

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN NGHIÊN CỨU

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

Nguyễn Anh Minh

minh.na194117@sis.hust.edu.vn

Ngành Toán tin

Giảng viên hướng dẫn: TS. Trần Ngọc Thắng

Chữ kí của GVHD

Bộ môn: Toán Tin

Viện: Toán ứng dụng và Tin học

HÀ NỘI, 7/2023

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục tiêu và nội dung của đề án

(a) Mục tiêu:

(b) Nội dung:

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên:

Hà Nội, tháng ... năm 2023

Giảng viên hướng dẫn

TS. Trần Ngọc Thắng

Lời cảm ơn

Tôi cảm thấy rất tự hào khi được học tại Đại học Bách Khoa Hà Nội và còn tự hào hơn nữa khi có 4 năm tuổi trẻ học tập và phát triển bản thân tại Viện Toán ứng dụng và Tin học. Trong suốt quá trình đó, tôi vô cùng may mắn khi được học cùng với những người bạn trong tập thể CTTN Toán Tin K64 và LCD - LCH SAMI, được học trao đổi, theo đuổi đam mê khoa học dữ liệu, công nghệ thông tin, toán học cùng với các bạn là cơ hội có một không hai mà SAMI đã trao cho tôi.

Trong suốt quá trình hoàn thiện Đề án nghiên cứu này, tôi xin gửi lời cảm ơn chân thành tới TS. Trần Ngọc Thăng, người thầy đáng kính đã hướng dẫn, dìu dắt tôi trong suốt quá trình nghiên cứu khoa học.

Tôi xin gửi làm cảm ơn chân thành tới những người anh em Nguyễn Ngọc Bảo, Đỗ Mạnh Dũng, Nguyễn Việt Dũng, Nguyễn Đình Nhật, Bùi Tiến Thành và Lê Nguyên Bách là những người bạn đại học đã giúp đỡ tôi rất nhiều trong suốt 4 năm qua. Đồng thời, tôi xin cảm ơn những anh chị tiền bối trong Mảng Học tập và NCKH thuộc LCD - LCH SAMI như anh Nguyễn Quang Huy, anh Trần Anh Tuấn, anh Trương Tuấn Khang đã định hướng và giúp đỡ tôi rất nhiều trong quá trình học tập.

Tôi may mắn có cơ hội làm việc tại Trung tâm phân tích dữ liệu Viettel DAC từ tháng 10 năm 2022, nơi đây là cái nôi nuôi dưỡng năng lực Khoa học dữ liệu của tôi. Tôi xin chân thành cảm ơn anh chị em trong trung tâm DAC và đặc biệt là anh Đoàn Xuân Tâm, anh Đoàn Văn Huy, anh Hoàng Anh Dũng, chị Lê Hoàng Ngân, chị Hoàng Phương Linh và chị Nguyễn Thùy Linh đã là những người Mentor, người thầy của tôi trong suốt quá trình tôi làm việc.

Tôi xin gửi làm cảm ơn từ tận đáy lòng tới gia đình tôi những người đã luôn ủng hộ và dõi theo tôi trong mọi chặng đường học tập và làm việc. Cuối cùng, tôi xin dành lời cảm ơn tới người con gái Đinh Thị Vân Anh vì đã luôn đồng hành cùng tôi trong những năm tháng tại Bách Khoa Hà Nội, cảm ơn cậu vì là một phần thanh xuân Bách Khoa của tôi.

Hà Nội, tháng ... năm 2023

Tác giả đề án

Nguyễn Anh Minh

Tóm tắt nội dung Đồ án

Đồ án này sẽ trình bày các nội dung chính sau đây:

1. Trình bày các kiến thức cơ bản liên quan đến tối ưu phi tuyến đa mục tiêu và các hướng giảm trong bài toán quy hoạch đa mục tiêu.
2. Đề xuất thuật toán kết hợp hướng giảm sâu đa mục tiêu và cỡ bước tự thích nghi để giải bài toán quy hoạch đa mục tiêu có ràng buộc cũng như không ràng buộc. Đồng thời áp dụng thuật toán giải quyết bài toán Multi-task Learning.
3. Thực nghiệm với các ví dụ minh họa giải bài toán tối ưu đa mục tiêu có ràng buộc, không ràng buộc với các lớp hàm giả lồi, tựa lồi, khả vi. Đồng thời áp dụng thuật toán hướng giảm đề xuất vào giải 2 bài toán Multi-task learning sau:
 - Thị giác máy tính: Tối ưu đồng thời 2 tác vụ nhận diện ảnh trên bộ dữ liệu Multi-Mnist [3].
 - Xử lý ngôn ngữ tự nhiên: Tối ưu đồng thời 2 tác vụ phân loại đề mục và phân loại cảm xúc khách hàng trên bộ dữ liệu Drug Review [37].

Hà Nội, tháng ... năm 2023

Tác giả đồ án

Nguyễn Anh Minh

Mục lục

Bảng ký hiệu và chữ viết tắt	i
Danh sách bảng	iii
Danh sách hình vẽ	iv
Chương 1 Cơ sở lý thuyết	2
1.1 Một số định nghĩa cơ bản	2
1.2 Phát biểu bài toán	3
1.2.1 Bài toán quy hoạch đa mục tiêu	3
1.2.2 Bài toán học đa nhiệm MTL	5
1.3 Phương pháp hướng giảm đa mục tiêu	7
1.3.1 Thuật toán Multi-Gradient	7
1.3.2 Thuật toán hướng giảm PMTL	9
Chương 2 Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu không ràng buộc	14
2.1 Cơ sở lý thuyết của thuật toán	15
2.2 Thuật toán đề xuất	18
2.3 Sự hội tụ của thuật toán đề xuất	19
2.4 Thuật toán đề xuất cải tiến với các véc tơ ưu tiên	22
2.5 Tính toán thử nghiệm và đánh giá kết quả	24
Chương 3 Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu có ràng buộc	30
3.1 Cơ sở lý thuyết của thuật toán	30
3.2 Thuật toán đề xuất	33

3.3	Sự hội tụ của thuật toán đề xuất	33
3.4	Thuật toán đề xuất cải tiến với các véc tơ ưu tiên	35
3.5	Tính toán thử nghiệm và đánh giá kết quả	37
Chương 4 Ứng dụng vào bài toán học đa nhiệm		41
4.1	Thuật toán đề xuất giải bài toán MTL	41
4.2	Bài toán MTL cho thị giác máy tính	42
4.3	Bài toán MTL cho xử lý ngôn ngữ tự nhiên	46
Kết luận		51
Tài liệu tham khảo		52

Bảng ký hiệu và chữ viết tắt

Ký hiệu	Ý nghĩa
\mathbb{R}^m	không gian véc tơ m chiều
$y^1 < y^2$	$y_k^1 < y_k^2$ với $k = 1, \dots, m$
$y^1 \leq y^2$	$y_k^1 \leq y_k^2$ với $k = 1, \dots, m$ nhưng $y^1 \neq y^2$
\mathbb{R}_+^m	$\{y \in \mathbb{R}^m : y \geq 0\}$
\mathbb{R}_{++}^m	$\{y \in \mathbb{R}^m : y > 0\}$
\mathbb{R}_+	$\{y \in \mathbb{R} : y \geq 0\}$
\mathbb{R}_{++}	$\{y \in \mathbb{R} : y > 0\}$

Chữ viết tắt	Ý nghĩa
CV	Computer Vision
NLP	Natural Language Processing
MTL	Multi-task Learning
PMTL	Thuật toán Pareto Multi-task Learning
MGDA	Multi Gradient Descent Algorithm
HV	Hyper Volume
MOP	Multi-objective Optimization Problem

Danh sách bảng

2.1	So sánh Thuật toán 4 và thuật toán MGDA giải Ví dụ 2.1	25
2.2	So sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.1	26
2.3	Kết quả so sánh Thuật toán 4 và thuật toán PMTL trong các trường hợp khác nhau.	26
2.4	Kết quả so sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.2 trong các trường hợp khác nhau.	27
2.5	Kết quả so sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.2 trong các trường hợp khác nhau.	28
4.1	So sánh 2 Thuật toán 7 và PMTL khi chạy trên Multi-MNIST . .	46
4.2	So sánh 2 Thuật toán 7 và PMTL khi chạy trên Drug Review . .	50

Danh sách hình vẽ

1.1	Mình họa Multi-task Learning	6
1.2	Phân rã bài toán tối ưu đa mục tiêu trong Thuật toán PMTL	9
2.1	Kết quả Thuật toán 3 giải Ví dụ 2.1	25
2.2	Kết quả Thuật toán 4 giải Ví dụ 2.1	25
2.3	Kết quả Thuật toán 3 giải Ví dụ 2.2	27
2.4	Kết quả Thuật toán 4 giải Ví dụ 2.2	27
2.5	Kết quả Thuật toán 4 giải Ví dụ 2.3	29
3.1	Kết quả Thuật toán 6 giải ví dụ 3.1	38
3.2	Kết quả Thuật toán 5 giải Ví dụ 3.2	38
3.3	Kết quả Thuật toán 6 giải Ví dụ 3.2	38
3.4	Kết quả Thuật toán 6 giải Ví dụ 3.3	39
3.5	Kết quả giải Ví dụ 3.4 của Thuật toán 6	40
4.1	Mình họa Multi-MNIST	43
4.2	Kiến trúc mạng LeNet sử dụng trong Đồ án	43
4.3	Quá trình huấn luyện Thuật toán 7 trên tập Multi-MNIST	44
4.4	So sánh Thuật toán 7 và PMTL trên tập Multi-MNIST	45
4.5	So sánh Pareto Front tìm được bởi Thuật toán 7 và PMTL trên tập Multi-MNIST	46
4.6	Mình họa tập dữ liệu Drug	46
4.7	Mô hình TextCNN xây dựng	47
4.8	Quá trình huấn luyện Thuật toán 7 trên tập Drug Review	48
4.9	So sánh 2 Thuật toán 7 và PMTL khi chạy trên Drug Review	49
4.10	So sánh Pareto Front của PMTL và Thuật toán 7 trên tập Drug Review	50

