可扩展的极小极大优化通过原对偶精确帕累托优化

Sangwoo Park
Imperial College London
London, United Kingdom
s.park@imperial.ac.uk

Stefan Vlaski
Imperial College London
London, United Kingdom
s.vlaski@imperial.ac.uk

Lajos Hanzo
University of Southampton
Southampton, United Kingdom
hanzo@soton.ac.uk

摘要—在多目标优化中，最小化最差的目标可能比最小化平均目标更可取，因为这确保了目标之间的改进的公平性。由于结果的 min-max 优化问题蚂蚁非光滑性质，经典次梯度-基础方法通常表现出收敛速度慢。受多代理优化和学习中的原始对偶共识技术启发，我们基于增广拉格朗日函数制定了 min-max 问题的光滑变体。结果蚂蚁通过增强拉格朗日 (EPO-AL) 算法进行的精确帕累托优化，在目标数量方面比基于次梯度-的策略扩展得更好，同时展现出比最近的基于平滑-的对应物算法更低的每迭代复杂度。我们证明在温和假设下，所提出算法的每一个不动点既是帕累托最优又是最小-最大最优，并通过数值模拟证明了其有效性。

***Index Terms*—多目标优化，极小化极大优化，精确帕累托最优性，原对偶共识，增广拉格朗日。**

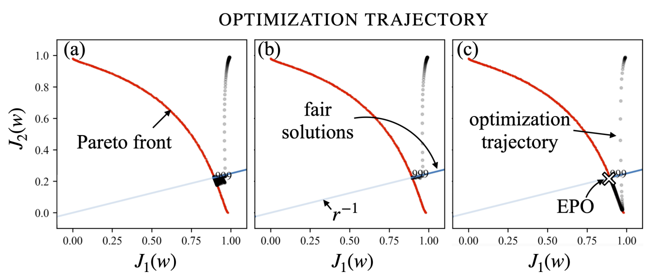
I. 介绍

我们考虑一个具有具有 K 可微、正的目标函数的多目标优化问题 $J_1(w),\dots,J_K(w)>0$，其中 $J_k(w)$ 是在模型 $w\in\mathbb{R}^d$ 处评估的第 k 个目标函数。我们希望解决加权最小最大问题 [1], [2]：

$$\min_{w\in\mathbb{R}^d}\max_{k\in[K]} r_k J_k(w), \tag{1}$$

给定一个预设的偏好向量 $r=[r_1,\dots,r_K]^\top$ 相关的 带有 $r_k>0$ 用于 $k=1,\dots,K$。我们关注我们的注意力在梯度信息 $\{\nabla J_k(w)\}_{k=1}^K$ 可用的设置。求解极小-极大问题 (1) 可能比经典线性标量化（即 $\min_{w\in\mathbb{R}^d}\sum_{k=1}^K r_k J_k(w)$）更优，特别是在公平性重要的 [3]–[7], 应用场景中，因为这确保了不会为了追求提高的平均性能而忽视个体目标 $J_k(\cdot)$。

This work was supported by EPSRC Grants EP/X04047X/1 and EP/Y037243/1.



PER-ITERATION COMPUTATIONAL COMPLEXITY

$O(\min\{K,d\})$	$O(K^2d)$	$O(Kd)$
(a) subgradient algorithm (2)	(b) EPO Search [8]	(c) EPO-AL (Algorithm 1)

图 1. 多目标优化轨迹（顶部）对于 subgradient 算法（2）在 (a)、EPO Search [8] 在 (b) 中 和通过增广拉格朗日提出的的方法在 (c) 中，称为 EPO-AL；表格显示了每迭代计算复杂度（底部）。优化轨迹对于 $K=2$ 非凸目标函数 $J_1(w)=1-e^{-\|w-1/\sqrt{d}\|^2}$ 和 $J_2(w)=1-e^{-\|w+1/\sqrt{d}\|^2}$ [9] 在 $w\in\mathbb{R}^3$ 条件下获得（详情见第 V-A 节）。帕累托前沿（红色弧线，见定义 1）与公平解（蓝色直线，见定义 2）的交点是一个精确帕累托最优 (EPO) [8] 解（白色十字，见定义 3），它在温和假设下满足最小-最大最优性（1）（见命题 1）。观察到所提出的策略首先找到帕累托前沿，然后根据（1）搜索最小最大最优的帕累托解。次梯度算法（2）由于最大算子在（1）中的非光滑特性，在最小化最大化最优解的帕累托前沿附近表现出震荡，而基于 EPO 的方法则平滑地收敛到最小化最大化最优解。

