面向供应链优化的分布式约束强化学习

Jaime Sabal Berm´udez[a], Antonio del Rio Chanona[b], Calvin Tsay[a,*]
[a]伦敦帝国理工学院计算系[b]伦敦帝国理工学院萨金特过程系统工程中心

arXiv:2302.01727v1 [cs.LG] 3 Feb 2023

摘要——本文研究了受约束条件（如生产和库存约束）的多周期供应链中的强化学习（RL）。我们提出了分布式约束策略优化（DCPO），这是一种旨在实现强化学习可靠约束满足的新方法。该方法基于约束策略优化（CPO），而后者会受到近似误差的影响，实际应用中因此可能收敛至不可行策略。为解决这一问题，我们将分布式强化学习的部分机制融入 DCPO。具体而言，我们使用输出离散分布的神经网络来表示回报和成本价值函数，并根据相关置信度调整成本。通过一项供应链案例研究，我们表明，DCPO 能够提高强化学习策略的收敛速度，并确保训练结束时可靠地满足约束。所提出的方法还提高了可预测性，大幅降低了不同运行次数之间回报的方差；这一结果对于策略梯度方法尤为重要，因为此类方法在训练过程中本身就会引入显著方差。

索引词—**安全强化学习、过程操作、库存管理**

I. 引言

近年来，现代社会的发展和运转凸显了高效供应链的重要性。库存管理领域致力于优化产品库存的订购、储存和销售。这个问题有许多方面颇具挑战性，同时解决它也能带来诸多益处，因而值得深入研究。挑战包括沿用过时的“遗留”流程，或未能有效利用现有物质资本。运筹学领域已采用多种方法来解决这一类问题，包括《动态规划》、线性规划和博弈论。然而，鉴于环境的复杂性，现代库存管理显然也适合采用强化学习（RL）进行优化。

过去几十年来，机器学习因能够模仿人类的学习方式，并复现认知推理的某些方面，而日益受到关注。强化学习（RL）为人工智能体提供了一个框架，使其能够通过与外部环境交互并不断更新自身的行为方式（即其策略）来学习。具体而言，每次交互都会使智能体进入一个新的状态，并产生可量化的奖励；智能体旨在通过一系列交互最大化累积奖励（即回报）。事实证明，强化学习在游戏等领域非常有效，能够在极其复杂的游戏中展现超越人类的表现；在机器人学、控制 [1], [2], 和调度 [3], [4] 等领域也表现出色，因为这些问题可能过于复杂，难以求得解析解。

供应链等现实环境可能会对智能体能够采取的动作施加物理或安全限制。本文针对这类情况提出一种强化学习框架，结合了约束强化学习和分布式强化学习的优势。约束强化学习算法将强化学习智能体的目标扩展为：除最大化期望回报 [5]之外，还要将期望成本控制在环境中约束所规定的某个阈值以下。相关地，分布式强化学习试图通过估计环境中所有状态的价值分布，模仿生物体在决策时考虑风险的方式 [6]。许多环境固有的随机性促使人们使用概率分布来评估某个动作所对应的未来可能奖励 [7]。通过估计这些分布，分布式强化学习简化了强化学习中的风险厌恶参数化，因此在不确定性下必须满足约束的问题中具有优势。

II. 强化学习（RL）背景

强化学习通常将智能体-环境交互建模为马尔可夫决策过程（MDP）。MDP 由元组 $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$ 定义，其中 $\mathcal{S}$ 是环境中的状态集合，$\mathcal{A}$ 是可能动作的集合，$\mathcal{R}: \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ 是奖励函数，$\mathcal{P}: \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ 是转移概率函数（$P(s' \mid s, a)$ 表示从状态 s采取动作 a 转移到状态 s' 的概率），而 γ 是未来奖励的折扣率。策略 $\pi: \mathcal{S}\to\mathcal{P}(A)$ 将状态映射为可能动作上的概率分布（$\pi(a \mid s)$ 表示从状态 s采取动作 a 的概率）。我们用 Π表示所有可能平稳策略的集合。强化学习旨在寻找一个策略 π，使性能指标 $J_R(\pi)$ 最大化；该指标通常取为 $J_R(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_t^\infty \gamma^t R(s_t, a_t, s_{t+1})]$,，其中 $\tau = (s_0, a_0, s_1, ...)$ 是使用 π采样得到的一条轨迹。将轨迹的折扣回报记为 $\hat{}R(\tau)$，则同策略状态价值函数记为 $V^\pi(s) = \mathbb{E}_{\tau\sim\pi}\left[\hat{}R(\tau) \mid s_0 = s\right]$，其最优策略为 $\pi^* = \text{argmax}_\pi V^\pi(s)$。类似地，状态-动作价值函数定义为 $Q^\pi(s\ a) = \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid s_0 = s,\ a_0 = a\right]$，动作的优势函数为 $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$。

