FERERO: A Flexible Framework for Preference-Guided Multi-Objective Learning

Lisha Chen [1], AFM Saif [1], Yanning Shen [2], Tianyi Chen [1]

[1]Rensselaer Polytechnic Institute [2]University of California, Irvine

Paper

Code

1. Why Do We Need a Flexible Framework for Preference-Guided MOL?

Multi-objective problems are common in AI.

$F(\theta) = [f_1(\theta); \dots; f_M(\theta)]$

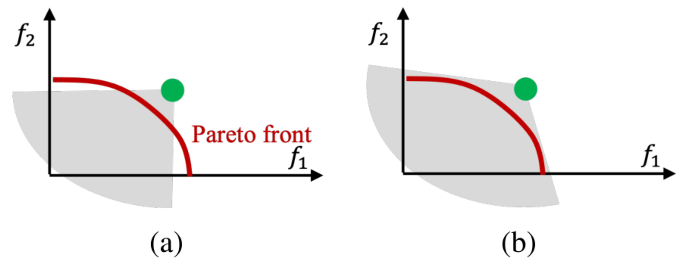
How to optimize a vector?

$\begin{bmatrix}10\\0\\10\end{bmatrix} \le_{Pareto} \begin{bmatrix}10\\10\\10\end{bmatrix}?$ $\quad \begin{bmatrix}-5\\2\\2\end{bmatrix} <_{LS} \begin{bmatrix}1\\1\\1\end{bmatrix}?$

Importance of general vector ordering

Pareto dominance (Fig.1a) is not enough: simply minimizing one objective achieves weak Pareto optimality.

Linear scalarization is not enough: objectives can be dominated by the one with the largest scale.



(a) (b)

Solid red curves: Pareto fronts
Green dots: reference points
Gray shaded regions: objectives dominating the reference points, under different C_A in both figures.

Figure 1. Illustration of relative preferences

Thus we introduce a more general vector ordering (preference 1) in Fig. 1b.

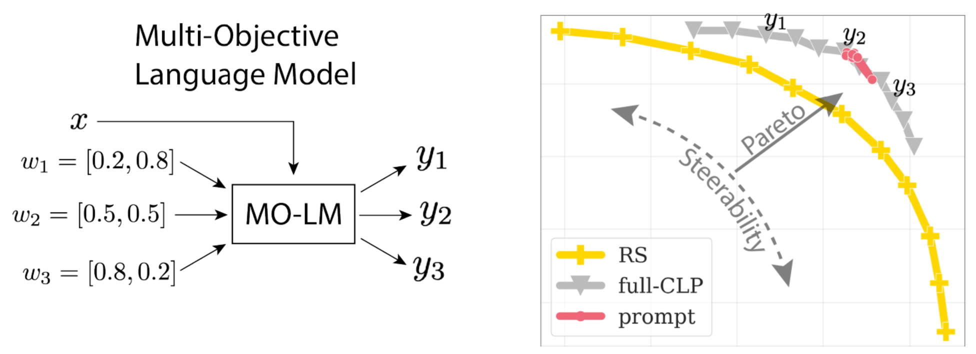
Preference 1: relative, determines the general dominance & optimality.

can be captured by the partial order, determines improving directions, e.g., induced by a cone C_A, $C_A = R_+^M$ corresponds to Pareto dominance (Fig. 1a).

e.g., a user gives at least α relative importance to each objective with $\alpha \in (0, 0.5)$, then this can be achieved by defining a partial order induced by the cone (example in [2]):
$C = \{F \in R^2 \,|\, \alpha f_1 + (1-\alpha) f_2 \ge 0, (1-\alpha) f_1 + \alpha f_2 \ge 0\}$

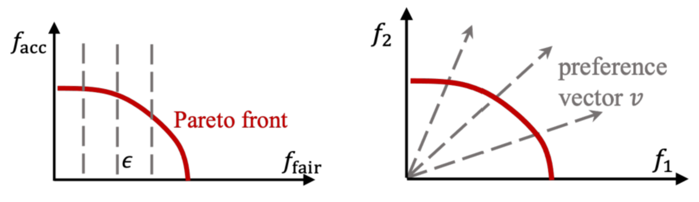
$F - F' \in -C$ means general dominance

Importance of constraints



Constraints can improve steerability in MO-LM [Kaiwen Wang et al. '24]

Preference 2: absolute, can be captured by the constraints



(a) (b)

Solid red curves: Pareto fronts
Gray dashed lines: absolute preference constraints.

