



# **PHƯƠNG PHÁP HƯỚNG GIẢM SÂU NHẤT TỰ THÍCH NGHI CHO HỌC ĐA NHIỆM**

Đồ án nghiên cứu  
GVHD: TS. Trần Ngọc Thăng

Sinh viên: Nguyễn Anh Minh

# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc
- 4 Bài toán học đa nhiệm
- 5 Kết luận

# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc
- 4 Bài toán học đa nhiệm
- 5 Kết luận

# Giới thiệu bài toán Multi-task Learning

- Học đa nhiệm (Multi-task learning - MTL) là một lĩnh vực con trong Học máy, trong đó nhiều nhiệm vụ khác nhau được giải quyết cùng một lúc, đồng thời tận dụng sự tương đồng và khác biệt giữa các nhiệm vụ nhằm tăng độ chính xác dự đoán của mô hình.<sup>1</sup>
- Ứng dụng thực tế:
  - Y Học [1], [5]
  - Thị Giác Máy Tính(CV) [9], [11]
  - Xử lý Ngôn Ngữ Tự Nhiên (NLP) [2]

---

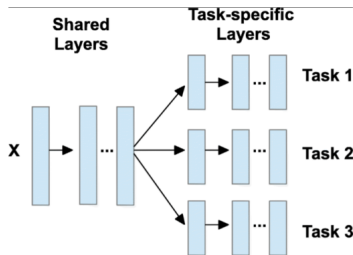
<sup>1</sup>Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey.

# Mô hình hóa MTL dưới dạng bài toán tối ưu đa mục tiêu

- Bài toán MTL có thể được mô hình dưới dạng một bài toán tối ưu đa mục tiêu MOP <sup>1</sup>.
- Các mô hình giải bài toán MTL sẽ tối ưu đồng thời  $m$  tác vụ với hàm véc tơ mất mát :

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \dots, \mathcal{L}_m(\theta))^T, \quad (\text{MTL})$$

trong đó,  $\mathcal{L}_i(\theta)$  là hàm mất mát cho tác vụ thứ  $i$  với bộ tham số  $\theta$  của mô hình.

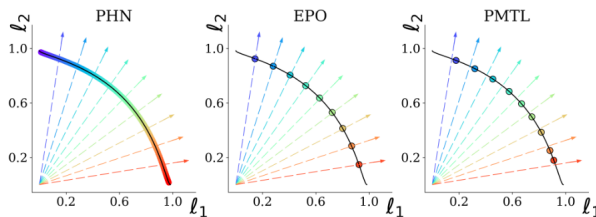


<sup>1</sup>Sener et al (2018). Multi-task learning as multi-objective optimization. Advances in neural information processing systems

# Phương pháp giải bài toán MOP

Hai hướng tiếp cận chính để giải bài toán MOP:

- **Gradient - free methods:** NSGA, ToP [10],... thực thi hiệu quả với lớp hàm mục tiêu và ràng buộc không lồi. Tuy nhiên, chỉ phù hợp với lớp bài toán có scale nhỏ, không phù hợp với các mô hình DNN có scale lớn <sup>1</sup>.
- **Gradient - based methods:** PMTL [8], PHN-HVI [4],... phù hợp với large-scale Deep Neural Network.



<sup>1</sup>Navon et al., Learning the Pareto Front with Hypernetworks, ICLR 2021

# Vấn đề khi giải bài toán MTL dưới dạng MOP

- Giải bài toán MTL dưới dạng MOP phát sinh các vấn đề:
  - Hàm mục tiêu không lồi.
  - Tối ưu có ràng buộc.
  - Tập ràng buộc có thể không bị chặn.
- Phương pháp Gradient-based giải bài toán MTL thường chỉ áp dụng cho trường hợp tối ưu không ràng buộc và cỡ bước (hệ số học) cố định. <sup>1</sup>

---

<sup>1</sup>Lin, Xi, et al. "Pareto multi-task learning." Advances in neural information processing systems 32 (2019).

# Phương pháp đề xuất

Nhằm giải bài toán MTL dưới dạng MOP hiệu quả, tác giả đề xuất:

- Sử dụng phương pháp hướng giảm sâu nhất đa mục tiêu giúp giải bài toán tối ưu đa mục tiêu không lồi với tập ràng buộc không bị chặn hiệu quả và phù hợp bài toán Large-scale DNN.
- Kết hợp việc cập nhật cỡ bước theo phương pháp tư thích nghi (Adaptive) <sup>1</sup> để tăng tốc độ hội tụ của thuật toán.

---

<sup>1</sup>Konnov, I. V. "Simplified versions of the conditional gradient method." Optimization 67.12 (2018): 2275-2290.



# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc**
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc
- 4 Bài toán học đa nhiệm
- 5 Kết luận

# Bài toán quy hoạch đa mục tiêu không ràng buộc (1)

- Xét bài toán tối ưu đa mục tiêu:

$$\text{Min}_{x \in U} F(x), \quad (\text{MOP}(F, U))$$

trong đó tập  $U \subseteq \mathbb{R}^n$  khác rỗng, lồi đóng và hàm véc tơ  $F : U \rightarrow \mathbb{R}^m$  là một hàm khả vi trên tập mở chứa  $U$ . Giả sử  $U = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, \dots, l\}$ .