传统强化学习算法受到维数灾难 [8]的困扰。这种现象描述了在处理

*c.tsay@imperial.ac.uk

高维状态空间时难以获得可靠结果的挑战，其原因是，为了得到结果所需的数据量会随着维度的增加而呈指数级增长。面对这一挑战，深度强化学习采用非线性函数逼近器，即带有参数 θ的深度神经网络（NN），来估计最优状态-动作价值函数 $Q(s,a;\theta) \approx Q^*(s,a)$，以应对非常复杂且规模庞大的状态空间和动作空间。或者，我们也可以直接学习一个用于策略的参数化函数 $\pi_\theta(a \mid s)$，从而能够优化具有连续动作空间的问题。在策略梯度方法中，我们直接学习一个用于策略的参数化函数，表示为$\pi_\theta(a|s)$，其参数为 θ。为此，我们训练一个神经网络，使目标函数 $J(\theta)$最大化；该目标函数衡量遵循策略 π_θ所获得的期望回报。策略梯度定理 [9]给出了目标函数对策略参数 θ的导数表达式：

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi \left[Q^\pi(s,a) \nabla_\theta \ln \pi_\theta(a|s) \right] \tag{1}$$

这会产生一种更新规则，虽然无偏，但梯度估计的方差较大。通常通过用优势函数代替 $Q^\pi(s,\ a)$ 来缓解这一问题；使用 $V^\pi(s)$ 作为基线进行广义优势估计（GAE）[10] 。

信赖域策略优化。近期一种处理噪声梯度估计的策略是信赖域策略优化（TRPO） [11], ，它将当前策略 π 与先前策略 π_k 之间的 Kullback-Leibler（KL）散度限制在阈值 δ 以内。因此，训练策略参数 θ 包括：

$$\begin{aligned} \max_\theta \quad & \mathbb{E}_{s\sim\rho^{\pi_k}, a\sim\pi_k} \left[\frac{\pi(a|s)}{\pi_k(a|s)} A^{\pi_k}(s,a) \right] \\ \text{s.t.} \quad & \mathbb{E}_{s\sim\rho^{\pi_k}} \left[\overline{D}_{\mathrm{KL}}\left(\pi_k(\cdot|s) \,||\, \pi(\cdot|s)\right) \right] \le \delta \end{aligned} \tag{2}$$

其中，ρ^{π_k} 是在 π_k 下的状态访问概率密度，D_{KL} 表示平均 KL散度。问题（2）通常通过对 $Q^{\pi_k}(s,\ a)$ 进行蒙特卡罗估计、对目标函数进行线性化，并对信赖域约束进行二阶近似来重新表述。随后，通常使用共轭梯度和回溯线搜索求解重新表述后的问题。

约束强化学习。一种自然的扩展是在环境受约束的场景中研究强化学习。约束强化学习算法通常假设一个约束马尔可夫决策过程（CMDP），其中对成本 C：$\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$的建模方式与奖励类似，并设定一个阈值 d。目标是在满足成本约束$J_C(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_t^\infty \gamma^t C(s_t,\ a_t,\ s_{t+1})] \le d$ 的条件下，最大化 $J_R(\pi)$。这可以用最优策略 $\pi^* = \operatorname{argmax}_{\pi\in\Pi_C} J_R(\pi)$来表示，其中 $\Pi_C \doteq \{\pi \in \Pi :\ J_C(\pi) \le d\}$ 是成本可行策略的集合。

对于基于深度 Q 网络的方法而言，纳入这种约束设定并非易事；一种常见策略是通过约束违反来惩罚奖励函数 [12]。另一方面，在策略更新策略中纳入环境约束则更为直接，因为可以将约束纳入现有的优化问题，例如问题（2）。

约束策略优化 (CPO) [13]与 TRPO 类似，会限制策略更新，但其目的是同时满足问题（2）中的信赖域约束和环境约束。所提出的优化问题可以利用对偶性高效求解，并在每一步估计目标函数和环境约束对应的拉格朗日乘子。另一种实施约束的方式是扩展状态空间并重塑问题的目标函数。安全增强型（*SAUTE*）马尔可夫决策过程 [14] 可将问题重新表述为最小化重塑后的目标函数，同时通过安全预算纳入安全约束。预算耗尽后，目标函数将取无穷大值，从而确保训练结束时约束满足的概率为 1。这种方法仅修改环境，并兼容大多数现成的强化学习算法；但状态空间会随环境约束数量增加而扩大，进而导致可扩展性问题；若约束数量较多，采样效率也会因此降低。