可能概念上最简单的解法来求解 (1) 是考虑次梯度算法 [10, Theorem 18.5]：

$$w_{i+1}=w_i-\mu\cdot r_{k^{\text{active}}}\nabla J_{k^{\text{active}}}(w_i) \tag{2}$$

其中 μ 是一个步长，而 $k^{\text{active}}\triangleq\arg\max_k r_k J_k(w)$ 表示在 (1) 中达到最大值的活跃目标的索引，即 $\max_k r_k J_k(w)=r_{k^{\text{active}}}J_{k^{\text{active}}}(w)$。请注意，如果对于特定模型 w，有多个目标达到最大值，即集合 $\mathcal{A}(w)=\{k'\in[K]:J_{k'}(w)=\max_k J_k(w)\}$ 包含多于一个元素，则 $\{\nabla J_k(w)\}_{k\in\mathcal{A}(w)}$ 的任何凸组合都是 min-max 目标

(1) 的次梯度，并可以在 (2) [11] 中使用。

然而，迭代更新规则（2）通常遭受收敛速度慢 [11] 的问题，主要是由于来自非活跃目标的忽略掉该部分由于内容不完整，无法提供准确翻译。请提供完整句子或段落以便进行翻译。根据您的要求，我仅能对可识别的部分进行翻译，“ignoring the” 可以翻译为 “忽略掉”。如果您有更完整的句子或更多内容需要翻译，请告知。梯度信息获取。这一观察结果推动了基于平滑-的替代方法的发展 [4], [11]–[13]。现有 (1) 的方法要么依赖于（i）直接用平滑近似替换最大值函数 [5], [12]–[15]，要么是 (二) 设计一个一个平滑鞍点问题 [4], [7], [11]，即，

$$w^\star \in \arg\min_{w\in\mathbb{R}^d}\max_{y\in\Delta^K}\sum_{k=1}^{K} r_k J_k(w) y_k, \tag{3}$$

其中 y_k 是给定的 $\Delta^K = \{y \in \mathbb{R}_+^K : y^\top \mathbb{1}_K = 1\}$ 中 $(K-1)$-单纯形 $y \in \Delta^K$ 的第 k 个元素，$\mathbb{1}_K$ 是全一矢量 $K \times 1$。

II. 预备知识

在本文中，我们考虑了一种受多目标优化经典结果启发的交替平滑方法来处理极小极大优化问题（1），这些结果捕捉了极小极大与帕累托优化之间的关系 [16]。为了正式讨论，我们引入以下定义。

定义 1 (弱帕累托最优 [1])**.** *模型 w 在不存在任何 $w' \neq w$ 使得所有目标函数都减少的情况下是弱帕累托最优,。相应地，所有弱帕累托最优点的集合 $\mathcal{P}$ 给出为*

$$\mathcal{P} = \left\{w \in \mathbb{R}^d : \nexists w' \in \mathbb{R}^d \ s.t.\ J_k(w') < J_k(w) \ for\ all\ k\right\}. \tag{4}$$

（弱）帕累托最优点的集合如图 1 所示以红色弧线表示。其次，我们引入点集公平 [7], [17]。

定义 2 (公平 [7], [17])**.** *模型 w 被称为相对于偏好向量 r 的公平，如果加权目标相等，即 $r_1 J_1(w) = \cdots = r_K J_K(w)$。所有这样的公平模型的集合用以下集合表示：*

$$\mathcal{F}_r = \left\{w \in \mathbb{R}^d : r_1 J_1(w) = \cdots = r_K J_K(w)\right\}. \tag{5}$$

集合公平的点在图 1 中用蓝色线条表示。当（弱）帕累托最优解集与集合公平解法相交时，这会产生集合精确帕累托最优解 (白色十字在图 1 中)。

定义 3 (精确帕累托最优性 [8])**.** *精确帕累托最优解的集合由以下给出：*

$$\mathcal{E}_r = \mathcal{P} \cap \mathcal{F}_r. \tag{6}$$

集合 $\mathcal{E}_r$ 由图 1 中红色弧线（弱 Pareto 最优解）和蓝色直线（公平的解决方案）的交集给出。当这个交集非空时，我们称 r 为存在 *帕累托可行*。我们注意到，与 [8, Eq. (8)] 相比，我们将 $\mathcal{E}_r$ 定义为全局（弱）帕累托最优，而不是仅仅局部帕累托最优。这使我们能够建立以下命题。