- Tiến hành giải bài toán  $(\text{MOP}(F, U))$  trong trường hợp  $U = \mathbb{R}^n$ .
- Xét  $x \in \mathbb{R}^n$ , định nghĩa hướng giảm sâu nhất  $s(x)$  tại  $x$  là nghiệm của bài toán tối ưu sau:

$$\begin{cases} \min & \max_{j=1, \dots, m} \nabla F_j(x)^T s + \frac{1}{2} s^T s \\ \text{v.đ.k} & s \in \mathbb{R}^n, \end{cases} \quad (P_1)$$

# Bài toán quy hoạch đa mục tiêu không ràng buộc (2)

- Nghiệm tối ưu cho  $(P_1)$  sẽ được ký hiệu bằng  $\Theta(x)$ . Do đó,

$$\Theta(x) = \inf_{s \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T s + \frac{1}{2} s^T s, \quad (1)$$

và

$$s(x) = \arg \min_{s \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T s + \frac{1}{2} s^T s \quad (2)$$

# Tính dừng của nghiệm

## Bổ đề 2.1

Với  $x \in \mathbb{R}^n$  thì

$$\Theta(x) = -\frac{1}{2}s(x)^T s(x) \text{ và } s(x) = -\sum_{j=1}^m \lambda_j \nabla F_j(x).$$

với  $\lambda_j \geq 0, \sum_{j=1}^m \lambda_j = 1$ .

## Định lý 2.1 ([3])

- ❶ Với mọi  $x \in \mathbb{R}^n$  thì  $\Theta(x) \leq 0$ .
- ❷ Những điều kiện sau là tương đương
  - (a) Điểm  $x$  là điểm không dừng (non-critical).
  - (b)  $\Theta(x) < 0$ .
  - (c)  $s(x) \neq 0$ . Cụ thể,  $x$  là điểm dừng Pareto khi và chỉ khi  $\Theta(x) = 0$ .
- ❸ Hàm  $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$  liên tục trên  $\mathbb{R}^n$ .

# Thuật toán đề xuất (1)

---

**Thuật toán 1** Thuật toán hướng giảm sâu nhất tự thích nghi phiên bản 1

---

**Bước 1.** Thiết lập  $\kappa \in [0, 1]$ ,  $\sigma \in [0, 1]$ ,  $\alpha_1 \in (0, +\infty)$ , và  $x^1 \in \mathbb{R}^n$ . Đặt  $k := 1$ .

**Bước 2.** (Vòng lặp chính)

(a) Tìm hướng giảm thông qua việc giải bài toán sau:

$$\min_{\lambda_j} \left\| \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) \right\|^2$$

$$\text{v.đ.k: } \lambda_j \geq 0, \sum_{j=1}^m \lambda_j^k = 1$$

Từ đó xác định được nghiệm tối ưu:

$$\lambda^k = (\lambda_j^k), s(x^k) = - \left( \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) \right) \text{ và } \Theta(x^k) = -\frac{1}{2} \|s(x^k)\|^2.$$


---

## Thuật toán đề xuất (2)

---

(b) Nếu  $\Theta(x^k) = 0$  thì **dừng lại**. Ngược lại tiếp tục **Bước 2(c)**.

(c) Đặt  $x^{k+1} := x^k + \alpha_k s(x^k)$ .

(d) Tính toán cỡ bước:

Nếu

$$F(x^{k+1}) \leq F(x^k) + \sigma \langle JF(x^k), x^{k+1} - x^k \rangle \quad (3)$$

thì  $\alpha_{k+1} = \alpha_k$ , ngược lại đặt  $\alpha_{k+1} := \kappa \alpha_k$

**Bước 3.** Đặt  $k := k + 1$  và chuyển đến **Bước 2**.

---

# Sự hội tụ của Thuật toán 1

Định nghĩa:

$$T = \{x \in \mathbb{R}^n : F(x) \leq F(x^k), \forall k\}.$$

## Định lý 2.2

*Giả sử dãy  $\{x^k\}$  được tạo bởi Thuật toán 1. Khi đó, mỗi điểm tới hạn (nếu có) của dãy  $\{x^k\}$  là điểm dừng Pareto của bài toán  $(\text{MOP}(F, U))$  với  $U = \mathbb{R}^n$ .*

## Định lý 2.3

*Giả sử  $F$  là hàm tựa lồi trên  $\mathbb{R}^n$ . Nếu  $T \neq \emptyset$  thì  $\{x^k\}$  hội tụ đến điểm dừng Pareto.*

## Định lý 2.4

*Giả sử  $F$  là hàm giả lồi trên  $\mathbb{R}^n$ . Nếu  $T \neq \emptyset$  thì  $\{x^k\}$  hội tụ tới điểm hữu hiệu yếu.*

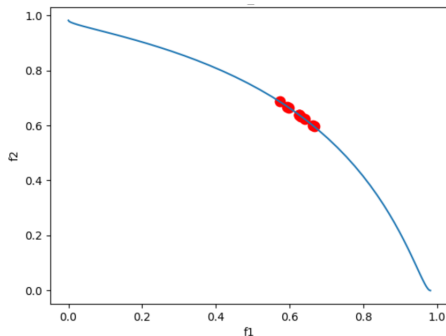
# Ví dụ minh họa Thuật toán 1

Xét bài toán tối ưu đa mục tiêu không lồi không ràng buộc:

$$\text{Min } F(x) = \left\{ 1 - \exp^{-\sum_{i=1}^d (x_i - \frac{1}{d})^2}, 1 - \exp^{-\sum_{i=1}^d (x_i + \frac{1}{d})^2} \right\}$$

v.đ.k:  $x \in \mathbb{R}^d$

Tiến hành thử nghiệm với  $d = 20$ .