Mở đầu

Đồ án này sẽ được trình bày gồm 3 chương chính như sau:

- **Chương 1: Cơ sở lý thuyết** sẽ trình bày các kiến thức cơ bản liên quan đến tối ưu phi tuyến và hướng giảm trong bài toán quy hoạch đa mục tiêu.
- **Chương 2: Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu không ràng buộc** sẽ trình bày về cơ sở lý thuyết và sự hội tụ của thuật toán hướng giảm đề xuất để giải bài toán quy hoạch đa mục tiêu không ràng buộc. Đồng thời tiến hành giải số minh họa các ví dụ quy hoạch đa mục tiêu không ràng buộc.
- **Chương 3: Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu có ràng buộc** sẽ gồm các nội dung tương ứng như **Chương 2** khi tiến hành đề xuất thuật toán giải quy hoạch đa mục tiêu có ràng buộc.
- **Chương 4: Ứng dụng vào bài toán học đa nhiệm** sẽ giới thiệu tổng quan về bài toán MTL và hướng tiếp cận xây dựng mô hình giải MTL. Đồng thời, áp dụng thuật toán đề xuất giải 2 bài toán cụ thể: phân loại ảnh thuộc lĩnh vực CV và phân loại cảm xúc khách hàng thuộc lĩnh vực NLP.

Chương 1

Cơ sở lý thuyết

1.1 Một số định nghĩa cơ bản

Trong mục này, chúng tôi sẽ quy ước một số kí hiệu dùng trong đề án, đồng thời giới thiệu một số kiến thức cơ bản về tối ưu đa mục tiêu và định nghĩa hướng giảm trong tối ưu đa mục tiêu.

Định nghĩa 1.1 [20] Một hàm khả vi $f : \mathbb{R}^n \rightarrow \mathbb{R}$ được gọi là:

i) lồi trên U nếu với mọi $x, y \in U$, $\lambda \in [0, 1]$ có

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

ii) giả lồi trên U nếu với mỗi $x, y \in U$ có

$$\langle \nabla f(x), y - x \rangle \geq 0 \Rightarrow f(y) \geq f(x).$$

iii) tựa lồi trên U nếu với mỗi $x, y \in U$, $\lambda \in [0; 1]$ có

$$f(\lambda x + (1 - \lambda)y) \leq \max \{f(x); f(y)\}.$$

Bổ đề 1.1 [9] Một hàm khả vi f được gọi là hàm tựa lồi trên U khi và chỉ khi

$$f(y) \leq f(x) \Rightarrow \langle \nabla f(x), y - x \rangle \leq 0.$$

Tính chất cơ bản: Nếu f là lồi $\Rightarrow f$ giả lồi $\Rightarrow f$ tựa lồi ([20]).

Ảnh của ma trận $M \in \mathbb{R}^{m \times n}$ sẽ được ký hiệu bởi $R(M)$ và $I \in \mathbb{R}^{n \times n}$ ký hiệu cho ma trận đơn vị. Cho hai ma trận $A, B \in \mathbb{R}^{n \times n}$, $B \leq A$ ($B < A$) nghĩa là $A - B$ xác định dương.

Trong phần tiếp theo, ký hiệu chuẩn Euclidean trong \mathbb{R}^n sẽ được ký hiệu bằng $\|\cdot\|$ và $B[x, r]$ ký hiệu cho quả cầu đóng bán kính r với tâm $x \in \mathbb{R}^n$. Chúng ta sẽ dùng ký hiệu $\|\cdot\|$ tương ứng cho chuẩn của không gian ma trận.

Ký hiệu $A \in \mathbb{R}^{m \times n}$

$$\|A\|_{\infty,2} := \max_{x \neq 0} \frac{\|Ax\|_{\infty}}{\|x\|}.$$

Khi đó $\|\cdot\|_{\infty,2}$ là chuẩn trong không gian $\mathbb{R}^{m \times n}$. Dễ dàng suy ra

$$\begin{aligned} \|A\|_{\infty,2} &:= \max_{i=1,\dots,m} \|A_{i,\cdot}\| \\ &= \max_{i=1,\dots,m} \left(\sum_{j=1}^n A_{i,j}^2 \right)^{1/2}. \end{aligned}$$

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

$$P_U(x) := \operatorname{argmin}\{\|z - x\| : z \in U\}. \quad (1.1)$$

1.2 Phát biểu bài toán

1.2.1 Bài toán quy hoạch đa mục tiêu

Ta xét bài toán tối ưu đa mục tiêu sau:

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

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 . Ta giả sử $U = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, \dots, l\}$. Ta ký hiệu ma trận Jacobian của F tại x có dạng sau:

$$JF(x) = \begin{bmatrix} \nabla F_1(x)^T \\ \vdots \\ \nabla F_m(x)^T \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Trong đồ án này, ta xét ma trận Jacobian JF của F là liên tục Lipschitz, nghĩa là các đạo hàm $\nabla F_j \in \mathbb{R}^n$ của hàm thành phần F_j là liên tục Lipschitz với mọi $j = 1, \dots, m$. Giả sử tập nghiệm của $(\text{MOP}(F, U))$ khác rỗng. Các định nghĩa dưới đây được trích ra từ [2].

Định nghĩa 1.2 Một điểm $x^* \in U$ được gọi là nghiệm hữu hiệu (efficient point) hay một điểm tối ưu Pareto của bài toán $(\text{MOP}(F, U))$ nếu không tồn tại $y \in U, F(y) \leq F(x^*)$ và $F(y) \neq F(x^*)$, trong đó dấu \leq giữa các véc tơ được hiểu theo nghĩa quan hệ giữa từng hàm thành phần. Tức là, không tồn tại $y \in U : F_i(y) \leq F_i(x^*), \forall i = 1, \dots, m$ và tồn tại $i_0 \in \{1, \dots, m\} : F_{i_0}(y) < F_{i_0}(x^*)$. Tập các điểm tối ưu Pareto tạo thành Pareto Front.

Xét $\mathbb{R}_+^m := \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$, khi đó thứ tự từng phần (partial order) giữa giá trị hàm véc tơ được định nghĩa:

$$F(y) \leq F(z) \iff F(z) - F(y) \in \mathbb{R}_+^m,$$

Do đó, để giải bài toán $(MOP(F, U))$, ta cần tìm điểm cực tiểu theo thứ tự từng phần.

Định nghĩa 1.3 Một điểm $x^* \in U$ được gọi là nghiệm hữu hiệu yếu (weakly efficient) hay điểm tối ưu Pareto yếu của bài toán $(MOP(F, U))$ nếu không tồn tại $y \in U$ sao cho $F(y) < F(x^*)$, trong đó bất đẳng thức $F(y) < F(x^*)$ cũng được hiểu theo nghĩa thứ tự từng phần.

Xét $\mathbb{R}_{++}^m := \mathbb{R}_{++} \times \cdots \times \mathbb{R}_{++}$ đồng thời định nghĩa

$$-\mathbb{R}_{++}^m = \{-v : v \in \mathbb{R}_{++}^m\}.$$

Định nghĩa 1.4 Một điểm $x^* \in U$ được gọi là điểm hữu hiệu địa phương (locally efficient) (tương ứng hữu hiệu địa phương yếu) của bài toán $(MOP(F, U))$ nếu tồn tại một vùng lân cận $V \subseteq U$ của x^* sao cho điểm x^* là nghiệm hữu hiệu (tương ứng hữu hiệu yếu) của bài toán $(MOP(F, U))$ giới hạn trên V .

Khi U lồi và F là hàm \mathbb{R}_+^m -lồi (tức các hàm thành phần của F lồi) thì mỗi điểm tối ưu Pareto địa phương là điểm tối ưu Pareto toàn cục. Đồng thời, mỗi điểm hữu hiệu địa phương cũng là điểm hữu hiệu địa phương yếu.

Định nghĩa 1.5 [8] Cho $z \in U$ một véc tơ $v \in \mathbb{R}^n$ được gọi là hướng tiếp xúc của U tại z nếu tồn tại một chuỗi $(z^k)_k \subseteq U$ và một giá trị vô hướng $\lambda \in \mathbb{R}_+$ sao cho

$$\lim_{k \rightarrow \infty} z^k = z \quad \text{và} \quad \lim_{k \rightarrow \infty} \lambda \frac{z^k - z}{\|z^k - z\|} = v$$

Tập tất cả các hướng tiếp xúc U tại z được gọi là nón tiếp xúc của U tại z và sẽ được ký hiệu bởi $T(U, z)$.

Với tập U là lồi đóng thì $T(U, z) := U(z) := \{s \in \mathbb{R}^n | s = u - z \text{ với } u \in U\}$.

Định nghĩa 1.6 Ta định nghĩa một điểm $x^* \in U$ là một điểm dừng Pareto (Pareto stationary point) (hoặc Pareto critical point) của hàm véc tơ F trên U nếu

$$JF(T(U, x^*)) \cap (-\mathbb{R}_{++}^m) = \emptyset, \quad (1.2)$$

trong đó $JF(x^*)$ là ma trận Jacobian của F tại x^* .