III. 方法：分布式约束策略优化

在这项工作中，我们表明，将分布式强化学习应用于 CPO 场景，能够在可靠的约束满足与稳定的采样效率之间取得平衡，且不受约束数量的影响。分布式强化学习 [6] 将传统强化学习扩展到考量所有状态-动作对上的回报概率分布。该方法提供了一系列辅助统计量，例如回报方差；这些统计量共同构成了一个更自然的归纳偏置框架。类似地，我们提出近似状态-动作对上的成本分布，以便在训练期间管理风险厌恶。我们将这个 CPO 的分布式版本称为 DCPO，其伪代码见算法1。

问题表述。 考虑带有单个环境约束的 CPO 策略更新：

$$\begin{aligned} \max_{\pi\in\Pi_\theta} \quad & \mathbb{E}_{s\sim\rho^{\pi_k}, a\sim\pi_k} \left[A^{\pi_k}(s,a) \right] \\ \text{s.t} \quad & J_C(\pi_k) + \frac{1}{1-\gamma} \mathbb{E}_{s\sim\rho^{\pi_k}, a\sim\pi_k} \left[A_C^{\pi_k}(s,a) \right] \le d \\ & \overline{D}_{\mathrm{KL}}\left(\pi_k \,||\, \pi\right) \le \delta \end{aligned} \tag{3}$$

其中，$A_C^{\pi_k}$表示在 π_k 下针对成本函数 C估计的成本优势。对于高维神经网络参数空间而言，该更新规则的计算成本较高。当δ较小时，我们可以围绕π_k对目标函数和安全约束进行线性化。分别将目标函数和约束的梯度表示为g 和 b，将KL散度的海森矩阵表示为 H（即费舍尔信息矩阵），并定义 $c = J_C\ (\pi_k) - d$，则式（3）可以用步进方向 $x = \theta - \theta_k$ 表示为：

$$\begin{aligned} p^* = \quad & \min_x \ g^T x \\ \text{s.t} \quad & c + b^T x \le 0,\ x^T H x \le \delta \end{aligned} \tag{4}$$

其中包括 $g,\ b,\ x \in \mathbb{R}^n$、$c,\ d \in \mathbb{R}$、$\delta > 0$和 $H \succ 0$。我们注意到，g 和 b 可以通过反向传播计算得出，并且 H 始终是半正定的，因此（4）是一个凸

优化问题，在可行时可通过 KKT驻点条件求解。给定单个成本函数，最优点 x^* （假设其存在）为

$$x^* = -\frac{1}{\lambda^*} H^{-1}\left(g - \nu^* b\right) \tag{5}$$

对偶变量 λ^* 和 ν^* （拉格朗日乘子）为:

$$\begin{aligned} \nu^* &= \left(\frac{\lambda^* c - r}{s}\right)_+, \\ \lambda^* &= \underset{\lambda \geq 0}{\operatorname{argmax}} \begin{cases} f_a(\lambda) & \text{if } \lambda c - r > 0 \\ f_b(\lambda) & \text{otherwise,} \end{cases} \end{aligned} \tag{6}$$

其中 $q = g^T H^{-1} g$、$r = g^T H^{-1} b$、$s = b^T H^{-1} b$ 和 $f_a(\lambda) \doteq -\frac{1}{2\lambda}\left(q + \lambda\delta\right) + \frac{1}{2\lambda}\left(r^2/s - q\right) + \frac{\lambda}{2}\left(c^2/s - \delta\right) - rc/s$ 和 $f_b(\lambda) \doteq -\frac{1}{2}\left(\delta - c^2/s\right)\lambda - 2c/s$。实际上，计算 H^{-1} 的开销很大，因此我们使用共轭梯度近似求解 $H^{-1}g$ 和 $H^{-1}b$。由于 $\lambda \geq 0$，我们必须通过投影，分别将上述两种情况下的最优 λ 限制在集合 $\Lambda_a \doteq \{\lambda \mid \lambda c - r > 0, \lambda \geq 0\}$ 和 $\Lambda_b \doteq \{\lambda \mid \lambda c - r \leq 0, \lambda \geq 0\}$ 中，使得

$$\lambda^* \in \left\{ \text{Proj}\left(\sqrt{\frac{q - r^2/s}{\delta - c^2 s}}, \Lambda_a\right), \text{Proj}\left(\sqrt{\frac{q}{\delta}}, \Lambda_b\right) \right\} \tag{7}$$

$$\lambda^* = \begin{cases} \lambda_a^* & \text{if } f_a(\lambda_a^*) \geq f_b(\lambda_b^*) \\ \lambda_b^* & \text{otherwise.} \end{cases} \tag{8}$$