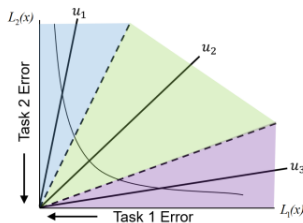
Các nghiệm Pareto thu được tập trung ở một vùng trên Pareto Front.



# Cải tiến Thuật toán 1 với véc tơ ưu tiên

- Xét tập  $K$  véc tơ ưu tiên  $\{u_1, u_2, \dots, u_K\} \in \mathbb{R}_+^m$ , tiến hành phân tách không gian ảnh thành các vùng không gian con  $\Omega_k$  ( $k = 1, \dots, K$ ), với

$$\Omega_k = \{v \in \mathbb{R}_+^m \mid \langle u_i, v \rangle \leq \langle u_k, v \rangle, \forall i = 1, \dots, K\}.$$



- Bài toán  $(\text{MOP}(F, U))$  với  $U = \mathbb{R}^n$  kết hợp các véc tơ ưu tiên có dạng:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} F(x) &= (F_1(x), F_2(x), \dots, F_m(x)) \\ \text{v.đ.k } \mathcal{G}_p(x) &= \langle u_p - u_k, F(x) \rangle \leq 0, \forall p = 1, \dots, K, \end{aligned} \quad (4)$$

# Thuật toán đề xuất phiên bản 2

---

## Thuật toán 2 Thuật toán hướng giảm sâu nhất tự thích nghi phiên bản 2

---

**Bước 1.** Thiết lập  $\kappa \in [0, 1], \sigma \in [0, 1], \alpha_1 \in (0, +\infty)$ , và  $x_r^1 \in \mathbb{R}^n$ . Đặt  $k := 1$ .

**Bước 2.** Tiến hành tìm điểm khởi tạo chấp nhận được  $x^1$  từ  $x_r^1$ .

**Bước 3.** (Vòng lặp chính)

(a) Tìm hướng giảm thông qua việc giải bài toán sau:

$$\min_{\lambda_j, \gamma_p} \left\| \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{p \in I_\epsilon(x^k)} \gamma_p^k \nabla \mathcal{G}_p(x^k) \right\|^2$$

$$\text{v.đ.k: } \lambda_j \geq 0, \gamma_p \geq 0, \sum_{j=1}^m \lambda_j^k + \sum_{p \in I_\epsilon(x^k)} \gamma_p^k = 1$$

Trong đó,  $I_\epsilon(x^k) := \{p \in 1, \dots, K | \mathcal{G}_p(x^k) \geq -\epsilon\}$ .

---

## Thuật toán đề xuất phiên bản 2 (2)

---

Từ đó xác định được nghiệm tối ưu:

$$\lambda^k = (\lambda_j^k), \quad \gamma^k = (\gamma_p^k), \quad s(x^k) = - \left( \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{p \in I_\epsilon(x)} \gamma_p^k \nabla \mathcal{G}_p(x^k) \right) \quad \text{và} \quad \Theta(x^k) = -\frac{1}{2} \|s(x^k)\|^2.$$

(b) Nếu  $\Theta(x^k) = 0$  thì **dừng lại**. Ngược lại tiếp tục **Bước 3(c)**.

(c) Đặt  $x^{k+1} := x^k + \alpha_k s(x^k)$ .

(d) Tính toán cỡ bước:

$$\text{Nếu } F(x^{k+1}) \leq F(x^k) + \sigma \langle JF(x^k), x^{k+1} - x^k \rangle$$

thì  $\alpha_{k+1} = \alpha_k$ , ngược lại đặt  $\alpha_{k+1} := \kappa \alpha_k$

**Bước 4.** Đặt  $k := k + 1$  và chuyển đến **Bước 3**.

---

# Sự hội tụ của Thuật toán 2

## Định lý 2.5

Giả sử dãy  $\{x^k\}$  được tạo bởi Thuật toán 2. Khi đó, mỗi điểm tới hạn (nếu có) của dãy  $\{x^k\}$  là điểm dừng Pareto của bài toán  $(\text{MOP}(F, U))$  với  $U = \mathbb{R}^n$ .

## Định lý 2.6

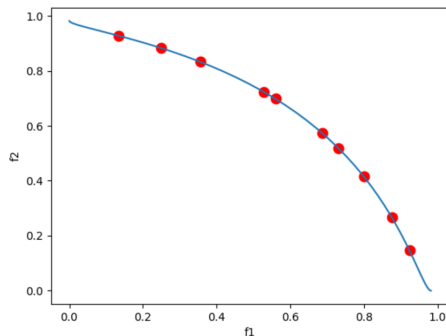
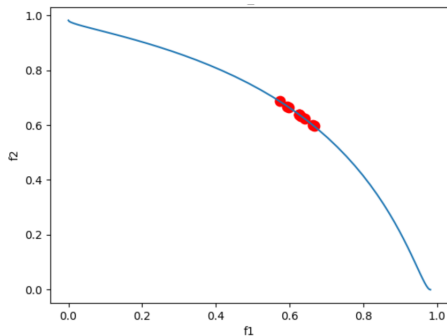
Giả sử  $F$  là hàm tựa lồi trên  $\mathbb{R}^n$ . Nếu  $T \neq \emptyset$  thì  $\{x^k\}$  hội tụ đến điểm dừng Pareto.

## Định lý 2.7

Giả sử  $F$  là hàm giả lồi trên  $\mathbb{R}^n$ . Nếu  $T \neq \emptyset$  thì  $\{x^k\}$  hội tụ tới điểm hữu hiệu yếu.