命题 1 (精确的帕累托最优性意味着最小-最大最优性)**.** *假设 rằng $w^{\mathrm{EPO}} \in \mathcal{E}_r = \mathcal{P} \cap \mathcal{F}_r$ 是弱帕累托最优和公平。然后 w^{WPO} 也如在 (1) 中定义的那样是最小最大最优。*

$$w^{\mathrm{EPO}} = \arg\min_{w\in\mathbb{R}^d}\max_{k\in[K]} r_k J_k(w). \tag{7}$$

证明. 我们通过反证法证明该命题。假设 $w^{\mathrm{EPO}} \in \mathcal{P} \cap \mathcal{F}_r$ 不是 (1) 的最小值。那么：

$$\exists w : \max_{k\in[K]} r_k J_k(w) < \max_{k\in[K]} r_k J_k(w^{\mathrm{EPO}}). \tag{8}$$

我们首先使用公平条件 (5) 简化右侧。从 $w^{\mathrm{EPO}} \in \mathcal{F}_r$，我们有

$$\max_{k\in[K]} r_k J_k(w) < r_\ell J_\ell(w^{\mathrm{EPO}}) \text{ for any } \ell \in [K]. \tag{9}$$

因为集合中的每个元素都以上界为其最大值：

$$r_\ell J_\ell(w) \leq \max_{k\in[K]} r_k J_k(w) < r_\ell J_\ell(w^{\mathrm{EPO}}) \text{ for all } \ell \in [K]. \tag{10}$$

在等式 (10) 的两边消去 $r_\ell > 0$ 后，我们得出 w^{EPO} 不能是弱帕累托最优。因此 $w^{\mathrm{EPO}} \notin \mathcal{P} \cap \mathcal{F}_r$，导致矛盾。 □

命题 1 提供了确保可以通过寻找位于（弱）Pareto 前沿 $\mathcal{P}$ 上的一个点来求解最小化最大化问题 (1) 的充分条件，该点同时也是 公平 $\mathcal{F}_r$ 。这一事实并不总是成立的—参见 [16] 中的反例 - 。只要 $\mathcal{E}_r = \mathcal{P} \cap \mathcal{F}_r$ 非空，那么任何精确的帕累托最优点也是最小-最大最优。受此考虑启发，在接下来的内容中我们将提出一个新的通过精确帕累托优化进行最小-最大优化的算法。与文献中存在的现有算法相比，所提出的策略将依赖于单一时间尺度，并且每迭代复杂度降低至 $O(Kd)$，而非 $O(K^2 d)$ [8], [18], [19] 。这导致了目标数量 K 的更好扩展性。

III. 精确帕累托最优性通过增广拉格朗日方法实现

在本节中，我们开发了一种通过增广拉格朗日实现精确帕累托优化的算法，该算法灵感来自于多智能体优化和学习中的经典原始对偶共识技术—早期示例见 [20], [21]，近期综述见 [22]。为此，注意可以通过以下方式追求精确的帕累托最优解：

$$\min_{w\in\mathbb{R}^d} \frac{1}{K}\sum_{k=1}^{K} J_k(w) \tag{11a}$$

$$\text{s.t. } r_k J_k(w) = r_\ell J_\ell(w) \quad \forall k,\ell. \tag{11b}$$

这里，的关系船舶 (11a) 鼓励帕累托最优性，而 (11b) 确保公平性条件 (5)。鉴于 (11b)，目标 (11a) 可以被 $J_k(\cdot)$ 具有 非负权重的任何线性组合所取代。由 $\frac{1}{K}$ 给出的等权重选择仅仅是出于简单考虑。类似于 [20]–[22]，我们将约束集合 (11b) 替换为一个涉及聚合约束违规的单一约束：

$$\min_{w\in\mathbb{R}^d} \frac{1}{K}\sum_{k=1}^{K} J_k(w) \tag{12a}$$

$$\text{s.t. } \frac{1}{2K}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \|r_k J_k(w) - r_\ell J_\ell(w)\|^2 = 0. \tag{12b}$$

公平性条件 (12b) 可以更紧凑地写成：

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \|r_k J_k(w) - r_\ell J_\ell(w)\|^2 = \mathcal{J}(w)^\top L_r \mathcal{J}(w) = 0, \tag{13}$$

其中 $\mathcal{J}(w) = [J_1(w),\dots,J_K(w)]^\top$ 是一个包含评估对于模型 w 的目标集合的向量，而 L_r 由以下给出：

$$L_r = \operatorname{diag}(r)\Big(I_{K\times K} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\top\Big)\operatorname{diag}(r). \tag{14}$$