值得一提的是，H 是通过二次约束计算得出的，因此采取的任何步骤都满足 KL 散度约束。在不可行的情况下，即当 $c^2 s - \delta > 0$和 $c > 0$时，我们采用一种旨在满足约束的恢复方法，$x^* = -\sqrt{2\delta / (b^T H^{-1} b)} H^{-1} b$。

分布式价值函数与安全基线。准确估计式 (3) 中的回报优势函数和安全优势 A^{π_k}和 $A_C^{\pi_k}$，对于降低策略梯度更新步骤带来的方差至关重要。这些优势函数可以通过广义优势估计（GAE） [10], 等方法进行估计，并使用神经网络近似。为了对不确定性建模，我们选择使用神经网络来表示回报价值函数和成本价值函数（用于估计 A^{π_k} 和 $A_C^{\pi_k}$），其输出为分布，并由范围 $[V_{\min}, V_{\min}]$ [6]内等间距的 N 个分位数参数化。我们为这些神经网络采用代理损失，该损失由在策略 π_k下采样得到的成本或回报的平均负对数似然构成。为简化对数似然的计算，我们将分位数分布近似为高斯分布。因此，对于参数为 ϕ 的价值函数，其损失为 $\mathcal{L}(\phi) = -\log\left(p(J_R(\pi_k) \mid \phi)\right)$，可通过随机梯度下降进行训练。

成本重塑。为了可靠地满足约束，我们根据智能体对于约束已满足的置信度，重新塑造成本 $J_C(\pi_k)$ 和安全阈值 d；这种置信度基于参数为 ϕ_c 的成本分布。具体而言，我们引入重塑参数 ρ: 当 $J_C^{\pi_k} > d$ 时（即约束被违反），其值设为 $\beta \mathbb{P}(J_C^{\pi_k} > d; \phi_c)$；当约束得到满足时，其值设为$-\beta \mathbb{P}(J_C^{\pi_k} < d; \phi_c)$，其中 $\beta \geq 0$。请注意，我们根据先前的策略π_k评估 J_C。换言之，当智能体完全确信在先前的策略 π_k 下约束被违反时，ρ 取值为 β；当智能体完全确信约束得到满足时，取值为 $-\beta$。接着，我们将 $J_C^{\pi_k}$ 和 $d = J_C^{\pi_k}$ 重塑为 $J_C^{\pi_k}(1 + \text{clip}(\rho, -\epsilon, \infty))$ 和 $d = d(1 + \text{clip}(\rho, -\epsilon, \infty))$。新参数 ϵ 是用于控制风险厌恶的安全裕度。只有在约束得到满足时，这种重塑才会影响策略更新，从而有效避免沿着可能增加成本的方向迈出过大的步子。直观而言，在受约束环境中最大化奖励通常会使智能体走到可行域的边界。因此，我们根据约束满足的置信度调整信赖域的可行性，以此降低风险，同时保持与 CPO 相同的优化问题（见命题1）。

命题1. 对于任意函数 $f: S \to \mathbb{R}^+$, $p: S \to \mathbb{R} \in [-1, 1]$, 以及 $d, \beta \in \mathbb{R}^+$, $\epsilon \in (0,1)$，定义

$$k(s) = \begin{cases} -\epsilon & \textit{if } \beta p(s) < -\epsilon, \\ \beta p(s) & \textit{otherwise}, \end{cases} \tag{9}$$

如果将 $f(s)$ 和 d 分别乘以标量 $1 + k(s)$，则有:

$$\tilde{f}(s) = f(s)\left(1 + k(s)\right) \tag{10}$$

$$\tilde{d}(s) = d\left(1 + k(s)\right) \tag{11}$$

那么，

$$f(s) - d > 0 \implies \tilde{f}(s) - \tilde{d}(s) > 0 \tag{12}$$

证明。 $1 + k(s) > 0$ 给定 $\epsilon \in (0,1)$ 和 $\beta \in \mathbb{R}^+$。因此，这立即成立。 □

IV. 结果与讨论

A. 供应链案例研究

我们研究多级、多周期供应链 [15]中的库存管理问题。供应链由生产不同阶段的一系列互联节点组成，旨在将原材料产品转化为可在供应链流程最终阶段供消费的产品。运营目标是在满足需求和库存约束的同时最大化利润。我们研究 or-gym库 [16]中的 *InvManagement-v0*问题。给定周期内的事件顺序可概括如下:

1) 主网络节点向其供应商（位于相应的上游阶段）下达补货订单。订单根据供应商可用的生产能力和库存进行履行。2) 主网络节点接收库存补货货物；在相应提前期（包括生产和运输时间）结束后，货物沿供应链流程运抵。3) 零售节点根据可用库存满足单一产品的客户需求。4) 未满足的销售需求会产生积压订单，并需支付相应成本；这些订单随后会优先在下一个周期处理，但销售利润较低。5) 各阶段的剩余库存需支付持有成本。6) 最后一个周期结束后，剩余库存将被损失。