Định nghĩa này là điều kiện cần (nhưng không phải là điều kiện đủ) cho một điểm là điểm hữu hiệu Pareto và được lần đầu sử dụng tại [8] để xác định thuật toán hướng giảm sâu nhất đa mục tiêu. Trong trường hợp $m = 1$, (1.2) trở thành bài toán tối ưu truyền thống trong tối ưu vô hướng (tối ưu đơn mục tiêu).

Bổ đề 1.2 *Khi hàm F giả lồi (tức các hàm thành phần F_i , $i = 1, \dots, m$ giả lồi) thì (1.2) là điều kiện cần và đủ cho một điểm là điểm hữu hiệu yếu.*

Chứng minh: Thật vậy giả sử ngược lại, tồn tại điểm $y \in U$ sao cho $F(y) < F(x)$. Vì F_i là giả lồi với mỗi $i = 1, \dots, m$ suy ra $\langle \nabla F_i(x), (y - x) \rangle < 0$ và do đó $JF(x)(y - x) \in -\mathbb{R}_{++}^m$ mâu thuẫn với (1.2) (chú ý ta đang xét tập U lồi, đóng).

Việc chứng minh tính hội tụ của thuật toán đề xuất ta dựa trên sự hội tụ quasi-Fejér. Một chuỗi $\{z^k\} \subset \mathbb{R}^n$ được gọi là hội tụ quasi-Fejér tới tập U , $U \neq \emptyset$, khi và chỉ khi với mỗi $z \in U$, tồn tại $\{\varepsilon_k\} \subset \mathbb{R}_+$ sao cho

$\sum_{k=1}^{+\infty} \varepsilon_k < +\infty$ và

$$\|z^{k+1} - z\|^2 \leq \|z^k - z\|^2 + \varepsilon_k.$$

Bổ đề 1.3 *Nếu $\{z^k\} \subset \mathbb{R}^n$ hội tụ quasi-Fejér đến tập $U \neq \emptyset$ thì:*

- *Chuỗi $\{z^k\}$ bị chặn.*
- *Nếu điểm hội tụ của $\{z^k\}$ thuộc U , thì $\{z^k\}$ hội tụ tới $z \in U$.*

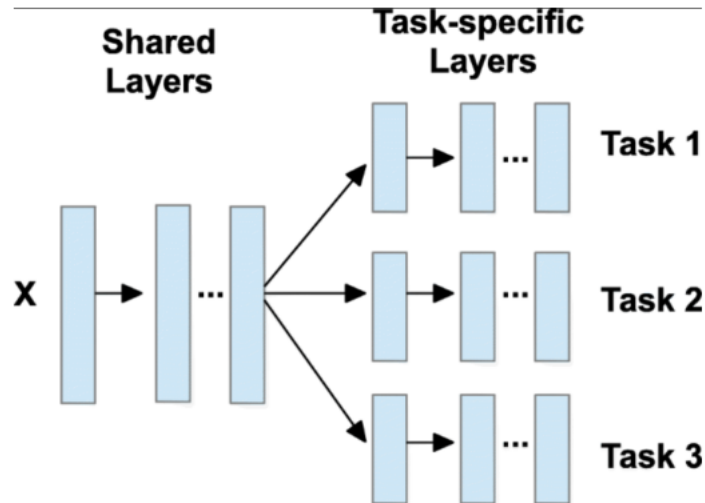
1.2.2 Bài toán học đa nhiệm MTL

Các phương pháp học máy truyền thống thường tập trung giải quyết một nhiệm vụ với một mô hình duy nhất. Điều này có thể khiến chúng ta bỏ qua những thông tin có thể giúp chúng ta thực hiện tốt hơn trên nhiệm vụ chúng ta quan tâm, đến từ những nhiệm vụ khác có liên quan tới nó. Lấy một ví dụ đơn giản khi bạn muốn dự đoán giá của một căn nhà dựa vào một số đặc trưng như diện tích, số lượng phòng, số tầng, có gần với các trung tâm thương mại nào hay không, . . . thì rõ ràng việc thêm vào một vài nhiệm vụ như dự đoán căn nhà này thuộc vùng nội thành hay ngoại thành hoặc là biệt thự hay là chung cư sẽ đem lại chúng ta rất nhiều thông tin cho việc dự đoán giá của nó.

Xét về góc độ chuyên ngành, Multi Task Learning (học đa tác vụ) thường được thực hiện thông qua việc chia sẻ các tham số chung (shared parameters) của các kiến trúc mạng chung và các tham số riêng (specific parameters) của

kiến trúc mạng từng tác vụ cụ thể, nhờ vậy mà chúng ta có thể thu được một mô hình có khả năng tổng quát hơn trên nhiệm vụ ban đầu mà chúng ta quan tâm.

Trong phạm vi nghiên cứu của đề án, chúng tôi tập trung vào việc tối ưu các tham số dùng chung (shared parameters) dựa trên các phương pháp hướng giảm đa mục tiêu. Sau các sự thành công của ứng dụng Multi-Task Learning vào thị giác máy tính của (Bilen và Vedaldi [4] năm 2016; Misra và cộng sự [6] năm 2016; Kokkinos [7] năm 2017; Zamir và cộng sự [8] năm 2018), trong ngôn ngữ tự nhiên (Collobert và Weston [9] năm 2008; Dong và cộng sự [10] năm 2015; Liu và cộng sự [13] năm 2015a; Luong và cộng sự [14] năm 2015; Hashimoto và cộng sự [15] năm 2017), và trong xử lý âm thanh (Huang và cộng sự [16] năm 2013; Seltzer và Droppo [17] năm 2013; Huang và cộng sự [18] năm 2015). Multi Task Learning còn được biết đến với một số tên gọi như Joint Learning, Learning to Learn, Learning with Auxiliary Task, mỗi khi chúng ta làm việc với một bài toán tối ưu có nhiều hơn một hàm mất mát, chúng ta ngầm định rằng chúng ta đang giải quyết một bài toán liên quan tới Multi Task Learning.



Hình 1.1: Minh họa Multi-task Learning

Mô hình hóa bài toán MTL dưới dạng MOP

Trong phần này, chúng tôi sẽ mô hình hóa bài toán MTL dưới dạng bài toán MOP dựa trên ý tưởng của Sener et.al [17]. Bài toán MTL bao gồm việc

tiến hành 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 θ cho các tác vụ. Các thuật toán giải MTL sẽ đi tối ưu hóa đồng thời tất cả các tác vụ. Bài toán (MTL) là một bài toán tối ưu đa mục tiêu, không một nghiệm tối ưu hay bộ tham số tối ưu nào có thể tối ưu đồng thời toàn bộ các tác vụ. Do đó, ta chỉ có thể đưa ra các nghiệm tối ưu Pareto biểu thị cho sự đánh đổi độ ưu tiên giữa các tác vụ.

1.3 Phương pháp hướng giảm đa mục tiêu

Phương pháp hướng giảm là công cụ thường được sử dụng trong việc giải các bài toán quy hoạch đa mục tiêu cũng như đơn mục tiêu với các lớp hàm phi tuyến từ lồi đến không lồi và có rất nhiều ứng dụng thực tế ([4], [7]). Tại mỗi bước lặp, phương pháp hướng giảm sẽ xây dựng một dãy điểm sao cho giá trị hàm mục tiêu giảm dần và dãy điểm đó hội tụ đến điểm dừng của thuật toán. Trong phần này, chúng tôi sẽ trình bày về hướng giảm sâu nhất giải bài toán quy hoạch đa mục tiêu trong trường hợp tổng quát có ràng buộc (với trường hợp không ràng buộc ta xét $U = \mathbb{R}^n$). Chúng tôi sẽ trình bày kiến thức về hướng giảm sâu nhất đa mục tiêu dựa trên bài báo của Fliege et.al [8] và Desideri et.al [18].

1.3.1 Thuật toán Multi-Gradient

Trong phần này, chúng tôi sẽ giới thiệu về thuật toán Multi-Gradient để giải bài toán quy hoạch đa mục tiêu ($\text{MOP}(F, U)$) với $U = \mathbb{R}^n$ được đề xuất tại [18]. Thuật toán này dựa trên định nghĩa của điểm dừng Pareto thông qua bao lồi của các véc tơ gradient thành phần.

Định nghĩa 1.7 Xét x^0 là tâm của một quả cầu mở trong tập U sao cho m hàm thành phần $F_i(x)$ ($1 \leq i \leq m$) trơn. Đặt $u_i = \nabla F_i(x^0)$ là các véc tơ gradient của hàm thành phần. Điểm x^0 được gọi là điểm dừng Pareto nếu tồn tại một tổ hợp lồi của các véc tơ gradient $\{u_i\}$ sao cho tổ hợp lồi đó bằng 0:

$$\sum_{i=1}^m \alpha_i u_i = 0, \quad \alpha_i \geq 0 \quad (\forall i), \quad \sum_{i=1}^m \alpha_i = 1$$

Bổ đề 1.4 [18] Nếu điểm x^0 là nghiệm tối ưu Pareto thì nó cũng là điểm dừng Pareto.

Nếu tại điểm không phải là điểm dừng x_0 thì tồn tại một hướng giảm cho toàn bộ các hàm thành phần $F_i(x_0), i = 1, \dots, m$. Việc xác định hướng giảm đó tương đương với tìm véc tơ $s \in U(x_0)$ sao cho: $\langle \nabla F_i(x^0), s \rangle \geq 0 (\forall i = 1, \dots, m)$. Khi đó, $-s$ là hướng giảm cần tìm.