## Ví dụ minh họa Thuật toán 2

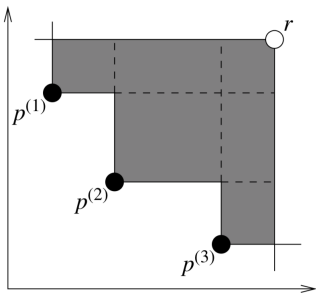
Khi tiến hành thử nghiệm Thuật toán 2 với cùng bài toán, với số điểm khởi tạo 10 điểm thu được kết quả:



Với đường màu xanh Pareto Front như hình bên phải, ta thấy các nghiệm tìm được đều thuộc Pareto Front và có trải đều trên toàn mặt.

# Chỉ số đánh giá

- Vùng không gian ảnh được tạo bởi (dominated) bởi Pareto front được gọi là Hypervolume <sup>2</sup>.
- Chỉ số Hypervolume càng cao thì Pareto front tìm được càng tốt.



<sup>2</sup>Zitzler et al (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach.

## Kết quả thử nghiệm của Thuật toán 2 (1)

So sánh hiệu quả của Thuật toán 2 với Thuật toán PMTL<sup>3</sup> bằng cách khởi tạo ngẫu nhiên cùng số điểm, cùng số vòng lặp để tìm nghiệm cho Ví dụ 1.

Số điểm khởi tạo (điểm)	Số vòng lặp (lần)	Số lần chạy (lần)	Thời gian trung bình thuật toán PMTL (s)	Thời gian trung bình Thuật toán 2 (s)	HV trung bình PMTL	HV trung bình Thuật toán 2
50	200	40	264.74	<b>224.06</b>	1.48	<b>1.53</b>
50	300	10	245.86	<b>181.84</b>	1.50	<b>1.53</b>
60	200	10	259.52	<b>236.51</b>	1.47	<b>1.51</b>

**Bảng:** So sánh Thuật toán 2 và thuật toán PMTL giải Ví dụ 1

<sup>3</sup>Lin, Xi, et al. "Pareto multi-task learning." Advances in neural information processing systems 32 (2019).

## Kết quả thử nghiệm của Thuật toán 2 (2)

Để so sánh sự vượt trội của thời gian giải ra nghiệm tối ưu trải đều trên Pareto Front ta tiến hành chạy Thuật toán 2 với số vòng lặp ít hơn so với Thuật toán PMTL [8].

Số điểm khởi tạo	Thuật toán PMTL			Thuật toán 2		
	HV trung bình	Thời gian (s)	Số vòng lặp	HV trung bình	Thời gian (s)	Số vòng lặp
10	1.49	2.01	100	<b>1.49</b>	<b>1.37</b>	<b>50</b>
20	1.53	4.38	100	<b>1.53</b>	<b>2.45</b>	<b>80</b>
30	1.49	17.97	100	<b>1.51</b>	<b>10.79</b>	<b>80</b>
40	1.49	54.01	100	<b>1.49</b>	<b>35.73</b>	<b>75</b>
50	1.44	99.42	100	<b>1.47</b>	<b>67.67</b>	<b>75</b>

**Bảng:** Kết quả so sánh Thuật toán 2 và thuật toán PMTL trong các trường hợp khác nhau.



# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc**
- 4 Bài toán học đa nhiệm
- 5 Kết luận

# Bài toán quy hoạch đa mục tiêu có ràng buộc

- Cho điểm  $x \in \mathbb{R}^n$ , ký hiệu  $P_U(x)$  là ảnh của điểm  $x$  trên  $U$ :

$$P_U(x) := \operatorname{argmin}\{\|z - x\| : z \in U\}. \quad (5)$$

- Xét tập chỉ số  $I := \{1, \dots, l\}$  với  $\varepsilon \in \mathbb{R}_+$  và  $x \in \mathbb{R}^n$ , định nghĩa tập  $I_\varepsilon$  như sau:

$$I_\varepsilon(x) := \{i \in I \mid g_i(x) \geq -\varepsilon\}.$$

- Tiến hành giải bài toán tối ưu sau nhằm tìm hướng giảm cho bài toán  $(\text{MOP}(F, U))$ :

$$\begin{aligned} & \min \alpha \\ & \text{v.đ.k } (\nabla F_j(x))^\top s \leq \alpha, \quad j = 1, \dots, m, \\ & (\nabla g_i(x))^\top s \leq \alpha, \quad i \in I_\varepsilon(x), \\ & \|s\|_\infty \leq 1, \end{aligned} \quad (P_2)$$

- Ký hiệu  $\Theta(x, \varepsilon)$  là giá trị tối ưu xấp xỉ tìm được.

# Cải tiến với véc tơ ưu tiên

Để nghiệm tìm được trải đều trên mặt Pareto, ta xét bài toán:

$$\min_{x \in U} F(x) = (F_1(x), F_2(x), \dots, F_m(x))$$

$$\text{v.đ.k } \mathcal{G}_p(x) = \langle \mathbf{u}_p - \mathbf{u}_k, F(x) \rangle \leq 0, \forall p = 1, \dots, K,$$

$\Rightarrow$  Tích hợp các ràng buộc  $\mathcal{G}_p, p = \{1, \dots, K\}$  vào tập chấp nhận được  $U$ .