算法1： 分布式约束策略优化（DCPO）

输入： 初始策略 $\pi_0 \in \Pi_\theta$，安全裕度 ϵ，重塑系数 β
对于 $k = 0, 1, 2, \ldots$ **执行**
  采样一组轨迹 $\mathcal{D} = \tau \sim \pi_k = \pi(\theta_k)$
  构造样本估计 $\hat{g}$，$\hat{b}$，$\hat{H}$，使用 $\hat{c}$ $\mathcal{D}$，并进行重塑 $\tilde{J}_C(\pi_k)$，$\tilde{d}$
  如果 近似 *CPO*(4) 可行**则**
    使用重塑后的约束求解对偶问题 $\tilde{c} = \tilde{J}_C(\pi_k) - \tilde{d}$
    使用 $x^* = -\frac{1}{\lambda^*} H^{-1}(g - \nu^* b)$计算新策略 θ^*
  else
    使用 $x^* = -\sqrt{2\delta/(b^T H^{-1} b)} H^{-1} b$计算恢复策略 θ^*
  end
  通过回溯线搜索获得 θ_{k+1}，以在 $\mathcal{D}$上满足式 (3) 中的约束
end

从数学上看，对于每个周期 t 和 $\forall m \in \mathcal{M}$，$t \in \mathcal{T}$，环境的动态过程可表示为：

$$I_{t+1}^m = I_t^m + R_{t-L_m}^m - S_t^m \tag{13a}$$
$$T_{t+1}^m = T_t^m - R_{t-L_m}^m + R_t^m \tag{13b}$$
$$S_t^m = \begin{cases} R_t^{m-1} & \text{if } m > 0, \\ \min\left(I_t^0 + R_{t-L_m}^0, D_t + B_{t-1}^0\right) & \text{if } m = 0 \end{cases} \tag{13c}$$
$$U_t^m = R_t^{m-1} - S_t^m \tag{13d}$$
$$P_t^m = (\alpha)^n \left(p^m S_t^m - r^m R_t^m - k^m U_t^m - h^m I_{t+1}^m\right) \tag{13e}$$

式 13a–b 是每个阶段 m 的现有库存 (I) 和供应链流程库存 (T) 的质量平衡，其中 R 和 L 分别表示补货订单和提前期。式 13c 根据需求 D 和积压订单 B，给出零售商 ($m = 0$) 和供应链流程其余部分 ($m > 0$) 的销售额 (S)。式 13d 描述未满足的再订购量 (U)，式 13e 则将利润 P 计算为销售收入减去采购成本、未满足需求惩罚和库存过剩惩罚，并受折扣因子 $\alpha = 0.97$约束。单位销售价格 p 和惩罚 k，h 均为已知量。

再订购量 R_t^m 受生产能力（c^{m+1}）和库存（I_t^{m+1}）约束。为施加这些约束，我们不采用 [16] 中的显式施加方式，而是改为定义一个成本函数，对约束违反的次数求和 $C(s, a) = \sum_{m \in \mathcal{M},\ t \in \mathcal{T}} C\left(I_t^{m+1}, R_t^m\right)$，其中 $C\left(I_t^{m+1}, R_t^m\right)$ 是节点 m 在时间 t 违反的约束数量。我们使用这个指示成本函数，而非连续成本函数，以限制生产能力和库存约束违反的次数，而不考虑违反程度。因此，智能体的目标是在生产能力和库存约束违反次数受限的条件下，实现利润最大化。

B. DCPO 结果

所有实验均使用十个随机种子运行，并通过反复试验手动调整每种算法的超参数，以便在合理的时间内获得合理的结果。此外，所有算法均为策略、回报值分布和成本值分布分别使用大小为（64，64）的前馈神经网络，并采用 tanh 激活函数。使用 [1] $N = 102$ 个分位数运行 DCPO，展示了训练过程中所学习回报分布的收敛情况（图 1）。直观而言，在较早的周期中，智能体观察到的状态转移较少，因此对某一状态的成本/价值（认知不确定性）缺乏确定性 [17]。这一点从训练轮次 1 和 80 中所学习分布相对平坦可以明显看出。随着观察到更多样本，该分布会围绕少数几个独立分位数收敛，而状态价值中剩余的不确定性可能源于马尔可夫决策过程的随机性（偶然不确定性）。在训练过程中，我们观察到所学习成本分布也呈现出类似趋势。