Figure 2. Illustration of absolute preferences

$\min_{\text{under preference 1}} F(\theta)$, s.t. preference 2($\theta$)

2. How to Solve Preference-Guided MOL?

Main program: $\min_{C_A} F(\theta)$ s.t. $H(\theta) = 0, G(\theta) \le 0$

Derivation of the subprogram to find the best update at each iteration

Given update $\Delta\theta$, approximate the objectives & constraints by:

$$F(\theta + \Delta\theta) \approx F(\theta) + \nabla F(\theta)^\top \Delta\theta + \frac{1}{2}\Delta\theta^\top \nabla^2 F(\theta) \Delta\theta$$

approximate the Hessian of F by identity

improvement defined by general partial order

$$\min_{c,\,d} c + \frac{1}{2}\| d \|^2, \quad \text{s.t.}\ A\nabla F(\theta)^\top d \le c \cdot A\mathbf{1}$$

Idea similar to SQP: Use local quadratic approximation to the objectives

$$G(\theta + \Delta\theta) \approx c_g G(\theta) + \nabla G(\theta)^\top d \le 0$$
$$H(\theta + \Delta\theta) \approx c_h H(\theta) + \nabla H(\theta)^\top d = 0$$

$$\lambda^*(\theta) \in \arg\min_{\lambda \in \Omega_\lambda} \varphi(\lambda; \theta) := \frac{1}{2}\left\| \nabla F(\theta) A_{ag}^\top \lambda \right\|^2 - c_g \lambda_g^\top G(\theta) - c_h \lambda_h^\top H(\theta)$$

The optimal direction: $d^*(\theta) = -[\nabla F(\theta) A^\top, \nabla G(\theta), \nabla H(\theta)] \lambda^*(\theta)$

Algorithm update

Update $\lambda^*(\theta_t)$: $\lambda^*(\theta_t) = \operatorname{argmin}_{\lambda \in \Omega_\lambda} \varphi(\lambda; \theta_t)$

Update θ_t along $d^*(\theta_t)$: $\theta_{t+1} = \theta_t + \alpha_t d^*(\theta_t)$

Single-loop variant

The first single-loop primal algorithm with convergence guarantees

Update λ_t : $\lambda_{t+1} = \Pi_{\Omega_\lambda}(\lambda_t - \nabla_\lambda \varphi(\lambda_t; \theta_t))$

$d_t = -[\nabla F(\theta_t) A^\top, \nabla G(\theta_t), \nabla H(\theta_t)] \lambda_{t+1}$

Update θ_t along d_t: $\theta_{t+1} = \theta_t + \alpha_t d_t$

Stochastic variant

Use the double sampling idea introduced in MoDo [6] to update λ_t

3. Theoretical Analysis

KKT condition

$\nabla F(\theta) A^\top \lambda_f + \nabla G(\theta) \lambda_g + \nabla H(\theta) \lambda_h = 0$ stationarity

$\lambda_f \in \Delta^M, \lambda_g \in R_+^{M_g}, \lambda_h \in R^{M_h}$ dual feasibility

$G(\theta) \le 0, H(\theta) = 0$ primal feasibility

$\lambda_g^*(\theta)^\top [-G(\theta)]_+ = 0$ complementary slackness

When constraints are linear functions of objectives, under some additional conditions, the **PMOL calmness** holds, thus the KKT condition is a necessary condition for optimality.

Merit function (KKT score)

$$J_1(\theta) = \underbrace{\| d^*(\theta) \|^2}_{\text{stationarity}} + \underbrace{\lambda_g^*(\theta)^\top [-G(\theta)]_+}_{\text{complementary slackness}} + \underbrace{\| [G(\theta)]_+ \|_1 + \| H(\theta) \|_1}_{\text{feasibility}}$$

$$J_2(\theta_t) = \underbrace{\| \nabla F(\theta_t) A_{ag}^\top \lambda_t \|^2}_{\text{stationarity}} + \underbrace{\| H(\theta_t) \|^2}_{\text{feasibility}}$$

ℓ_2-norm

J_2 is used for equality constraints only, i.e., $M_g = 0$

Convergence rates

For general nonconvex functions, under smoothness assumptions

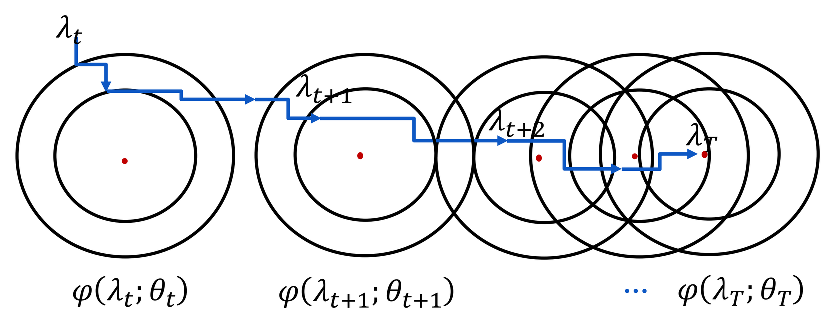
Double-loop variant: $\frac{1}{T}\sum_{t=0}^{T-1} J_1(\theta_t) = O(T^{-1})$

Single-loop variant: $\frac{1}{T}\sum_{t=0}^{T-1} J_1(\theta_t) = O(T^{-\frac{1}{6}})$