这里，$\operatorname{diag}(r)$ 是对角矩阵，其对角线上包含 r 的元素。由于 L_r 是对称且半正定的，它有一个平方根 $\sqrt{L_r}$ 满足 $\sqrt{L_r}\sqrt{L_r} = L_r$，因此 (12b) 等价于：

$$\Big\|\sqrt{L_r}\mathcal{J}(w)\Big\|^2 = 0 \iff \sqrt{L_r}\mathcal{J}(w) = 0. \tag{15}$$

然后我们可以定义相应的增广拉格朗日函数 [23, Sec. 4] 作为

$$\mathcal{L}(w,\lambda) = \frac{1}{K}\mathbb{1}_K^\top\mathcal{J}(w) + \lambda^\top\sqrt{L_r}\mathcal{J}(w) + \frac{\eta}{2}\big\|\sqrt{L_r}\mathcal{J}(w)\big\|^2, \tag{16}$$

其中 $\eta > 0$ 是一个惩罚参数，而 λ 是对应的拉格朗日乘子。然后我们以迭代的一阶方法更新变量原始问题的 w 和变量对偶的 λ，如 [24], [25] 所示：

$$\begin{aligned} w_i &= w_{i-1} - \mu\nabla_w\mathcal{L}(w_{i-1},\lambda_{i-1}) && \text{(17a)}\\ &= w_{i-1} - \mu G(w_{i-1})\left[\frac{1}{K}\mathbb{1}_K + \sqrt{L_r}\lambda_{i-1} + \eta L_r\mathcal{J}(w_{i-1})\right] \\ \lambda_i &= \lambda_{i-1} + \mu\nabla_\lambda\mathcal{L}(w_{i-1},\lambda_{i-1}) = \lambda_{i-1} + \mu\sqrt{L_r}\mathcal{J}(w_{i-1}), && \text{(17b)}\end{aligned}$$

其中 $G(w) = \big[\nabla J_1(w),\dots,\nabla J_K(w)\big]$ 是一个收集来自评估为了模型 w 的 K 目标的梯度的 $d\times K$ 矩阵，而 $\mu > 0$ 是步长。将 (17b) 从左侧乘以 $\sqrt{L_r}$ 并定义 $p_i \triangleq (1/K)\mathbb{1}_K + \sqrt{L_r}\lambda_i$，我们得到以下等效形式：

$$w_i = w_{i-1} - \mu G(w_{i-1})\big[p_{i-1} + \eta L_r\mathcal{J}(w_{i-1})\big] \tag{18a}$$

$$p_i = p_{i-1} + \mu L_r\mathcal{J}(w_{i-1}). \tag{18b}$$

从初始化 $\lambda_0 = 0$，我们找到初始条件 $p_0 = (1/K)\mathbb{1}_K$。最后，我们对 (18a) 中的 p_{i-1} 逐元素应用 positivity 算子 $[\cdot]_+ = \max\{\cdot,0\}$，从而得到算法 1。

Algorithm 1: 精确帕累托优化通过增广拉格朗日（EPO-AL）

Input: K positive, differentiable, objectives $J_1(\cdot),\dots,J_K(\cdot)$ with $J_k:\mathbb{R}^d\to\mathbb{R}_+$; step size $\mu > 0$; penalty parameter $\eta > 0$.

初始化 $w_0\in\mathbb{R}^d, p_0 = \mathbb{1}_K/K, z_t = \mathbf{0}_K$ 其中 $\mathbf{0}_K$ 是大小为 K 的全零向量。**for** $i = 1,2,\dots$ **do**

$$w_i = w_{i-1} - \mu G(w_{i-1})\big([p_{i-1}]_+ + \eta L_r\mathcal{J}(w_{i-1})\big) \tag{19a}$$

$$p_i = p_{i-1} + \mu L_r\mathcal{J}(w_{i-1}) \tag{19b}$$

end

EPO-AL 的每次迭代复杂度与目标数量的良好扩展性如以下备注所述。

备注 1 (每迭代计算复杂度)**.** *EPO-AL 的每次迭代需要进行 $O(Kd)$ 次计算，这来自于评估梯度矩阵 $d\times KG(w_{i-1})$ 并与向量 $K\times 1([p_{i-1}]_+ + \eta L_r\mathcal{J}(w_{i-1}))$ 相乘。*

