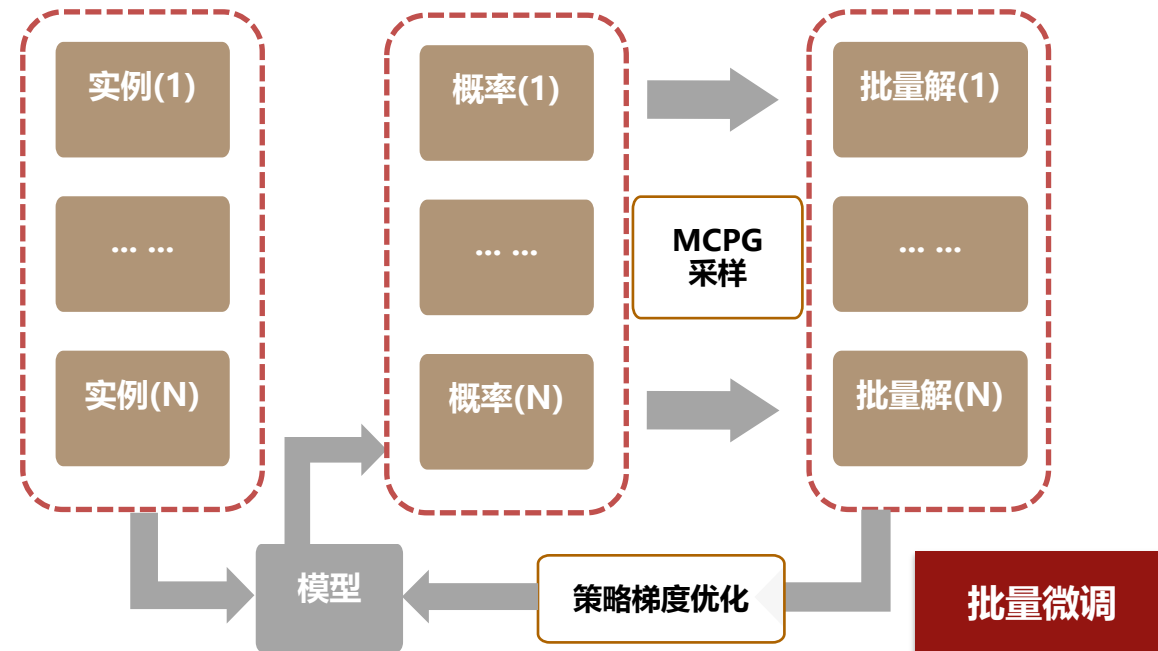


- **图内信息嵌入**：将矩阵D/F视为带权邻接矩阵分别进行图卷积网络，得到两组节点特征
- **图间信息交互**：对以上两组节点特征做交叉注意力，进行信息的跨图传递
- **生成匹配分数**：对两组特征做内积后接非线性激活
- **核心优势**：该架构无需构建 n^2 节点关联图，保证其可扩展性

热启动MCMC微调

相较于MCPG框架，针对置换问题做了以下调整：

- **原生可行**：直接在置换空间建模概率，以2-swap交换作为基础的改进动作，确保采样可行性与高效性
- **高效并行**：微调过程以批处理的方式对多个实例并行优化，共享同一套网络参数更新，提升了计算效率
- **支持预训练**：预训练能够为微调提供一个良好的初始概率分布



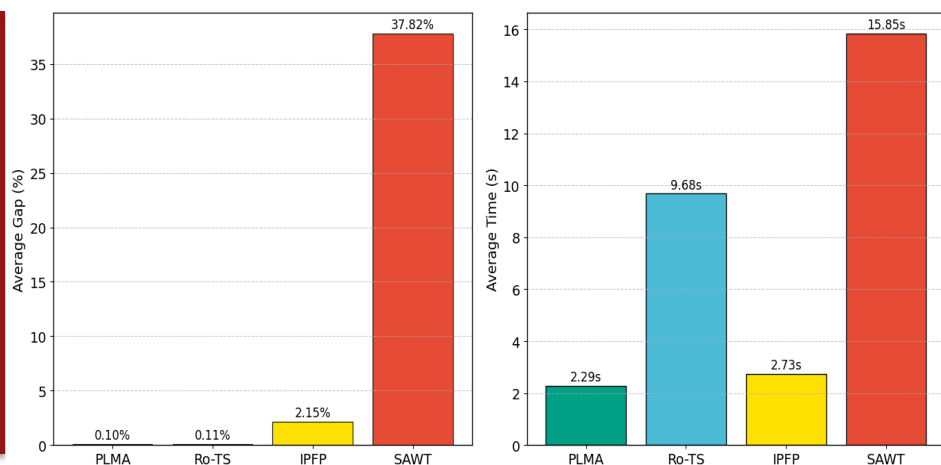
面向二次指派问题的学习优化方法：PLMA

两类合成数据集

Table 1: Performance comparison on synthetic datasets (256 instances).