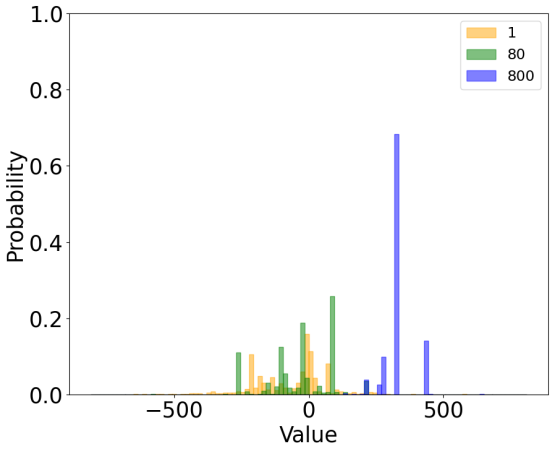

图 1：训练过程中第 1、80 和 800 轮次时，初始状态 s_0 的回报分布的学习分位数（$N = 102$）。

成本重塑的重要性。 我们首先进行一项关于成本重塑的消融实验，以考察将分布式强化学习应用于 CPO 所产生的独立影响。图 2 展示了 CPO、DCPO 和“消融”实验（未进行成本重塑的 DCPO）在训练期间产生的成本和回报。我们观察到，使用参数化回报和成本价值分布能够改善训练的初始阶段。具体而言，“消融”智能体能更快地学会同时降低成本和提高回报，同时收敛

[1] http://github.com/jaimesabalimperial/jaisalab

到与 CPO 相近的数值。此外，与 CPO 相比，这还降低了重复试验中回报的标准差；在训练结束时，降幅为 40。0%。当约束经过重塑时，这一效果甚至更加显著：DCPO 的标准差较 CPO 下降了 76。6%。

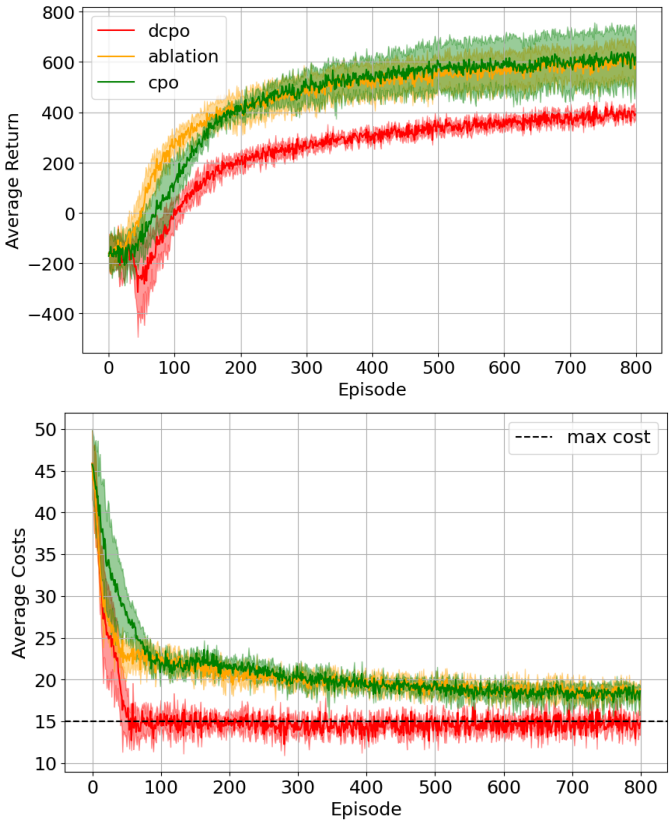

图 2：CPO、DCPO 以及未进行成本重塑（“消融”）的 DCPO 在 5 次运行中的回报（上）和成本（下）。DCPO （$\beta = 100.0$, $\epsilon = 0.15$）收敛至成本上限。

与 TRPO 和 Saute 的比较。图 3 展示了 TRPO、Saute TRPO、CPO 和 DCPO 在训练期间产生的成本和回报。TRPO 不受约束，会产生不可行策略，违规次数超过允许上限的两倍，但其回报最高。另一方面，Saute TRPO 在训练结束时的成本远低于上限，但回报较低，形成了明显的权衡。DCPO 似乎平衡了回报与成本之间固有的权衡：平均成本低于设定上限，同时回报高于 Saute TRPO。此外，到训练结束时，与 TRPO 相比，DCPO 的回报标准差降低了 70.8%，这可能归因于约束重塑机制固有的风险厌恶。具体而言，到训练结束时，约束始终得到满足；若某一步沿着违反约束的方向移动，后续步骤便会强力将其拉回信赖域中的可行区域。由于该环境中的回报与成本之间存在明显的权衡，这在很大程度上将策略更新限制在成本阈值附近的区域。最后，CPO 收敛至成本和回报的中间值，这可能是由于目标函数梯度和成本梯度的近似误差与采样误差所致。