请注意，典型的多目标优化算法 [26]，包括那些旨在寻找精确帕累托最优解对于的方法 [8], [18], [19] 通常

涉及评估 $G(w_{i-1})^\top G(w_{i-1})$, 这需要大约 $O(K^2d)$ 次计算；基于次梯度的方法（2）需要 $O(K)$ 次计算来识别主动目标，以及 $O(d)$ 次计算来评估对应次梯度。

IV. 不动点分析

在分析算法 1 的固定点行为之前，我们首先回顾一下帕累托平稳性 [27] 的概念。

定义 4 (帕累托站性)**.** *一个模型被称为帕累托平稳的，如果可以找到梯度 $\{\nabla J_k(w)\}_{k=1}^K$ 的一个凸组合，使得结果为全零向量。因此，所有帕累托平稳点 $\mathcal{P}^{st}$ 的集合由以下给出：*

$$\mathcal{P}^{st} = \Big\{w \in \mathbb{R}^d : \min_{p \in \Delta^K} ||G(w)p|| = 0\Big\}. \tag{20}$$

注意弱帕累托最优性意味着帕累托平稳性，即 $\mathcal{P} \subseteq \mathcal{P}^{\mathrm{st}}$ [28, Lemma 2.2]。我们现在 characterize EPO-AL 的不动点行为。

定理 1 (不动点分析)**.** *假设 算法 1 收敛到一对不动点 w_∞ 和 p_∞。然后 w_∞ 是帕累托平稳的且为公平。*

证明. 我们从将 w_∞ 替换 w_i 和 w_{i-1} 以及将 p_∞ 替换 p_i 和 p_{i-1} 在 (19a)–(19b) 开始，得到：

$$G(w_\infty)\big([p_\infty]_+ + \eta L_r \mathcal{J}(w_\infty)\big) = 0; \tag{21}$$

$$L_r \mathcal{J}(w_\infty) = 0. \tag{22}$$

从 (22)，(21) 简化为：

$$G(w_\infty)[p_\infty]_+ = \sum_{k=1}^K [p_{k,\infty}]_+ \nabla J_k(w_\infty) = 0, \tag{23}$$

这确保了在 $[p_\infty]_+$ 包含至少一个非零项的前提下，w_∞ 是帕累托平稳的,，我们将在下面进一步调查这一点。鉴于 (13)，第二个条件 (22) 暗示了 w_∞ yields $r_1 J_1(w_\infty) = \cdots = r_K J_K(w_\infty)$，从而导致 $w_\infty \in \mathcal{F}_r$ 满足公平性条件。

唯一剩下的步骤是证明 p_∞ 中至少包含一个严格正的元素。为此，从 (14) 观察到 $r^{-1} \triangleq [r_1^{-1}, \ldots, r_K^{-1}]$ 在 L_r 的零空间中：

$$\begin{aligned} L_r r^{-1} &= \mathrm{diag}(r)\Big(I_{K\times K} - \frac{1}{K}\mathbb{1}_K \mathbb{1}_K^\top\Big)\mathrm{diag}(r) r^{-1} \\ &= \mathrm{diag}(r)\Big(I_{K\times K} - \frac{1}{K}\mathbb{1}_K \mathbb{1}_K^\top\Big)\mathbb{1}_K = 0. \end{aligned} \tag{24}$$

因此，通过对 (19b) 与 r^{-1} 取内积，我们得到有：

$$(r^{-1})^\intercal p_i = (r^{-1})^\intercal p_{i-1} + \mu (r^{-1})^\intercal L_r \mathcal{J}(w_{i-1}) = (r^{-1})^\intercal p_{i-1}. \tag{25}$$

于当 i 迭代回到 p_0，我们发现那个：

$$\sum_{k=1}^K \frac{p_{k,i}}{r_k} = \sum_{k=1}^K \frac{p_{k,0}}{r_k} = \sum_{k=1}^K \frac{1}{K r_k} \triangleq \epsilon > 0. \tag{26}$$

为了使 $\sum_{k=1}^K p_{k,i}/r_k$ 大于 ϵ，必须存在至少一个 k' 使得 $p_{k',i}/r_{k'} \geq \epsilon$ 并且因此 $p_{k',i} \geq \epsilon r_{k'} \geq \epsilon \min_k r_k$。我们得出结论，对于所有 i 我们有：

$$\max_k p_{k,i} \geq \Big(\min_k r_k\Big)\left(\sum_{k=1}^K \frac{1}{K r_k}\right). \tag{27}$$