Khi đó việc giải bài toán quay trở về tìm nghiệm trong bao lồi của họ véc tơ $\{u_i\} (i = 1, \dots, m)$:

$$\bar{U} = \left\{ u \in U(x) \mid u = \sum_{i=1}^m \alpha_i u_i; \alpha_i \geq 0 (\forall i = 1, \dots, m); \sum_{i=1}^m \alpha_i = 1 \right\}$$

Kết quả này sẽ được dùng để xây dựng Thuật toán MGDA sẽ được giới thiệu trong phần dưới đây đồng thời nghiệm của Thuật toán MGDA tìm được sẽ được chứng minh là điểm dừng Pareto.

Bổ đề 1.5 [18] Tồn tại duy nhất một nghiệm $\omega \in \bar{U}$ sao cho $\omega = \text{Argmin}_{u \in \bar{U}} \|u\|$

Bổ đề 1.6 [18] Với mọi $u \in U$: $\langle u, \omega \rangle \geq \|\omega\|^2$.

Định lý 1.1 Khảo sát tính chất dừng của điểm x_0 , ta thu được hai tính chất sau:

- Nếu $\omega = 0$ thì điểm x^0 là điểm dừng Pareto.
- Nếu $\omega \neq 0$ thì tại $x = x^0$ véc tơ $-\omega$ xác định một hướng giảm cho toàn bộ hàm thành phần.

Từ các Định lý và Bổ đề trên, ta có Thuật toán MGDA được đề xuất tại [18] như sau:

Algorithm 1 Thuật toán MGDA

Bước 1. Tính toán tổ hợp lồi ω

- Xác định véc tơ gradient có chuẩn hóa bởi đại lượng S_i như sau $u_i = \nabla F_i(x^0) / S_i$ và tìm phần tử có chuẩn cực tiểu ω trong bao lồi \bar{U} .
- Nếu $\omega = 0$ (hoặc đủ nhỏ) thì dừng lại, ngược lại chuyển qua Bước 2.

Bước 2. Tính toán cỡ bước

Xác định cỡ bước h là số thực dương lớn nhất thỏa mãn mọi hàm $f_i(t) = F_i(x^0 - t\omega)$ ($1 \leq i \leq m$) đơn điệu giảm trên $[0, h]$.

Bước 3. Cập nhật

Đặt $x^0 = x^0 - h\omega$ và quay trở lại Bước 1.

Cỡ bước t_k được xác định trong thuật toán trên dựa vào quy tắc Armijo. Ngoài ra, ta có thể có một số cách thiết lập cỡ bước khác như:

- Cỡ bước cố định: $t_k = h > 0$.
- Cỡ bước giảm theo bước lặp: $t_k = \frac{h}{\sqrt{k+1}}$.
- Thủ tục tìm chính xác theo tia: $t_k = \arg \min_{h \geq 0} F(x^k + hs(x^k))$.

1.3.2 Thuật toán hướng giảm PMTL

Trong phần này, chúng tôi sẽ giới thiệu Thuật toán PMTL được đề xuất bởi [3] để giải bài toán (MTL). Ý tưởng chính của thuật toán dựa trên việc phân rã không gian ảnh thành các bài toán con đồng thời giải bài toán tối ưu đa mục tiêu kết hợp các ràng buộc bất đẳng thức tuyến tính để các nghiệm tìm được trải đều trên Pareto Front.

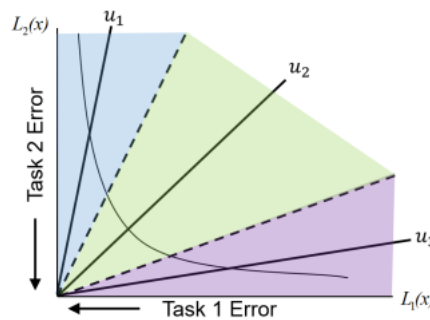
Phân rã bài toán Multi-task Learning

Xuất phát từ ý tưởng của [33], ta phân rã bài toán (MTL) thành K bài toán con tương ứng với tập véc tơ ưu tiên trải đều $\{u_1, u_2, \dots, u_K\} \in \mathbb{R}_+^m$. Giả sử các hàm thành phần đều dương, bài toán con tương ứng với véc tơ ưu tiên u_k được định nghĩa:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \dots, \mathcal{L}_m(\theta))^T, \text{ v.đ.k } \mathcal{L}(\theta) \in \Omega_k, \quad (1.3)$$

trong đó $\Omega_k (k = 1, \dots, K)$ là một vùng trong không gian ảnh:

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



Hình 1.2: Phân rã bài toán tối ưu đa mục tiêu trong Thuật toán PMTL

Nghĩa là, $v \in \Omega_k$ khi và chỉ khi v hợp với u_k một góc nhỏ nhất so với K véc tơ ưu tiên còn lại. Bài toán (1.3) tương đương:

$$\begin{aligned} \min_{\theta} \mathcal{L}(\theta) &= (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \dots, \mathcal{L}_m(\theta))^T \\ \text{v.đ.k } \mathcal{G}_p(\theta_t) &= \langle u_p - u_k, \mathcal{L}(\theta_t) \rangle \leq 0, \forall p = 1, \dots, K, \end{aligned}$$

Như đã chỉ ra trong Hình 1.2 thì các véc tơ ưu tiên chia không gian ảnh thành các vùng, mỗi vùng sẽ thu được một nghiệm tối ưu Pareto.

Tìm điểm khởi tạo

Để giải bài toán (1.3), ta cần tìm một điểm khởi tạo hữu hiệu, thỏa mãn mọi ràng buộc. Cho trước nghiệm ngẫu nhiên θ_r , cách làm đơn giản nhất là tìm θ_0 thỏa mãn:

$$\min_{\theta_0} \|\theta_0 - \theta_r\|^2 \quad \text{v.đ.k } \mathcal{L}(\theta_0) \in \Omega_k.$$

Tuy nhiên điều này dẫn tới việc giải bài toán n chiều có ràng buộc [3]. Do vậy, ta thực hiện cách làm hiệu quả hơn như sau.

Với nghiệm θ_r , ta xác định tập các ràng buộc chặt $I(\theta_r) = \{p \mid \mathcal{G}_p(\theta_r) \geq 0, p = 1, \dots, K\}$. Ta có thể tìm một hướng giảm s_r như sau:

$$(s_r, \alpha_r) = \arg \min_{s \in R^n, \alpha \in R} \alpha + \frac{1}{2} \|s\|^2, \text{ v.đ.k } \nabla \mathcal{G}_p(\theta_r)^T s \leq \alpha, p \in I(\theta_r).$$

và tiến hành cập nhật $\theta_{r_{t+1}} = \theta_{r_t} + \eta_r s_{r_t}$ cho đến khi tìm được một nghiệm chấp nhận được.

Giải các bài toán con

Khi đã có điểm khởi tạo ta sẽ dùng phương pháp gradient để giải các bài toán con. θ^* là nghiệm tối ưu Pareto cho $\mathcal{L}(\theta)$ trên Ω_k nếu $\theta^* \in \Omega_k$ và không tồn tại $\hat{\theta} \in \Omega_k$ sao cho $\hat{\theta} \leq \theta^*$.

Dựa trên [8], ta xác định hướng giảm cho bài toán MOP có ràng buộc tuyến tính dựa trên việc giải bài toán sau:

$$\begin{aligned} (s_t, \alpha_t) &= \arg \min_{d \in R^n, \alpha \in R} \alpha + \frac{1}{2} \|s\|^2 \\ \text{v.đ.k } \nabla \mathcal{L}_i(\theta_t)^T s &\leq \alpha, i = 1, \dots, m. \\ \nabla \mathcal{G}_p(\theta_t)^T s &\leq \alpha, p \in I_\varepsilon(\theta_t), \end{aligned} \tag{1.4}$$

Tổng quát hóa công thức hướng giảm

Bằng việc giải bài toán tối ưu có ràng buộc (1.4), ta sẽ xác định được một hướng giảm đa mục tiêu. Tuy nhiên việc giải bài toán (1.4) là không hiệu quả trong trường hợp không gian có số chiều lớn, nhất là trong bài toán MTL. Do đó ta sẽ viết lại bài toán (1.4) dưới dạng đối ngẫu. Dựa trên điều kiện KKT, ta có:

$$s_t = - \left(\sum_{i=1}^m \lambda_i \nabla \mathcal{L}_i(\theta_t) + \sum_{p \in I_e(\theta)} \gamma_p \nabla \mathcal{G}_p(\theta_t) \right), \quad \sum_{i=1}^m \lambda_i + \sum_{p \in I_e(\theta)} \gamma_p = 1, \quad (1.5)$$

với $\lambda_i \geq 0$ và $\beta_i \geq 0$ là các nhân tử Lagrange. Khi đó, ta giải bài toán đối ngẫu tương đương:

$$\begin{aligned} \max_{\lambda_p, \beta_i} & -\frac{1}{2} \left\| \sum_{i=1}^m \lambda_i \nabla \mathcal{L}_i(\theta_t) + \sum_{p \in I_e(\theta)} \gamma_p \nabla \mathcal{G}_p(\theta_t) \right\|^2 \\ \text{v.đ.k} & \sum_{i=1}^m \lambda_i + \sum_{p \in I_e(\theta)} \gamma_p = 1, \lambda_i \geq 0, \gamma_p \geq 0, \forall i = 1, \dots, m, \forall p \in I_e(\theta). \end{aligned} \quad (1.6)$$

Dựa trên việc giải bài toán đối ngẫu (1.6) ta có thể tìm được một hướng giảm hiệu quả và có công thức tại (1.5). Việc giải các bài toán con có thể diễn ra đồng thời dựa trên các véc tơ ưu tiên. Ví dụ với bài toán MTL 2 mục tiêu ta có thể sử dụng $K + 1$ véc tơ ưu tiên $\left\{ \left(\cos\left(\frac{k\pi}{2K}\right), \sin\left(\frac{k\pi}{2K}\right) \right) \mid k = 0, 1, \dots, K \right\}$ để phân chia không gian ảnh thành các bài toán con khác nhau. Do đó ta có Thuật toán PMTL như sau:

Algorithm 2 Thuật toán PMTL

Input. Tập K véc tơ ưu tiên trái đều $\mathbf{u}_1, \dots, \mathbf{u}_K$.