Algorithm	QAP20			QAP50			QAP100		
	Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
<i>Geometrically Structured Datasets</i>									
Ro-TS (1k)	54.38	0.01%	17.04s	375.99	0.14%	4m35s	1593.27	0.13%	38m56s
Ro-TS (5k)	54.37	0.00%	1m25s	375.48	0.00%	22m53s	1591.25	0.00%	3h15m
IPFP	55.11	1.37%	2.04s	378.76	0.88%	11.47s	1600.27	0.57%	1m34s
IPFP (10)	54.54	0.31%	20.97s	376.60	0.30%	2m30s	1594.76	0.22%	17m34s
NGM	62.93	15.87%	24.78s	429.69	14.46%	1m16s	1773.71	11.47%	2m29s
SAWT (10k)	54.72	0.64%	3m41s	380.92	1.45%	5m36s	1617.30	1.64%	10m43s
PLMA ($T = 1$)	54.63	0.48%	0.06s	379.79	1.15%	0.41s	1607.84	1.04%	4.27s
PLMA ($T = 50$)	54.37	0.00%	2.57s	375.55	0.20%	19.88s	1591.73	0.03%	3m30s
PLMA ($T = 200$)	54.37	0.00%	9.36s	375.48	0.00%	1m19s	1591.23	0.00%	13m58s
<i>Uniformly Random Datasets</i>									
Ro-TS(1k)	76.61	0.07%	17.56s	523.08	0.22%	4m36s	2195.98	0.13%	38m59s
Ro-TS(5k)	76.56	0.00%	1m27s	521.91	0.00%	22m59s	2193.16	0.00%	3h15m
IPFP	79.13	3.39%	2.12s	530.74	1.69%	7.45s	2211.38	0.83%	41.95s
IPFP(25)	77.60	1.37%	54.70s	526.96	0.97%	4m20s	2203.29	0.46%	19m20s
NGM	88.34	15.49%	25.08s	594.99	14.01%	1m17s	2438.52	11.19%	2m29s
PLMA ($T = 1$)	78.23	2.20%	0.06s	538.42	3.17%	0.40s	2243.43	2.29%	4.32s
PLMA ($T = 50$)	76.59	0.05%	2.47s	523.95	0.40%	19.51s	2200.40	0.34%	3m30s
PLMA ($T = 200$)	76.56	0.00%	9.60s	521.83	-0.01%	1m18s	2193.13	0.00%	14m1s

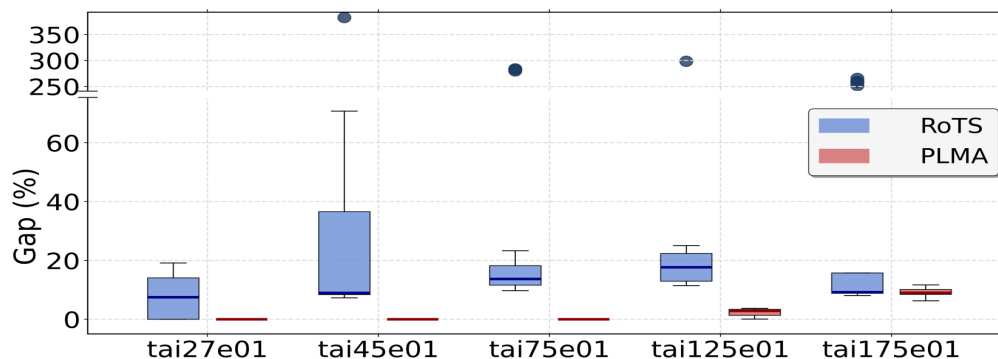
基准数据集 QAP LIB



鲁棒性测试

Table 4: Grouped results on Taixxey instances.

Class	Ro-TS			IPFP			PLMA		
	mean	[min, max]	Time (s)	mean	[min, max]	Time (s)	mean	[min, max]	Time (s)
tai27e	41.50%	[0.11%, 221.08%]	0.57	19.47%	[6.14%, 34.11%]	0.38	0.00%	[0.00%, 0.00%]	0.08
tai45e	101.89%	[1.00%, 400.60%]	3.83	22.01%	[8.26%, 36.98%]	1.03	0.03%	[0.00%, 0.35%]	0.28
tai75e	111.28%	[6.20%, 280.01%]	18.49	27.64%	[17.56%, 36.78%]	2.52	0.09%	[0.00%, 0.45%]	1.48
tai125e	82.53%	[7.65%, 265.54%]	72.52	26.93%	[20.64%, 32.67%]	8.52	2.67%	[-0.08%, 5.62%]	8.18
tai175e	67.86%	[9.11%, 260.98%]	158.18	23.32%	[17.70%, 28.11%]	16.86	9.11%	[5.61%, 12.04%]	14.63
Average	81.01%	[4.82%, 285.64%]	50.72	23.87%	[14.06%, 33.73%]	5.86	2.38%	[1.11%, 3.69%]	4.93



- PLMA($T=1$): **秒级速度**批量生成解, 精度远超以往学习方法
- PLMA($T=200$): 合成数据集优化间隙**0.00%**, 较传统方法速度大大提升 (QAP100上**14倍**, QAPLIB上**4倍**)
- 鲁棒性: 在Taixxey上无优化间隙大于100%的异常, 且平均间隙亦显著更小 (**2.38%** vs 81.01%)



谢 谢