在之上一个假设 p_i 趋向于不动点 p_∞ 并取极限得到期望的结果。 □

推论 1 (凸目标函数)**.** *假设目标函数 $J_k(w)$ 对所有 $k = 1, \ldots, K$ 都是凸的。那么，$w_\infty \in \mathcal{E}_r = \mathcal{P} \cap \mathcal{F}_r$ 是精确帕累托最优，并解决了 min-max 问题 (1)。*

证明. 结果在认识到帕累托平稳性意味着凸目标下的弱帕累托最优性 [28, Lemma 2.2] 后立即得出。因此，$w^\infty \in \mathcal{P}$。定理 1 已经建立 $w^\infty \in \mathcal{F}_r$。在这些条件下，命题 1 确保 $w^\infty \in \mathcal{E}_r$ 并且也解决了 (1)。 □

V. 经验评估

我们通过一对个合成实验来实证评估我们的算法：当目标函数 $\{J_k(w)\}_{k=1}^K$ 是（i）全部凸的和（二）全部非凸的 [1]。特别是对于凸场景，我们考虑 (我)$J_k(w) = \sqrt{1 + ||w - w_k||^2} - 1$；对于非凸场景，我们采用 (二)$J_k(w) = 1 - e^{-||w-w_k||^2}$ 从 [9] 调整以处理多于两个目标的情况。具体地说，点 K 锚点 $\{w_k\}_{k=1}^K$ 是在单位 $(d-1)$ 表面上均匀随机选择的，我们还通过在内部的概率单形 Δ_+^K 中均匀随机采样来选择偏好向量 r，我们将其定义为 $\Delta_+^K = \{y \in \Delta^K : y_k > 1/3K\ \forall k\}$ 我们施加如此严格的正性以避免某些目标被基本上忽略的极端情况。。初始模型 w_0 是通过从的单位 $(d-1)$-球中随机采样选择的。我们通过运行 30 次独立实验来考虑这些随机性，除非另有说明。

我们将提出的算法到 (我) 的次梯度算法 (2) 进行比较，其中活跃索引 k^{active} 通过随机打破平局来选择；(二)the 光滑最大方法 [5], [13], [15] 更新 $w \leftarrow w - \mu \nabla \mathrm{LSE}_\tau[r_1 J_1(w), \ldots, r_K J_K(w)]$,， 其中 $\mathrm{LSE}_\tau[v_1, \ldots, v_K]$ 是定义为 $\mathrm{LSE}_\tau[v_1, \ldots v_K] = \log \sum_{k=1}^K e^{v_k/\tau}$ 的光滑最大函数；以及 EPO 搜索 [8],，

[1] 代码可在以下位置获取 https://github.com/sangwoo-p/EPO_AL

其形式为 $w \leftarrow w - \mu G(w)\beta$，其中 $\beta \in \Delta^K$ 通过在每次迭代中求解一个 K 维线性规划问题 [8] 来选择。

A. 优化轨迹的可视化

我们首先可视化当所有目标均为非凸在图中 1 时考虑的方案优化轨迹。我们省略了光滑极大方法，因为它未能收敛到对于足够大的 τ 所对应的最优点 [13], [14]，这给出了给我们一个个不同的优化轨迹沿着，带有子梯度的算法 (2)。我们设置 $d = 3$ 和 $K = 2$。一个一个有趣的观察结果是现有方法优先收敛于公平约束后再寻找 Pareto 前沿，而提出的 EPO-AL 算法能快速收敛到 Pareto 前沿并遍历它以寻找一种也满足满足公平性条件的解。我们为所有方案沿着设置 $\mu = 0.1$，并为 EPO-AL 设置 $\eta = 10$，$r = [0.2, 0.8]^\top$，并选择以下 [9] 的两个锚点。

B. 迭代/时间复杂度

我们现在考虑四个算法的迭代复杂度，通过测量达到指定目标精度所需的最小迭代次数需要。为了公平比较不同的算法，我们分别通过在大小为 10 的日志缩放网格中搜索 $\mathcal{G}_\mu = [10^{-3}, 10^{-1}]$ 来设置每个算法的步长 μ。至于该 EPO-AL 和 smooth-max 算法，我们也分别通过搜索 $\mathcal{G}_\eta = [10^{-1}, 10^2]$ 和 $\mathcal{G}_\tau = [10^{-2}, 10]$ 来设置惩罚参数 η 和温度参数 τ，都在大小为 10 的对数缩放网格中。我们将迭代的最大次数设置为 1000，并将模型的维度设置为 $d = 100$，即 $w \in \mathbb{R}^{100}$。