- 1: **Bước 2.** Cập nhật (Tiến hành giải song song cho từng bài toán con)
- 2: **for** $k = 1$ to K **do**
- 3: Khởi tạo bộ tham số ngẫu nhiên θ_r^k .
- 4: Tìm tham số hữu hiệu θ_0^k từ θ_r^k sử dụng phương pháp hướng giảm.
- 5: **for** $t=1$ to T **do**
- 6: Xác định bộ tham số λ^k, β^k dựa vào việc giải bài toán (1.6) từ đó xây dựng công thức hướng giảm như công thức (1.5)
- 7: Cập nhật tham số $\theta_{t=1}^k = \theta_t^k + \eta s_t^k$.
- 8: **end for**
- 9: **end for**

Output: Bộ tham số tối ưu ứng với K véc tơ ưu tiên.

Nhận xét: Thuật toán PMTL vẫn được sử dụng để giải bài toán $(\text{MOP}(F, U))$ với việc thay đổi các hàm mục tiêu và nghiệm x vào công thức (1.5) và (1.6).

Giải bài toán MTL bằng phương pháp vô hướng hóa

Khi giải bài toán quy hoạch đa mục tiêu không ràng buộc, ta có thể vô hướng hóa các hàm tác vụ dưới dạng:

$$\mathcal{L}(\theta_t) = \sum_{i=1}^m \lambda_i \mathcal{L}_i(\theta_t)$$

trong đó các tham số λ_i được giải bằng bài toán sau:

$$\max_{\lambda_i} -\frac{1}{2} \left\| \sum_{i=1}^m \lambda_i \nabla \mathcal{L}_i(\theta_t) \right\|^2, \quad \text{v.đ.k.} \quad \sum_{i=1}^m \lambda_i = 1, \quad \lambda_i \geq 0, \forall i = 1, \dots, m.$$

Khi thêm các ràng buộc bất đẳng thức \mathcal{G}_j ta có thể biểu diễn chúng dưới dạng:

$$\nabla \mathcal{G}_j(\theta_t) = (u_j - u_k)^T \nabla \mathcal{L}(\theta_t) = \sum_{i=1}^m (u_{ji} - u_{ki}) \nabla \mathcal{L}_i(\theta_t)$$

cho véc tơ ưu tiên u_k tương ứng. Do đó hướng giảm d_t có thể viết được dưới dạng tổ hợp tuyến tính của $\nabla \mathcal{L}_i(\theta_t)$ và $\nabla \mathcal{G}_j(\theta_t)$ như sau:

$$\mathcal{L}(\theta_t) = \sum_{i=1}^m \alpha_i \mathcal{L}_i(\theta_t), \quad \text{v.đ.k} \quad \alpha_i = \lambda_i + \sum_{j \in I_{\varepsilon}(\theta)} \beta_j (u_{ji} - u_{ki}), \quad (1.7)$$

trong đó, λ_i và β_j được xác định bằng cách giải bài toán (1.6) với véc tơ ưu tiên tương ứng u_k . Dựa vào công thức (1.7), ta có thể giải bài toán MTL bằng phương pháp vô hướng hóa và thiết lập hướng giảm sâu nhất để cập nhật cho tham số của mô hình.

Chương 2

Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu không ràng buộc

Trong phần này, chúng tôi sẽ trình bày về thuật toán đề xuất tìm hướng giảm kết hợp cỡ bước tự thích nghi để giải bài toán tối ưu đa mục tiêu trong cả hai trường hợp không ràng buộc, đồng thời chứng minh độ hội tụ của thuật toán. Thuật toán đề xuất cũng được áp dụng để giải cho lớp hàm tựa lồi và chúng tôi đã chứng minh được sự hội tụ của nghiệm về điểm dừng Pareto.

Khảo sát cách thiết lập cỡ bước trong các thuật toán hướng giảm

Gần đây, sự phát triển của các mô hình Học máy xử lý dữ liệu lớn với các hàm mất mát không lồi cần có các phương pháp thiết lập cỡ bước trong việc cập nhật tham số bằng hướng giảm một cách hiệu quả ([7], [14]). Việc thiết lập cỡ bước bằng thủ tục chính xác theo tia hoặc thủ tục quay lui tỏ ra không hiệu quả, có chi phí tính toán lớn trong trường hợp hàm phức tạp hoặc hàm véc tơ có số thành phần lớn ([4]). Để tránh cách làm đó, ta có thể thiết lập cỡ bước cố định dựa trên hệ số Lipschitz của hàm. Tuy nhiên, cách làm này có thể gây ra các ước lượng không chính xác và dẫn tới việc làm giảm tốc độ hội tụ của thuật toán ([22]).

Mặc dù Kiwiel đã phát triển phương pháp hướng giảm giải bài toán quy hoạch tựa lồi vào 2001 nhưng kết quả hội tụ khá chậm do việc giảm dần giá trị cỡ bước ([12]). Sau đó, Yu et al và ([31], 2019), Hu et al. ([11], 2020), đã

cải tiến cách làm này bằng việc sử dụng cỡ bước cố định nhưng các hàm mục tiêu vẫn phải thỏa mãn điều kiện Holder. Một số cách tiếp cận khác sử dụng hệ động lực hoặc mạng recurrent neural để giải bài toán quy hoạch giả lồi với tập ràng buộc không bị chặn ([15], [6]), trong đó cỡ bước được giữ cố định và không tự thích nghi.

Cỡ bước tự thích nghi đã được đề xuất bởi Konnov et al ([13]) và Ferreira et al ([10]). Trong [13], tác giả đề xuất hướng giảm điều hướng (conditional gradient) kết hợp cỡ bước tự thích nghi để áp dụng cho lớp hàm mục tiêu giả lồi với tập chấp nhận được bị chặn. Phương pháp này tiếp tục được mở rộng trong [10] cho trường hợp tập chấp nhận được không bị chặn nhưng không thể áp dụng cho trường hợp giải bài toán tối ưu có ràng buộc.

Để khắc phục các hạn chế trong các bài báo nêu trên, đồng thời áp dụng cỡ bước tự thích nghi trong việc giải các bài toán quy hoạch đa mục tiêu, trong báo cáo này, chúng tôi đề xuất một thuật toán hướng giảm với cỡ bước tự thích nghi để giải lớp bài toán với hàm mục tiêu không lồi, trơn và tập chấp nhận được có tính lồi, đóng, không bị chặn.

2.1 Cơ sở lý thuyết của thuật toán

Ta tiến hành giải bài toán ($\text{MOP}(F, U)$) trong trường hợp $U = \mathbb{R}^n$. Cho $x \in \mathbb{R}^n$, ta 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 (P_1) luôn có nghiệm tối ưu duy nhất vì hàm $\nabla F_j(x)^T s + \frac{1}{2} s^T s$ là hàm lồi mạnh theo biến s với $j = 1, \dots, m$. Trong trường hợp $m = 1$ hướng $s(x)$ giảm sâu nhất chính là $-\nabla F(x)$.

Việc xây dựng bài toán (P_1) do chúng ta đang tiến hành xấp xỉ

$$\max_{j=1, \dots, m} F_j(x+s) - F_j(x)$$

bằng khai triển Taylor cấp 1 tại x của mỗi hàm F_j . Việc thiết lập bài toán tối ưu (P_1) để tìm hướng giảm sâu nhất cho bài toán quy hoạch đa mục tiêu đã được đề xuất tại [8]. Nghiệm tối ưu cho (P_1) sẽ được ký hiệu bằng $\Theta(x)$. Do đó,

$$\Theta(x) = \inf_{s \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T s + \frac{1}{2} s^T s, \quad (2.1)$$

và

$$s(x) = \arg \min_{s \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T s + \frac{1}{2} s^T s \quad (2.2)$$

Mặc dù bài toán (P_1) là bài toán không trơn, nó vẫn có thể được xem là bài toán tối ưu bậc hai và có thể được giải một cách hiệu quả dựa vào điều kiện KKT (Karush-Kuhn-Tucker), (P_1) tương đương với

$$\begin{cases} \min & g(t, s) = t \\ \text{v.đ.k} & \nabla F_j(x)^T s + \frac{1}{2} s^T s - t \leq 0 \quad (1 \leq j \leq m) \\ & (t, s) \in \mathbb{R} \times \mathbb{R}^n. \end{cases} \quad (2.3)$$

Hàm mục tiêu Lagrangian tương ứng là

$$L((t, s), \lambda) = t + \sum_{j=1}^m \lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right).$$