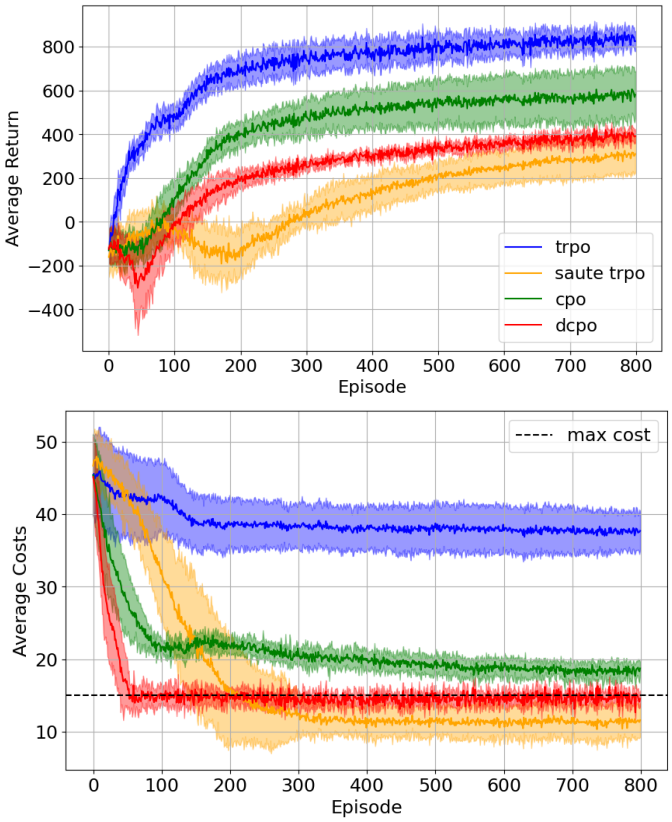

图 3：TRPO、Saute TRPO、DCPO 和 CPO 在 10 次运行中的回报（上）和成本（下）。

图 4 展示了训练后的算法经过 30 个训练轮次测试后，归一化成本（安全性）和回报（目标函数）的箱线图。推理期间的结果与训练结束时的结果类似，详见图 5。

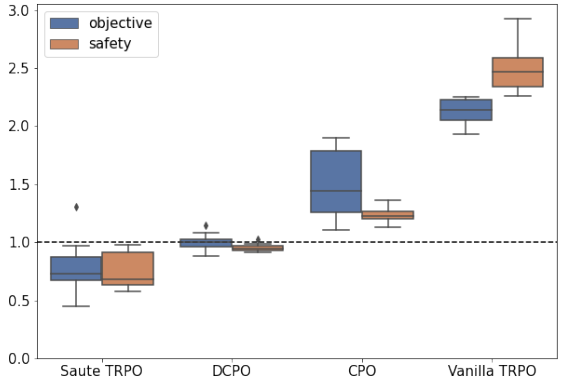

图 4：TRPO、Saute TRPO、DCPO 和 CPO 在 10 次运行中运行 30 个回合后的归一化回报和成本。回报除以 329。5（DCPO 的平均性能），成本除以 15。0（安全预算；以虚线标出）。

事实上，DCPO 的回报和成本方差均为最小；与 CPO 相比，其成本标准差降低了 52.1%，回报标准差降低了 74.8%。此外，DCPO 的平均约束满足表现可靠，处于距 d一个标准差的范围内，尽管仍有一些违反约束的离群轨迹。尽管如此，它在满足约束方面仍优于 CPO，后者完全没有满足约束的轨迹。这表明，此处的案例研究在约束方面或许比 CPO 更适用的环境更具挑战性。此外，这也说明，对于约束满足占主导地位的环境，DCPO 尤为有益。

	$J_R(\pi^*)$	$J_C(\pi^*)$
原始 TRPO	**2.123**	2.493
Saute TRPO	0.786	**0.757**
CPO	1.499	1.232
DCPO	1.000	0.953

表 I：均值。

	$J_R(\pi^*)$	$J_C(\pi^*)$
原始 TRPO	0.112	0.198
Saute TRPO	0.231	0.162
CPO	0.298	0.071
DCPO	**0.075**	**0.034**

表 I：标准差。

图 5：折扣回报 $J_R(\pi^*)$ 和成本 $J_C(\pi^*)$ 的归一化评估性能。回报通过除以 329.5（DCPO 的平均性能）进行归一化，成本则除以安全预算 15.0。