Single-loop variant (sharper): $\frac{1}{T}\sum_{t=0}^{T-1} J_2(\theta_t) = O(T^{-1})$

Stochastic variant (sharper): $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[J_2(\theta_t)] = O(T^{-\frac{1}{2}})$

with additional regularity assumptions



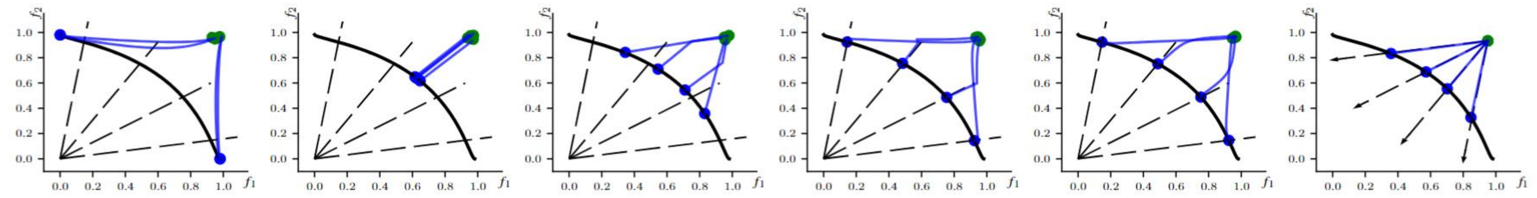
Key to obtain a sharper single-loop analysis is to use a different merit function and exploiting smoothness of $\varphi(\lambda; \theta)$.

Intuitive explanation: the subprogram objective $\varphi(\lambda; \theta)$ w.r.t. λ is **smooth & convex**; its drift is small!

4. Empirical Results

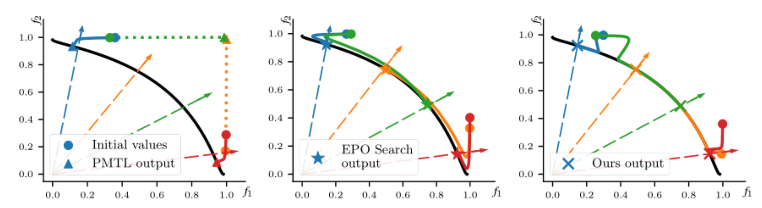
Toy experiments

$F(\theta) = \left(1 - e^{-\|\theta - \frac{1}{\sqrt{q}}\mathbf{1}\|_2^2},\ 1 - e^{-\|\theta + \frac{1}{\sqrt{q}}\mathbf{1}\|_2^2}\right).$



(a) LS (b) MGDA (c) PMTL (d) EPO (e) FERERO (f) FERERO

- ❏ PMTL does not align exactly to preference vectors.
- ❏ EPO only aligns to preference vectors passing through the origin.



(d) PMTL (e) EPO (f) FERERO

- ❏ PMTL does not allow controlled ascent, thus not converging in some problems.
- ❏ EPO & FERERO allow controlled ascent and converge in those problems.

Multi-MNIST classification

Table 2: Hypervolumes of different methods ($\times 10^{-2}$)

Datasets	LS	PMTL [23]	EPO [29]	XWC-MGDA [32]	FERERO
Multi-MNIST loss	1.68	1.41	1.35	1.42	**1.97**±0.21
Multi-Fashion loss	6.75	5.90	6.02	6.77	**7.76**±0.18
Multi-F+M loss	3.63	3.03	3.76	**3.89**	3.82±0.21
Multi-MNIST accuracy	0.19	0.15	0.15	0.16	**0.24**±0.04
Multi-Fashion accuracy	0.99	0.87	0.87	0.99	**1.17**±0.07
Multi-F+M accuracy	0.48	0.40	0.50	0.52	**0.53**±0.04
Emotion loss	0.0258	0.0230	**0.0366**	0.0348	0.0357±0.0006

Hypervolume measures the relative distance to the Pareto front of all solutions

Multi-lingual speech recognition

$$\min_\theta\ F(\theta) := \left(\underbrace{f_p(\theta)}_{\text{self-supervised loss}}, \overbrace{f_t^{\text{ch}}(\theta), f_t^{\text{en}}(\theta)}^{\text{supervised loss}}\right)^\top \ \text{s.t.}\ f_p(\theta) \le \epsilon_1,\ f_t^{\text{ch}}(\theta) - f_t^{\text{en}}(\theta) = \epsilon_2$$

- ❏ EPO & PMTL do not capture flexible preferences to solve this problem.

Table 3: WERs (%) on Librispeech and AISHELL v1.

Method	English	Chinese	Average
Komatsu et al. [19]	7.11	-	-
w/o CPC [38]	11.8	10.2	11.0
Init. (M2ASR) [38]	7.3	6.2	6.7
LS-FT	6.8	5.9	6.4
FERERO-FT	**5.4**	**4.9**	**5.1**

Check out more results & benchmarks in the paper.