# Thuật toán đề xuất giải (MOP( $F, U$ )) (1)

---

## Thuật toán 3 Thuật toán đề xuất

---

**Bước 1.** Thiết lập  $\kappa \in [0, 1]$ ,  $\sigma \in [0, 1]$ ,  $\alpha_1 \in (0, 1)$ , và  $x_r^1 \in \mathbb{R}^n$ . Khởi tạo  $k := 1$ .

**Bước 2.** Tìm điểm khởi tạo chấp nhận được  $x^1$  từ  $x_r^1$ .

**Bước 3.** (Vòng lặp chính)

(a) Tìm hướng giảm chấp nhận được bằng cách giải bài toán sau:

$$\min_{\lambda_j, \beta_i, \gamma_p} \left\| \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{i \in I_\epsilon(x^k)} \beta_i^k \nabla g_i(x^k) + \sum_{p \in I_\epsilon(x^k)} \gamma_p^k \nabla \mathcal{G}_p(x^k) \right\|^2$$

$$\text{v.đ.k } \lambda_j \geq 0, \beta_i \geq 0, \gamma_p \geq 0, \sum_{j=1}^m \lambda_j^k + \sum_{i \in I_\epsilon(x^k)} \beta_i^k + \sum_{p \in I_\epsilon(x^k)} \gamma_p^k = 1$$

Trong đó,  $I_\epsilon(x^k) := \{i \in 1, \dots, l; p \in 1, \dots, K \mid \mathcal{G}_p(x^k) \geq 0 \text{ và } g_i(x^k) \geq -\epsilon\}$ .

---

# Thuật toán đề xuất giải (MOP( $F, U$ )) (2)

Từ đó ta thu được nghiệm tối ưu sau:

$$\lambda^{\mathbf{k}} = (\lambda_j^k), \beta^{\mathbf{k}} = (\beta_i^k), \gamma^{\mathbf{k}} = (\gamma_p^k),$$

$$s(x^k) = - \left( \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{i \in I_\epsilon(x^k)} \beta_i^k \nabla g_i(x^k) + \sum_{p \in I_\epsilon(x^k)} \gamma_p^k \nabla \mathcal{G}_p(x^k) \right)$$

$$\text{và } \Theta(x^k, \epsilon) = -\frac{1}{2} \|s(x^k)\|^2.$$

(b) Nếu  $\Theta(x^k, \epsilon) = 0$  thì **dừng lại**. Ngược lại, tiếp tục **Bước 3(c)**

(c) Cập nhật  $x^{k+1} := P_U(x^k + \alpha_k s(x^k))$ .

(d) Tính toán cỡ bước:

Nếu

$$F(x^{k+1}) \leq F(x^k) + \sigma \langle JF(x^k), x^{k+1} - x^k \rangle \quad (6)$$

thì  $\alpha_{k+1} = \alpha_k$ , ngược lại  $\alpha_{k+1} := \kappa \alpha_k$ .

**Bước 4.** Đặt  $k := k + 1$  và chuyển sang **Bước 3**.

# Sự hội tụ của Thuật toán 3

## Định lý 3.1

Xét  $JF$  và  $\nabla g_i (i \in I)$  liên tục Lipshitz và điểm bắt đầu  $x^1 \in U$ . Khi đó với  $\varepsilon \in \mathbb{R}_{++}$ , chuỗi  $\{x^k\}_k \subset U$  được thiết lập dựa vào giải bài toán  $(P_2)$  và cỡ bước được thiết lập dựa vào điều kiện Adaptive (6) thì ta có:

$$\lim_{k \rightarrow \infty} F_i(x^k) = -\infty$$

với mọi  $i$  hoặc:

$$\lim_{k \rightarrow \infty} \Theta(x^k, \varepsilon) = 0$$

## Ví dụ minh họa Thuật toán 3 (1)

Xét ví dụ quy hoạch đa mục tiêu lồi có ràng buộc với 3 hàm mục tiêu sau đây:

$$\text{Min } F(x) = \{f_1, f_2, f_3\}$$

$$\text{v.đ.k } x_1^2 + x_2^2 + x_3^2 = 1, \quad x_i \in [0, 1], i = 1, 2, 3$$

trong đó:

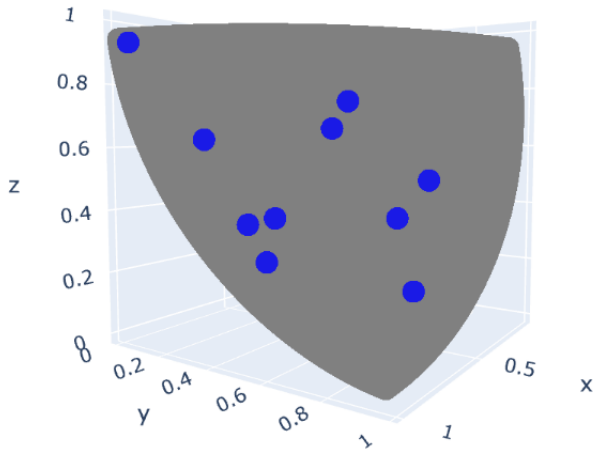
$$f_1 = \frac{x_1^2 + x_2^2 + x_3^2 + x_2 - 12x_3 + 12}{14},$$

$$f_2 = \frac{x_1^2 + x_2^2 + x_3^2 + 8x_1 - 44.8x_2 + 8x_3 + 44}{57},$$

$$f_3 = \frac{x_1^2 + x_2^2 + x_3^2 - 44.8x_1 + 8x_2 + 8x_3 + 43.7}{56}$$

## Ví dụ minh họa Thuật toán 3 (2)

Kết quả thử nghiệm thuật toán 3 với 10 điểm khởi tạo ngẫu nhiên cho kết quả:





# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc
- 4 Bài toán học đa nhiệm**
- 5 Kết luận

# Phương pháp đề xuất giải bài toán MTL (1)

- Tương tự Thuật toán PMTL, sử dụng  $K$  véc tơ  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$  để biểu thị cho độ ưu tiên giữa các tác vụ trong bài toán MTL.
- Sử dụng các ràng buộc bất đẳng thức tuyến tính  $\mathcal{G}_p(\theta_t) = \langle \mathbf{u}_p - \mathbf{u}_k, \mathcal{L}(\theta_t) \rangle \leq 0, \forall p = 1, \dots, K$  để phân tách không gian ảnh thành các bài toán con.
- Đề xuất sử dụng cỡ bước tự thích nghi để cập nhật cho *learning rate* của mạng Neural.

# Thuật toán đề xuất giải MTL

---

## Thuật toán 4 Thuật toán đề xuất

---

**Input:** Tập các véc tơ ưu tiên  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$ .

- 1: Thiết lập  $\kappa \in [0, 1], \sigma \in [0, 1], \alpha_1 \in (0, +\infty)$ .
- 2: **for**  $k = 1$  to  $K$  **do**
- 3:     Khởi tạo bộ tham số  $\theta_r^k$  cho mạng Neural.
- 4:     Tìm bộ tham số hữu hiệu  $\theta_0^k$  từ  $\theta_r^k$  sử dụng phương pháp hướng giảm.
- 5:     **for**  $t = 1$  to  $T$  **do**

# Thuật toán đề xuất giải MTL

6: Tìm hướng giảm thông qua việc giải bài toán sau:

$$\min_{\lambda_j, \gamma_p} \left\| \sum_{j=1}^m \lambda_j^k \nabla \mathcal{L}_j(\theta_t^k)^T + \sum_{p \in I(\theta_t^k)} \gamma_p^k \nabla \mathcal{G}_p(\theta_t^k) \right\|^2$$

$$\text{v.đ.k: } \lambda_j \geq 0, \gamma_p \geq 0, \sum_{j=1}^m \lambda_j^k + \sum_{p \in I(\theta_t^k)} \gamma_p^k = 1$$

7: Trong đó,  $I_\epsilon(\theta_t^k) := \{p \in 1, \dots, K | \mathcal{G}_p(\theta_t^k) \geq -\epsilon\}$ .

8: Từ đó xác định hướng giảm:

$$s(\theta_t^k) = - \left( \sum_{j=1}^m \lambda_j^k \nabla \mathcal{L}_j(\theta_t^k) + \sum_{p \in I(\theta_t^k)} \gamma_p^k \nabla \mathcal{G}_p(\theta_t^k) \right)$$

9: Đặt  $\theta_{t+1}^k := \theta_t^k + \alpha_t s(\theta_t^k)$ .

# Thuật toán đề xuất giải MTL

```
10:   Tính toán cỡ bước:  
11:   Nếu  $\mathcal{L}(\theta_{t+1}^k) \leq \mathcal{L}(\theta_t^k) + \sigma \langle s(\theta_t^k), \theta_{t+1}^k - \theta_t^k \rangle$   
12:   thì  $\alpha_{t+1} = \alpha_t$ , ngược lại đặt  $\alpha_{t+1} := \kappa \alpha_t$ .  
13:   Cập nhật bước lặp  $t := t + 1$ .  
14: end for  
15: end for
```

---

## Ví dụ bài toán MTL (1)

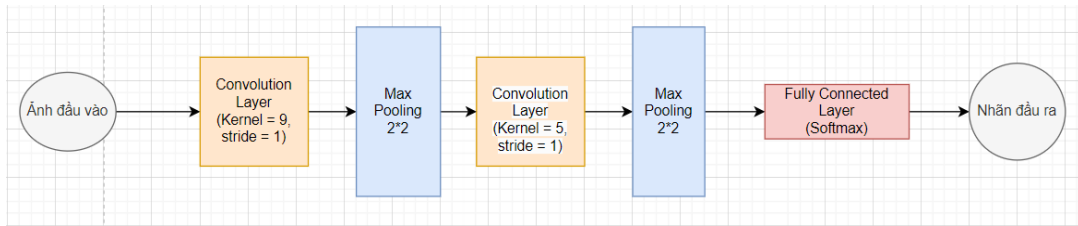
Multi-MNIST là bộ dữ liệu được xây dựng dựa trên việc ghép 2 chữ số trong bộ dữ liệu MNIST [6] vào 1 ảnh.



Hình: Bộ dữ liệu Multi-MNIST

## Ví dụ bài toán MTL (2)

- Tiến hành xây dựng model LeNet tương tự như trong bài [7].
- Tiến hành quá trình huấn luyện với 5 véc tơ ưu tiên  $[\cos(\frac{k\pi}{2K}), \sin(\frac{k\pi}{2K})]$ ,  $k = 0, 1, 2, 3, 4$  với mỗi lần 50 epoch.