具体来说，我们将目标性能 $J^\star$ 定义为在整个可能的步长选择中由次梯度算法 (2) 达到的最小值。然后我们评估在固定选择的 μ（对于所有算法）以及 η（对于 EPO-AL）和 τ（对于 smooth-max）的情况下迭代复杂度，通过 $i^o(\mu, \eta, \tau) = \min\{i : |\max_k r_k J_k(w_i) - J^\star| \leq \epsilon\}$ 设置到 $\epsilon = 0.01$ 的的容差水平。然后，我们最终评估每个方案的迭代复杂度 i^o，通过选择可能的 μ、η 和 τ 中的最小值 $i^o(\mu, \eta, \tau)$，即针对的次梯度算法 (2) 和 EPO 搜索的 $i^o = \min_{\mu \in \mathcal{G}_\mu} i^o(\mu, \eta, \tau)$；针对平滑-max 的 $i^o = \min_{(\mu,\tau) \in \mathcal{G}_\mu \times \mathcal{G}_\tau} i^o(\mu, \eta, \tau)$；以及针对 EPO-AL 的 $i^o = \min_{(\mu,\eta) \in \mathcal{G}_\mu \times \mathcal{G}_\eta} i^o(\mu, \eta, \tau)$。

图 2（左）显示了迭代复杂度作为目标数量 K 的函数，对于凸函数和非凸函数 $J_k(w)$。观察到无论是的经典的 EPO 搜索 [8] 还是提出的 EPO-AL 算法都与目标数量, 成比例增长，而子梯度算法（2）却在增加的目标数量时表现不佳在。由于迭代次数无法捕捉每次迭代相关的计算复杂度（见图 1），我们接下来研究达到目标性能 J^* 所需的最小总数复杂度。

图 2（右）显示了时钟时间复杂度 t^o，定义为处理迭代次数 i^o 所需的实际总时间。时钟时间是在 Apple M1 硬件上评估的。EPO 搜索 [8] 在每次迭代中涉及线性程序的求解，这导致了更高的每迭代复杂度，并因此与仅涉及单一时尺度的所提出的 EPO-AL 策略相比具有更高的总运行时间。

VI. 结论

一种新的通过精确帕累托优化进行最小最大优化的算法被提出。为了推导出该策略，我们使用了增广拉格朗日方法下的原对偶共识技术，从而得到一个比基于次梯度-的方法在目标数量增加时有更好的扩展性，并且每轮迭代的复杂度低于其他基于平滑化-的方法。实验结果表明，所提出的算法通过施加较低的总复杂度达到了目标性能，相比于其他基准而言，展示了其随着目标数量增加的可扩展性。

参考文献

[1] K. Miettinen, *Nonlinear multiobjective optimization*, vol. 12. Springer Science & Business Media, 1999.

[2] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, “A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550–586, 2016.

[3] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[4] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in *ICML*, pp. 4615–4625, PMLR, 2019.

[5] Q. Hou, Y. Cai, Q. Hu, M. Lee, and G. Yu, “Joint resource allocation and trajectory design for multi-UAV systems with moving users: Pointer network and unfolding,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 3310–3323, 2022.

[6] Y. Wang, C. Yang, and M. Peng, “Hybrid precoding with low-resolution pss for wideband Terahertz communication systems in the face of beam squint,” *arXiv preprint arXiv:2406.16303*, 2024.

[7] S. M. Hamidi, A. Bereyhi, S. Asaad, and H. V. Poor, “Over-the-air fair federated learning via multi-objective optimization,” *arXiv preprint arXiv:2501.03392*, 2025.

[8] D. Mahapatra and V. Rajan, “Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization,” in *ICML*, pp. 6597–6607, PMLR, 2020.

[9] X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong, “Pareto multi-task learning,” *NeurIPS*, vol. 32, 2019.

[10] H. H. Bauschke, P. L. Combettes, H. H. Bauschke, and P. L. Combettes, *Correction to: convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2017.