Dựa vào điều kiện KKT, ta có hệ:

$$\sum_{j=1}^m \lambda_j = 1, \quad \sum_{j=1}^m \lambda_j (\nabla F_j(x) + s) = 0, \quad (2.4)$$

$$\lambda_j \geq 0, \quad \nabla F_j(x)^T s + \frac{1}{2} s^T s \leq t \quad (1 \leq j \leq m), \quad (2.5)$$

$$\lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right) = 0 \quad (1 \leq j \leq m). \quad (2.6)$$

Bài toán (3.3) có nghiệm tối ưu duy nhất, $(\Theta(x), s(x))$ do đây là bài toán tối ưu lồi và giả sử điều kiện Slater thỏa mãn khi đó tồn tại bộ số KKT $\lambda = \lambda(x)$ tương ứng với $s = s(x)$ và $t = \Theta(x)$ thỏa mãn điều kiện (3.4)-(3.7). Cụ thể từ (3.4) ta xác định:

$$s(x) = - \sum_{j=1}^m \lambda_j(x) \nabla F_j(x). \quad (2.7)$$

Hướng giảm sâu nhất được tìm ra bằng bộ trọng số λ sau khi giải hệ KKT đưa bài toán tối ưu đa mục tiêu về bài toán tối ưu vô hướng. Giả sử điều kiện chính quy được thỏa mãn, khi đó nghiệm KKT cho bài toán quy hoạch lồi (3.3) chính là nghiệm của bài toán tối ưu ban đầu và bài toán đối ngẫu. Từ

(3.9), ta có:

$$\begin{aligned}
\Theta(x) &= \sup_{\lambda \geq 0} \inf_{s \in \mathbb{R}^n} L((t, s), \lambda) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{s \in \mathbb{R}^n} \sum_{j=1}^m \lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s \right) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{s \in \mathbb{R}^n} \left(\sum_{j=1}^m \lambda_j(x) \nabla F_j(x)^T s + \frac{1}{2} s^T s \right) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} -\frac{1}{2} \left\| \sum_{j=1}^m \lambda_j(x) \nabla F_j(x) \right\|^2
\end{aligned} \tag{2.8}$$

Phần tiếp theo, ta sẽ đi khảo sát tính chất của Θ và phân tích quan hệ của nó với $s(x)$ và tính dừng của x .

Bổ đề 2.1 [8] Với những điều kiện giả thiết đặt ra thì

1. Với mọi $x \in \mathbb{R}^n$ thì $\Theta(x) \leq 0$.
2. 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, $x \in \mathbb{R}^n$ là điểm dừng Pareto khi và chỉ khi $\Theta(x) = 0$.
3. Hàm $s : \mathbb{R}^n \rightarrow \mathbb{R}^n$, cho bởi (2.2) bị chặn trên tập compact và $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$ cho bởi công thức (2.1) liên tục trên \mathbb{R}^n .

2.2 Thuật toán đề xuất

Algorithm 3 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:

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

(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

$$F(x^{k+1}) \leq F(x^k) + \sigma \left\langle JF(x^k), x^{k+1} - x^k \right\rangle \quad (2.9)$$

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**.

Thay vì giải trực tiếp bài toán (P_1) , ta tiến hành giải bài toán tìm nghiệm xấp xỉ sau: Nếu x không phải là điểm dừng Pareto, ta có s là nghiệm xấp xỉ của bài toán (P_1) với ngưỡng $\sigma \in]0, 1]$ nếu

$$f_x(s) + \frac{1}{2} \|s\|^2 \leq \sigma \Theta(x), \quad s \in U(x)$$

trong đó, $f_x(s) := \max_j (JF(x)s)_j$ và $\Theta(x)$ là giá trị tối ưu của bài toán (P_1) .

Bổ đề 2.2 *Giả sử x không phải điểm dừng Pareto và s là nghiệm xấp xỉ của bài toán (P_1) với ngưỡng $\sigma \in]0, 1]$. Khi đó*

$$\|s\| \leq 2\|A\|_{\infty,2}$$

Chứng minh. Nếu x không phải điểm dừng Pareto $\Rightarrow \Theta(x) < 0$. Đồng

thời, Θ là nghiệm xấp xỉ của bài toán (P_1) do đó

$$\begin{aligned}
f_x(s) + \frac{1}{2}\|s\|^2 &\leq \sigma\Theta(x) \\
\Rightarrow \|s\|^2 &\leq 2(\sigma\Theta(x) - f_x(s)) \leq 2(-f_x(s)) \\
&\leq 2(-JF(x)s)_j \quad \forall j = \overline{1, m} \\
&\leq 2\|JF(x)s\|_{\infty, 2} \\
&\leq 2\|JF(x)\|_{\infty, 2}\|s\| \\
\Rightarrow \|s\| &\leq 2\|A\|_{\infty, 2}
\end{aligned}$$

Bổ đề 2.3 [8] Nếu F là hàm khả vi và $JF(x)s < 0$, khi đó tồn tại $\varepsilon > 0$ (có thể phụ thuộc vào x, s và σ) sao cho

$$F(x + ts) < F(x) + \sigma t JF(x)s$$

với mọi $t \in]0, \varepsilon]$.

2.3 Sự hội tụ của thuật toán đề xuất

Định lý 2.1 Giả sử chuỗi $\{x^k\}$ được tạo bởi Thuật toán 5. Khi đó, mỗi điểm tới hạn (nếu có) của chuỗi $\{x^k\}$ là điểm dừng Pareto của bài toán $(MOP(F, U))$ với $U = \mathbb{R}^n$.

Chứng minh. Với y là điểm tới hạn (accumulation poin) của (x^k) , $s(y)$ và $\Theta(y)$ là nghiệm tối ưu của bài toán (P_1) tại y .

Theo Bổ đề 2.1, ta cần chứng minh $\Theta(y) = 0$. Rõ ràng dãy $F(x^k)$ là dãy giảm dần và ta có

$$\lim_{k \rightarrow \infty} F(x^k) = F(y)$$

Do đó,

$$\lim_{k \rightarrow \infty} \|F(x^k) - F(x^{k+1})\| = 0.$$

Ta xét chuỗi thứ tự $\{k_p\} = \{k_1, k_2, \dots\}$ với $p \rightarrow \infty$, thỏa mãn bất đẳng thức (2.9). Với các bước lặp trong chuỗi $\{k_p\}$ thì giá trị α_{k_p} không đổi.

Trường hợp 1: $\{k_p\}$ hữu hạn.

Nếu chuỗi $\{k_p\}$ hữu hạn với p đủ lớn, khi đó tồn tại bước K trở đi sao cho bất đẳng thức (2.9) không thỏa mãn nữa. Nói cách khác,

$$F_j(x^{K+1}) \geq F_j(x^K) + \sigma \alpha_K \langle JF(x^K), s(x^K) \rangle_j$$

với ít nhất một $j \in \{1, \dots, m\}$. Dựa theo Bổ đề 2.3, ta có $\max_j \langle JF(x^K), s(x^K) \rangle \geq 0$. Kết hợp với (2.11), ta thu được $\Theta(x^K) = 0$ và x^K là điểm dừng Pareto.

Trường hợp 2: $\{k_p\}$ vô hạn.

Trong trường hợp chuỗi $\{k_p\}$ vô hạn, ta luôn thu được

$$F(x^k) - F(x^{k+1}) \geq -\alpha_k \sigma \langle JF(x^k), s(x^k) \rangle \geq 0,$$

với mỗi $k \in \{k_p\}$ và do đó

$$\lim_{k \rightarrow \infty} \alpha_k \sigma \langle JF(x^k), s(x^k) \rangle = 0. \quad (2.10)$$

Xét $\alpha_k \in]0, 1]$ với mọi $k \in \{k_p\}$. Ta xét chuỗi $(x^{k_u})_u$ hội tụ về y . Ta xét hai khả năng:

$$\limsup_{u \rightarrow \infty} \alpha_{k_u} > 0$$

và

$$\limsup_{u \rightarrow \infty} \alpha_{k_u} = 0.$$

Trường hợp 2.1 Trong trường hợp này, tồn tại một chuỗi $(x^{k_\ell})_\ell$ hội tụ về y và thỏa mãn

$$\lim_{\ell \rightarrow \infty} \alpha_{k_\ell} = \bar{\alpha} > 0$$

Sử dụng (2.10) ta thu được

$$\lim_{\ell \rightarrow \infty} JF(x^{k_\ell}) s^{k_\ell} = 0$$

từ đó

$$\lim_{\ell \rightarrow \infty} \Theta(x^{k_\ell}) = 0$$

Vì $x \mapsto \Theta(x)$ là hàm liên tục, do đó $\Theta(y) = 0$ nên y là điểm dừng Pareto.