## Ví dụ MTL cho CV (3)

Hai tác vụ của chúng ta là tác vụ phân loại ảnh nên hàm mục tiêu hay hàm mất mát được mô hình dưới dạng hàm Cross-entropy:

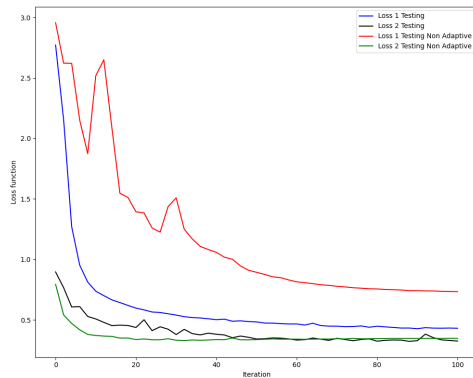
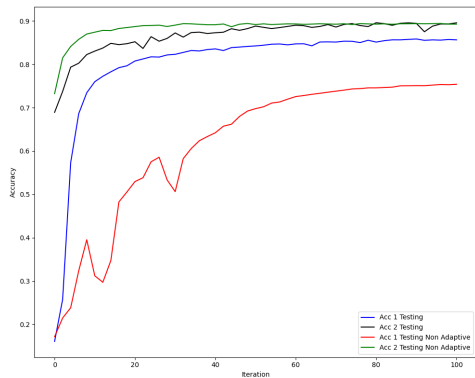
$$L_1(\hat{y}_1, y_1) = - \sum_k^K y_1^{(k)} \log \hat{y}_1^{(k)}$$
$$L_2(\hat{y}_2, y_2) = - \sum_k^K y_2^{(k)} \log \hat{y}_2^{(k)}$$

Trong đó  $y_i^{(k)}$ ,  $(i = 1, 2)$  có giá trị 0 hoặc 1 là chỉ số biểu thị nhãn  $k$  được phân loại đúng và  $\hat{y}_i^{(k)}$ ,  $(i = 1, 2)$  là giá trị dự đoán nhãn của mô hình.



## Ví dụ bài toán MTL (3)

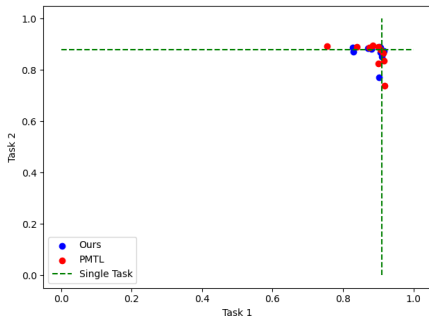
Hình ảnh so sánh quá trình huấn luyện của thuật toán đề xuất (đường màu xanh dương cho tác vụ 1 và đường màu đen cho tác vụ 2) và thuật toán PMTL (đường màu đỏ cho tác vụ 1 và đường màu xanh lá cho tác vụ 2) trong trường hợp véc tơ ưu tiên  $\mathbf{u}_1$  như sau:



Quá trình huấn luyện với cỡ bước Adaptive cho kết quả hàm mất mát giảm đều hơn và do vậy độ chính xác cũng tăng đều liên tục.

## Ví dụ bài toán MTL (3)

Bảng so sánh giá trị độ chính xác trên tập Test giữa thuật toán đề xuất và PMTL sẽ được minh họa dưới đây:



	Trung bình Accuracy Task 1	Trung bình Accuracy Task 2
PMTL	$0.87 \pm 0.01$	$0.85 \pm 0.01$
Đề xuất	<b><math>0.88 \pm 0.01</math></b>	<b><math>0.86 \pm 0.01</math></b>

**Bảng:** So sánh 2 Thuật toán 4 và PMTL khi chạy trên Multi-MNIST

**Hình:** So sánh Pareto Front tìm được bởi Thuật toán 4 và PMTL trên tập Multi-MNIST

## Ví dụ MTL cho NLP (1)

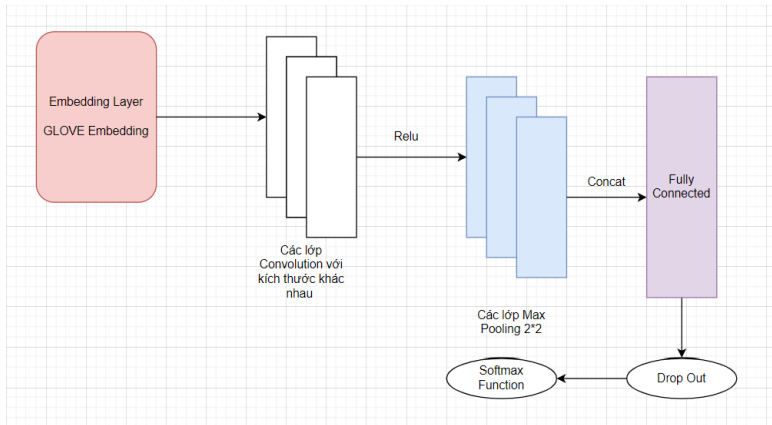
- Tiến hành kiểm thử độ hiệu quả của thuật toán đề xuất trong việc áp dụng cho bài toán MTL thuộc lĩnh vực NLP trên tập Drug review.
- Bộ dữ liệu gồm 111564 câu nhận xét.

	review	condition	rating
0	"It has no side effect, I take it in combinati...	Left Ventricular Dysfunction	9
1	"My son is halfway through his fourth week of ...	ADHD	8
2	"I used to take another oral contraceptive, wh...	Birth Control	5
3	"This is my first time using any form of birth...	Birth Control	8
4	"Suboxone has completely turned my life around...	Opiate Dependence	9