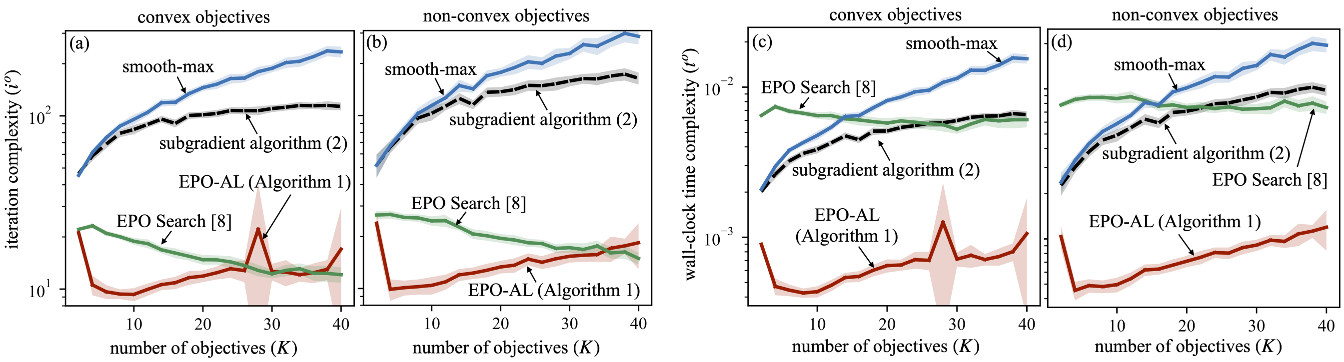


图 2. 迭代复杂度 i^o(a,b) 和时钟时间复杂度 t^o(c,d) 作为目标数量 K 的函数。结果是基于 30 次独立实验移除最小值和最大值后，平均得出的，其中每次实验假设不同的偏好向量 r 和不同的初始模型 w_0。阴影区域对应于 99% 置信区间。

[11] C. Ras, M. Tam, and D. Uteda, “Identification of active subfunctions in finite-max minimisation via a smooth reformulation,” *arXiv preprint arXiv:2404.10326*, 2024.

[12] I. Zang, “A smoothing-out technique for min—max optimization,” *Mathematical Programming*, vol. 19, pp. 61–77, 1980.

[13] H. Gokcesu, K. Gokcesu, and S. S. Kozat, “Accelerating min-max optimization with application to minimal bounding sphere,” *arXiv preprint arXiv:1905.12733*, 2019.

[14] A. Epasto, M. Mahdian, V. Mirrokni, and E. Zampetakis, “Optimal approximation-smoothness tradeoffs for soft-max functions,” *NeurIPS*, vol. 33, pp. 2651–2660, 2020.

[15] X. Lin, X. Zhang, Z. Yang, F. Liu, Z. Wang, and Q. Zhang, “Smooth Tchebycheff scalarization for multi-objective optimization,” *arXiv preprint arXiv:2402.19078*, 2024.

[16] J. G. Lin, “On min-norm and min-max methods of multi-objective optimization,” *Mathematical programming*, vol. 103, no. 1, pp. 1–33, 2005.

[17] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” *arXiv preprint arXiv:1905.10497*, 2019.

[18] D. Mahapatra and V. Rajan, “Exact Pareto optimal search for multi-task learning and multi-criteria decision-making,” *arXiv preprint arXiv:2108.00597*, 2021.

[19] M. Momma, C. Dong, and J. Liu, “A multi-objective/multi-task learning framework induced by Pareto stationarity,” in *ICML*, pp. 15895–15907, PMLR, 2022.

[20] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in ad hoc wsns with noisy links —Part I: Distributed estimation of deterministic signals,” *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[21] Z. J. Towfic and A. H. Sayed, “Stability and performance limits of adaptive primal-dual networks,” *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2888–2903, 2015.

[22] S. Vlaski, S. Kar, A. H. Sayed, and J. M. Moura, “Networked signal and information processing: Learning by multiagent systems,” *IEEE Signal Processing Magazine*, vol. 40, no. 5, pp. 92–105, 2023.

[23] D. P. Bertsekas, “Nonlinear programming,” *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[24] K. J. Arrow, L. Hurwicz, H. Uzawa, H. B. Chenery, S. Johnson, and S. Karlin, *Studies in linear and non-linear programming*, vol. 2. Stanford University Press Stanford, 1958.

[25] S. A. Alghunaim and A. H. Sayed, “Linear convergence of primal–dual gradient methods and their performance in distributed optimization,” *Automatica*, vol. 117, p. 109003, 2020.

[26] L. Chen, H. Fernando, Y. Ying, and T. Chen, “Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance,” *NeurIPS*, vol. 36, pp. 70045–70093, 2023.

[27] J.-A. Désidéri, “Multiple-gradient descent algorithm (MGDA) for multiobjective optimization,” *Comptes Rendus Mathematique*, vol. 350, no. 5-6, pp. 313–318, 2012.

[28] H. Tanabe, E. H. Fukuda, and N. Yamashita, “Proximal gradient methods for multiobjective optimization and their applications,” *Computational Optimization and Applications*, vol. 72, pp. 339–361, 2019.