Trường hợp 2.2 Dựa vào kết quả trong Bổ đề 2.2 ta có chuỗi $(s^{k_u})_u$ bị chặn. Do đó, ta có thể lấy một chuỗi con $(x^{k_r})_r$ của $(x^{k_u})_u$ sao cho chuỗi $(s^{k_r})_r$ cũng hội tụ tới \bar{s} . Lưu ý rằng với mọi r ta đều có:

$$\max_i \left(JF(x^{k_r}) s^{k_r} \right)_i \leq \sigma \Theta(x^{k_r}) < 0$$

Do đó, xét giới hạn $r \rightarrow \infty$ ta thu được

$$\frac{1}{\sigma} \max_i (JF(y) \bar{s})_i \leq \Theta(y) \leq 0 \quad (2.11)$$

Lấy bất kỳ $q \in \mathbb{N}$. Với r đủ lớn,

$$t_{k_r} < \alpha_1^q$$

nghĩa là cỡ bước Adaptive không thỏa mãn (2.9) với $t = \alpha_1^q$, khi đó:

$$JF \left(x^{k_r} + (\alpha_1^q) s^{k_r} \right) \not\leq F \left(x^{k_r} \right) + \sigma \left(\alpha_1^q \right) JF \left(x^{k_r} \right) s^{k_r}$$

(với r đủ lớn). Xét giới hạn $r \rightarrow \infty$ ta thu được:

$$F_j \left(y + (\alpha_1^q) \bar{s} \right) \geq F_j(y) + \sigma \left(\alpha_1^q \right) (JF(y)\bar{s})_j$$

với ít nhất một $j \in \{1, \dots, m\}$. Lưu ý rằng bất đẳng thức này thỏa mãn với mọi $q \in \mathbb{N}$. Dựa vào Bổ đề 2.3 ta thu được:

$$\max_i (JF(y)\bar{s})_i \geq 0$$

Kết hợp với (2.11), thu được $\Theta(y) = 0$. Do đó y vẫn là điểm dừng Pareto.

Bây giờ, chúng ta quan tâm đến việc nghiên cứu tính hội tụ của thuật toán đề xuất khi hàm mục tiêu là hàm quasi-convex. Ta định nghĩa:

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

Bổ đề 2.4 [32] Nếu F là hàm tựa lồi và $x \in \mathbb{R}^n$ thì $\langle \nabla F_i(x^k), x - x^k \rangle \leq 0$ với mọi k và mọi $i = 1, \dots, m$.

Định lý 2.2 Giả sử F là hàm tựa lồi trên U . Nếu $T \neq \emptyset$ thì $\{x^k\}$ hội tụ đến điểm dừng Pareto.

Chứng minh. Dựa trên ý tưởng của Định lý 2.1, chúng ta lại xem xét dãy $k_p = k_1, k_2, \dots$ với $p \rightarrow \infty$, thỏa mãn bất đẳng thức (2.9).

Trường hợp 1: $\{k_p\}$ hữu hạn.

Do tính hữu hạn của dãy $\{k_p\}$, chúng ta có thể chứng minh sự tồn tại của một điểm x^K là điểm dừng Pareto sau một số bước hữu hạn. Do đó, khi F là một hàm tựa lồi trên U , dãy $\{x^k\}$ hội tụ đến một điểm dừng.

Trường hợp 2: $\{k_p\}$ vô hạn.

Theo Bổ đề 2 trong [32], ta có cho mọi $x \in U$ và mỗi k ,

tồn tại $\left\{ \lambda_j^k \right\}_{j=1}^m \subset [0, 1]$ sao cho $\sum_{j=1}^m \lambda_j^k = 1$ và

$$\left\| x^{k+1} - x \right\|^2 \leq \left\| x^k - x \right\|^2 + 2\alpha_k \left\langle \sum_{j=1}^m \lambda_j^k \nabla F_j \left(x^k \right), x - x^k \right\rangle + \frac{2}{\sigma} \sum_{j=1}^m \left(F_j \left(x^k \right) - F_j \left(x^{k+1} \right) \right) \quad (2.12)$$

với $k \in \{k_p\}$. Ta có chuỗi $\{F(x^k)\}$ là chuỗi giảm chặt. Do đó,

$\varepsilon_k := \sum_{j=1}^m [F_j(x^k) - F_j(x^{k+1})]$ dương với mọi k . Từ Bổ đề 2.4, ta có:

$$\|x^{k+1} - x\|^2 \leq \|x^k - x\|^2 + \frac{2}{\sigma} \varepsilon_k$$

Xét dãy $\sum_{k=1}^{\infty} \varepsilon_k$ hội tụ. Thật vậy, $\varepsilon_k > 0$ với mọi k và

$$\begin{aligned} \sum_{k=0}^{\ell} \varepsilon_k &= \sum_{k=0}^{\ell} \left(\sum_{i=1}^m (F_i(x^k) - F_i(x^{k+1})) \right) = \sum_{i=1}^m \left(\sum_{k=0}^{\ell} (F_i(x^k) - F_i(x^{k+1})) \right) \\ &= \sum_{i=1}^m (F_i(x^0) - F_i(x^{\ell+1})) \leq \sum_{i=1}^m (F_i(x^0) - F_i(x)). \end{aligned}$$

Do đó, $\{x^k\}$ hội tụ quasi-Fejér đến T . Sử dụng Bổ đề 1.3 chuỗi $\{x^k\}$ bị chặn. Xét \bar{x} là điểm tới hạn của $\{x^k\}$. Dãy $\{F(x^k)\}$ giảm dần đến $\bar{x} \in T$. Dựa theo Bổ đề 1.3, ta có $\{x^k\}$ hội tụ tới \bar{x} , cũng chính là điểm dừng Pareto dựa theo chứng minh của Định lý 2.1.

Định lý 2.3 *Giả sử F là hàm giả lồi. Nếu $T \neq \emptyset$ thì $\{x^k\}$ hội tụ tới điểm hữu hiệu yếu.*

Chứng minh. Theo [20], hàm giả lồi thì tựa lồi. Do đó, sử dụng Định lý 2.2, ta có $\{x_k\}$ hội tụ đến điểm dừng Pareto, cũng chính là điểm hữu hiệu yếu theo tính chất của hàm giả lồi.

2.4 Thuật toán đề xuất cải tiến với các 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 Ω_k ($k = 1, \dots, K$), với

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

Ta tiến hành giải bài toán tối ưu không ràng buộc trên từng vùng Ω_k thông qua việc kết hợp các ràng buộc bất đẳng thức \mathcal{G}_p giúp phân tách không gian. Ta cần giải bài toán sau:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} F(x) &= (F_1(x), F_2(x), \dots, F_m(x)) \\ \text{v.đ.k } \mathcal{G}_p(x) &= \langle u_p - u_k, F(x) \rangle \leq 0, \forall p = 1, \dots, K, \end{aligned} \tag{2.13}$$

Khi đó, ta tiến hành giải bài toán (MOP(F, U)) với tập $U = \{x \in \mathbb{R}^n \mid \mathcal{G}_p(x) \leq 0, p = 1, \dots, K\}$.

Tìm điểm khởi tạo

Để giải bài toán (2.13), ta cần tìm một điểm khởi tạo chấp nhận được thỏa mãn mọi ràng buộc. chúng tôi sử dụng phương pháp tìm điểm chấp nhận được dựa trên hướng giảm dựa trên ý tưởng của [3]. Trước khi giải bài toán tối ưu, ta sẽ tìm điểm khởi tạo dựa trên các bước sau:

- **Bước 1:** Khởi tạo ngẫu nhiên một điểm $x_0 \in \mathbb{R}^n$.
- **Bước 2:** Nếu x_0 là điểm chấp nhận được thì dừng lại. Ngược lại chuyển xuống **Bước 3**.
- **Bước 3:** Ta xác định tập các ràng buộc chặt $I(x_0) = \{p \mid \mathcal{G}_p(x_0) \geq 0, p = 1, \dots, K\}$. Ta có thể tìm một hướng giảm s_r như sau:

$$(s_r, \alpha_r) = \arg \min_{s \in \mathbb{R}^n, \alpha \in \mathbb{R}} \alpha + \frac{1}{2} \|s\|^2, \text{ v.đ.k } \nabla \mathcal{G}_p(x_0)^T s \leq \alpha, p \in I(x_0).$$

- **Bước 4:** Tiến hành cập nhật $x_0 = x_0 + \eta_r s_r$ và quay trở lại **Bước 2**.

Sau khi tìm được điểm khởi tạo chúng tôi đề xuất thuật toán như sau:

Algorithm 4 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:

$$\begin{aligned} \min_{\lambda_j, \gamma_p} & \left\| \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{p \in I_\varepsilon(x^k)} \gamma_p^k \nabla \mathcal{G}_p(x^k) \right\|^2 \\ \text{v.đ.k: } & \lambda_j \geq 0, \gamma_p \geq 0, \sum_{j=1}^m \lambda_j^k + \sum_{p \in I_\varepsilon(x^k)} \gamma_p^k = 1 \end{aligned}$$

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

Từ đó xác định được nghiệm tối ưu:

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

(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**.

Nhận xét 2.1 Định lý 2.1, Định lý 2.2, Định lý 2.3 ứng với Thuật toán 4 vẫn chính xác.

2.5 Tính toán thử nghiệm và đánh giá kết quả

Trong đồ án này, chúng tôi sẽ tiến hành kiểm thử độ hiệu quả của thuật toán đề xuất với các ví dụ quy hoạch đa mục tiêu không ràng buộc và có ràng buộc. Để chứng minh độ hiệu quả của thuật toán đề xuất chúng tôi tiến hành so sánh với các thuật toán giải MOP khác bằng các metric đánh giá sau:

- **Baselines:** Trong đồ án này, chúng tôi sẽ so sánh với Thuật toán PMTL [28] và Thuật toán MGDA [18].
- **Metric:** Vùng không gian ảnh được tạo bởi (dominated) bởi Pareto front được gọi là Hypervolume [23]. Chỉ số Hypervolume càng cao thì Pareto front tìm được càng tốt.