V. 结论

本文提出了一种用于 CPO 信赖域优化的方法，通过利用智能体对约束满足的置信度来调整回合折扣成本和最大允许值，而这种置信度由近似的成本分布给出。所提出的算法称为 DCPO，在计算供应链案例研究中能够平衡最大化回报与最小化成本之间的权衡，使约束在十次重复试验中均得到满足（在一个标准差范围内）。不过，DCPO 仍会产生一些违反约束的轨迹，但其程度远低于 CPO。此外，与其他方法相比，该算法产生的成本和回报也更加可预测：训练结束时，成本方差较 CPO 惊人地降低了 52.1%，奖励方差则降低了 74.8%（见图 5）。未来的工作可以探索如何利用学到的成本/回报分布，在训练期间直接施加约束，而不是调整安全约束的拉格朗日乘子。另一个值得探索的方向是采用不同的方法对成本/价值分布进行参数化，例如使用 QR-DQN [18] 方法学习分位数的位置，而非将其固定。

VI. 致谢

作者感谢工程与物理科学研究委员会（EPSRC）通过研究员奖学金 EP/T001577/1 提供的支持，以及帝国理工学院向 CT 提供的研究员奖学金。

参考文献

[1] R. Nian、J. Liu 和 B. Huang，《强化学习综述：工业过程控制中的介绍与应用》，《计算机与化学工程》，第 139 卷，第 106886 页，2020 年。[2] J. Shin、T. A. Badgwell、K.-H. Liu 和 J. H. Lee，《强化学习——近期进展综述及其对过程控制的启示》，《计算机与化学工程》，第 127 卷，第 282–294 页，2019 年。[3] T. J. Ikonen、K. Heljanko 和 I. Harjunkoski，《自适应在线重调度时机与计算时间分配的强化学习》，《计算机与化学工程》，第 141 卷，第 106994 页，2020 年。[4] C. Shyalika、T. Silva 和 A. Karunananda，《动态任务调度中的强化学习：综述》，《*SN Computer Science*》，第 1 卷，第 1–17 页，2020 年。[5] J. Garcıa 和 F. Fern´andez，《安全强化学习综合综述》，《机器学习研究杂志》，第 16 卷，第 1 期，第 1437–1480 页，2015 年。[6] M. G. Bellemare、W. Dabney 和 R. Munos，《强化学习的分布式视角》，载于国际机器学习会议。PMLR，2017 年，第 449–458 页。[7] P. Petsagkourakis、I. O. Sandoval、E. Bradford、F. Galvanin、D. Zhang 和 E. A. del Rio-Chanona，《过程控制与优化的机会约束策略优化》，《过程控制杂志》，第 111 卷，第 35–45 页，2022 年。[8] R. Bellman，《动态规划》。多佛出版社，1957 年。[9] R. S. Sutton、D. McAllester、S. Singh 和 Y. Mansour，《结合函数逼近的强化学习策略梯度方法》，载于第十二届国际神经信息处理系统会议论文集，系列 NIPS'99。美国马萨诸塞州剑桥：麻省理工学院出版社，1999 年，第 1057–1063 页。[10] J. Schulman、P. Moritz、S. Levine、M. Jordan 和 P. Abbeel，《使用广义优势函数估计进行高维连续控制》，*arXiv:1506.02438*，2015 年。[11] J. Schulman、S. Levine、P. Abbeel、M. Jordan 和 P. Moritz，《信赖域策略优化》，载于国际机器学习会议。PMLR，2015 年，第 1889–1897 页。[12] H. Yoo、V. M. Zavala 和 J. H. Lee，《深度强化学习中状态约束处理的动态惩罚方法》，《过程控制杂志》，第 115 卷，第 157–166 页，2022 年。[13] J. Achiam、D. Held、A. Tamar 和 P. Abbeel，《约束策略优化》，载于国际机器学习会议。PMLR，2017 年，第 22–31 页。[14] A. Sootla、A. I. Cowen-Rivers、T. Jafferjee、Z. Wang、D. H. Mguni、J. Wang 和 H. Ammar，《Saut´e RL：通过状态增广实现几乎必然安全的强化学习》，载于国际机器学习会议。PMLR，2022 年，第 20 423–20 443 页。[15] H. D. Perez、C. D. Hubbs、C. Li 和 I. E. Grossmann，《库存管理优化的算法方法》，《*Processes*》，第 9 卷，第 1 期，2021 年。[16] C. D. Hubbs、H. D. Perez、O. Sarwar、N. V. Sahinidis、I. E. Grossmann 和 J. M. Wassick，《Or-gym：面向运筹学问题的强化学习库》，*arXiv:2008.06319*，2020 年。[17] W. R. Clements、B. Van Delft、B.-M. Robaglia、R. B. Slaoui 和 S. Toth，《深度强化学习中的风险与不确定性估计》，*arXiv:1905.09638*，2019 年。[18] W. Dabney、M. Rowland、M. G. Bellemare 和 R. Munos，《结合分位数回归的分布式强化学习》，*arXiv:1710.10044*，2017 年。