Hình: Minh họa tập dữ liệu Drug

# Ví dụ MTL cho NLP (2)

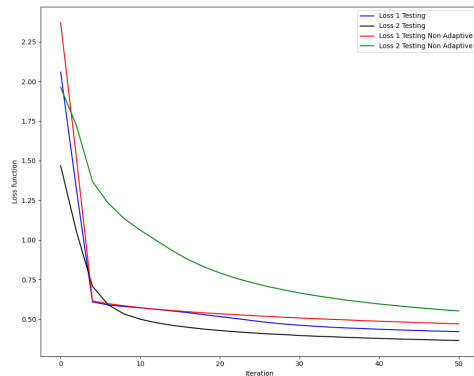
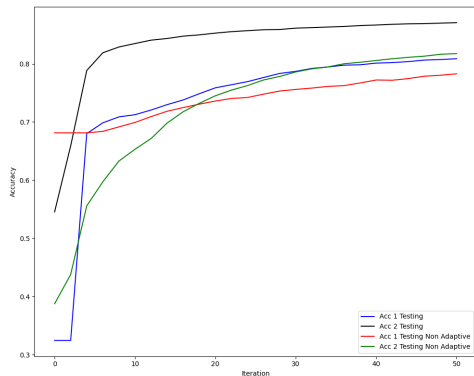
Chi tiết mô hình được xây dựng như sau:



**Hình:** Mô hình TextCNN xây dựng

# Ví dụ MTL về NLP (3)

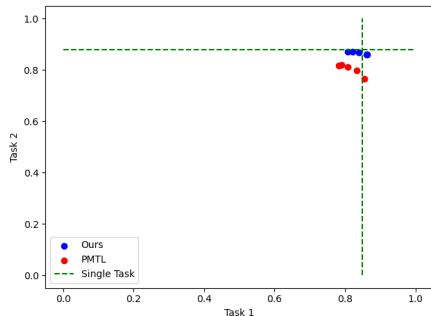
Tiến hành so sánh thuật toán 4 với thuật toán PMTL thu được kết quả như sau:



Kết quả cho thấy chỉ sau 10 epochs thì Accuracy của Thuật toán 4 đã cao hơn hẳn so với Thuật toán PMTL.

# Ví dụ MTL về NLP (4)

Tiến hành đo kết quả Accuracy trung bình 2 tác vụ của 2 Thuật toán PMTL và Thuật toán 4 sau 10 lần chạy khác nhau với cùng bộ thông số khởi tạo ta thu được kết quả dưới đây:



	Trung bình Accuracy Task 1	Trung bình Accuracy Task 2
PMTL	$0.82 \pm 0.01$	$0.83 \pm 0.01$
Đề xuất	<b><math>0.84 \pm 0.01</math></b>	<b><math>0.86 \pm 0.01</math></b>

**Bảng:** So sánh 2 Thuật toán 4 và PMTL khi chạy trên Drug Review

**Hình:** So sánh Pareto Front của PMTL và Thuật toán 4 trên tập Drug Review

# Nội dung chính

- 1 Giới thiệu
- 2 Bài toán quy hoạch đa mục tiêu không ràng buộc
- 3 Bài toán quy hoạch đa mục tiêu có ràng buộc
- 4 Bài toán học đa nhiệm
- 5 Kết luận**

# Đóng góp chính

## Kết quả đạt được

- Đề xuất 4 thuật toán mới bao gồm đầy đủ mô tả, sự hội tụ và tính toán thử nghiệm của thuật toán.
- Đề xuất cách thiết lập cỡ bước tự thích nghi cho phương pháp hướng giảm sâu nhất đa mục tiêu giải bài toán tối ưu phức tạp với hàm mục tiêu không lồi, bài toán có ràng buộc, tập ràng buộc không bị chặn.
- Thực nghiệm độ hiệu quả của thuật toán đề xuất giải bài toán MTL trong lĩnh vực CV và NLP cho kết quả tốt.
- Kết quả của Đồ án này được đóng gói thành bài báo nghiên cứu khoa học và nộp tại Tạp chí Computational Optimization and Applications (Q1).



# Hướng phát triển

## Các hướng phát triển

- Nghiên cứu các cách thiết lập cỡ bước khác nhau giúp nâng cao hiệu quả của thuật toán trong việc giải bài toán tối ưu đa mục tiêu.
- Áp dụng thuật toán đề xuất vào nhiều bài toán MTL khác như xử lý tiếng nói, dịch máy,...

***XIN TRÂN TRỌNG CẢM ƠN!***

## Tài liệu tham khảo

- [1] Alireza Chamanzar and Yao Nie. “Weakly supervised multi-task learning for cell detection and segmentation”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2020, pp. 513–516.
- [2] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).
- [3] Jörg Fliege and Benar Fux Svaiter. “Steepest descent methods for multicriteria optimization”. In: *Mathematical methods of operations research* 51 (2000), pp. 479–494.
- [4] Long P Hoang et al. “Improving pareto front learning via multi-sample hypernetworks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 7. 2023, pp. 7875–7883.
- [5] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7482–7491.

## Tài liệu tham khảo

- [6] Yann LeCun, Corinna Cortes, and CJ Burges. “MNIST handwritten digit database”. In: *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist> 2 (2010).
- [7] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [8] Xi Lin et al. “Pareto multi-task learning”. In: *Advances in neural information processing systems* 32 (2019).
- [9] Shikun Liu, Edward Johns, and Andrew J Davison. “End-to-end multi-task learning with attention”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 1871–1880.
- [10] Christian Von Lucken and Carlos Brizuela. “A survey on multi-objective evolutionary algorithms for many-objective problems”. In: *Computational optimization and applications* 58 (2014), pp. 707–756.

## Tài liệu tham khảo

- [11] Li Wang et al. “Learning deep features for multiple object tracking by using a multi-task learning strategy”. In: *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2014, pp. 838–842.