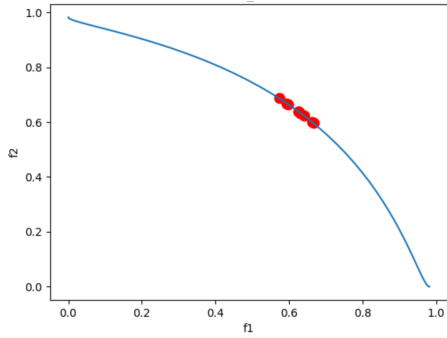
Cho tập k điểm $\mathcal{M} = \{m^j \mid m^j \in \mathbb{R}^m; j = 1, \dots, k\}$ và 1 điểm tham chiếu $\rho \in \mathbb{R}_+^m$, giá trị Hypervolume của S được đo bởi vùng không gian tạo bởi các điểm không trội bị chặn bởi $m \in \mathcal{M}$, khi đó thông số Hypervolume được tính bằng:

$$HV(S) = VOL \left(\bigcup_{m \in \mathcal{M}, m < \rho} \Pi_{i=1}^m [m_i, \rho_i] \right)$$

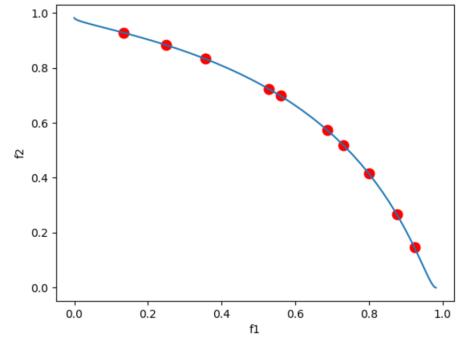
Ví dụ 2.1 (*Toy example (Lin et al [3])*) Xét bài toán tối ưu đa mục tiêu không lồi không ràng buộc sau:

$$\begin{aligned} \text{Min } F(x) = & \left\{ 1 - \exp^{-\sum_{i=1}^d (x_i - \frac{1}{d})^2}, 1 - \exp^{-\sum_{i=1}^d (x_i + \frac{1}{d})^2} \right\} \\ \text{v.đ.k } & x \in \mathbb{R}^2 \end{aligned}$$

Ta tiến hành thực nghiệm với số chiều $d = 20$. Ta tiến hành sử dụng Thuật toán 4 và Thuật toán 3 để giải ví dụ trên với 10 điểm khởi tạo thu được kết quả sau:



Hình 2.1: Kết quả Thuật toán 3 giải Ví dụ 2.1



Hình 2.2: Kết quả Thuật toán 4 giải Ví dụ 2.1

Các nghiệm thu được bởi Thuật toán 3 tập trung tại 1 vùng trên Pareto Front, trong khi đó nếu sử dụng Thuật toán 4, ta sẽ thu được các nghiệm trải đều trên Pareto Front.

Để so sánh độ hiệu quả của thuật toán đề xuất đầu tiên ta sẽ tiến hành giải ví dụ trên bằng Thuật toán 4 và Thuật toán MGDA [18] với cùng số điểm khởi tạo và cùng số vòng lặp. Chúng tôi sẽ chạy 2 thuật toán qua nhiều lần để giải ví dụ và đưa ra được kết quả trung bình cho các lần chạy với metric HV dưới đây:

Số điểm khởi tạo (điểm)	Số vòng lặp (lần)	Số lần chạy (lần)	HV trung bình PMTL	HV trung bình Thuật toán 4
50	100	10	0.85	1.52
40	200	13	0.87	1.49
45	150	15	0.88	1.53

Bảng 2.1: So sánh Thuật toán 4 và thuật toán MGDA giải Ví dụ 2.1

Với việc cải tiến sử dụng các véc tơ ưu tiên, ta có thể thấy HV trung bình thu được bởi Thuật toán 4 tốt hơn hẳn so với Thuật toán MGDA.

Tiếp theo, để so sánh hiệu quả của Thuật toán 4 với Thuật toán PMTL [3], chúng tôi tiến hành thực nghiệm bằng cách khởi tạo ngẫu nhiên cùng số điểm, cùng số vòng lặp để tìm nghiệm cho Ví dụ 1. Chúng tôi tiến hành giải nhiều lần Ví dụ 1 và thu được kết quả trung bình theo thời gian chạy thuật toán và HV như sau:

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 4 (s)	HV trung bình PMTL	HV trung bình Thuật toán 4
50	200	40	264.74	224.06	1.48	1.53
50	300	10	245.86	181.84	1.50	1.53
60	200	10	259.52	236.51	1.47	1.51

Bảng 2.2: So sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.1

Có thể thấy thuật toán đề xuất 4 có thời gian chạy nhanh hơn Thuật toán PMTL, đồng thời nghiệm thu được hội tụ về Pareto Front tốt hơn so với Thuật toán PMTL.

Để so sánh sự vượt trội của thời gian giải ra nghiệm tối ưu trải đều trên Pareto Front ta sẽ tiến hành chạy Thuật toán 4 với số vòng lặp ít hơn so với Thuật toán PMTL. Kết quả thu được cho thấy với số vòng lặp ít hơn thuật toán đề xuất 4 vẫn thu được tập nghiệm có HV trung bình tốt hơn hoặc bằng so với Thuật toán PMTL có số vòng lặp nhiều hơn, cụ thể như sau:

Số điểm khởi tạo	Thuật toán PMTL			Thuật toán 4		
	HV trung bình	Thời gian (s)	Số vòng lặp	HV trung bình	Thời gian (s)	Số vòng lặp
10	1.49	2.01	100	1.49	1.37	50
20	1.53	4.38	100	1.53	2.45	80
30	1.49	17.97	100	1.51	10.79	80
40	1.49	54.01	100	1.49	35.73	75
50	1.44	99.42	100	1.47	67.67	75

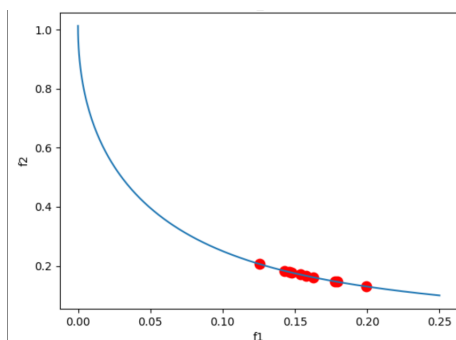
Bảng 2.3: Kết quả so sánh Thuật toán 4 và thuật toán PMTL trong các trường hợp khác nhau.

Ví dụ 2.2 Xét bài toán tối ưu đa mục tiêu không ràng buộc sau:

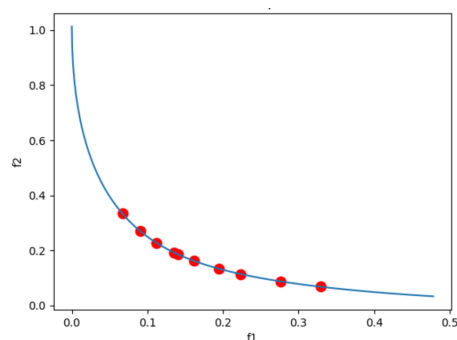
$$\text{Min } F(x) = \left\{ \frac{1}{25}x_1^2 + \frac{1}{100} \left(x_2 - \frac{9}{2}\right)^2, \frac{1}{25}x_2^2 + \frac{1}{100} \left(x_1 - \frac{9}{2}\right)^2 \right\}$$

v.đ.k $x \in \mathbb{R}^n$

Kết quả của Thuật toán 5 và Thuật toán 6 khi giải ví dụ 2 với 10 điểm khởi tạo như sau:



Hình 2.3: Kết quả Thuật toán 3 giải Ví dụ 2.2



Hình 2.4: Kết quả Thuật toán 4 giải Ví dụ 2.2

Với Thuật toán 3, ta có thể thấy các nghiệm Pareto tìm được tập trung tại 1 vùng trên Pareto Front. Nhược điểm này đã được khắc phục với Thuật toán 4 khi thêm các véc tơ ưu tiên.

Tương tự, đối với Ví dụ 2 để so sánh hiệu quả của thuật toán đề xuất với Thuật toán PMTL, chúng tôi tiến hành thực nghiệm bằng cách khởi tạo ngẫu nhiên các điểm để tìm nghiệm cho Ví dụ 2. Ta có kết quả như sau:

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 4 (s)	HV trung bình PMTL	HV trung bình Thuật toán 4
40	1000	7	629.65	474.18	1.02	1.03
40	500	10	250.18	230.21	1.01	1.02
45	300	15	320.29	456.25	0.98	1.01

Bảng 2.4: Kết quả so sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.2 trong các trường hợp khác nhau.

Ta có thể thấy thuật toán đề xuất có thời gian chạy nhanh hơn so với Thuật toán PMTL và nghiệm thu được hội tụ về Pareto Front tốt hơn so với Thuật

toán PMTL. Để so sánh sự vượt trội của thời gian giải ra nghiệm tối ưu trải đều trên Pareto Front ta sẽ tiến hành chạy Thuật toán 4 với số vòng lặp ít hơn so với Thuật toán PMTL. Kết quả thu được như sau:

Số điểm khởi tạo	Thuật toán PMTL			Thuật toán 4		
	HV trung bình	Thời gian (s)	Số vòng lặp	HV trung bình	Thời gian (s)	Số vòng lặp
10	0.97	3.31	200	0.97	2.59	150
20	0.96	15.73	200	0.96	12.97	150
30	0.95	77.04	300	0.96	55.07	200
40	0.95	251.4	300	0.95	174.13	200
50	0.94	835.25	300	0.95	564.18	200

Bảng 2.5: Kết quả so sánh Thuật toán 4 và thuật toán PMTL giải Ví dụ 2.2 trong các trường hợp khác nhau.

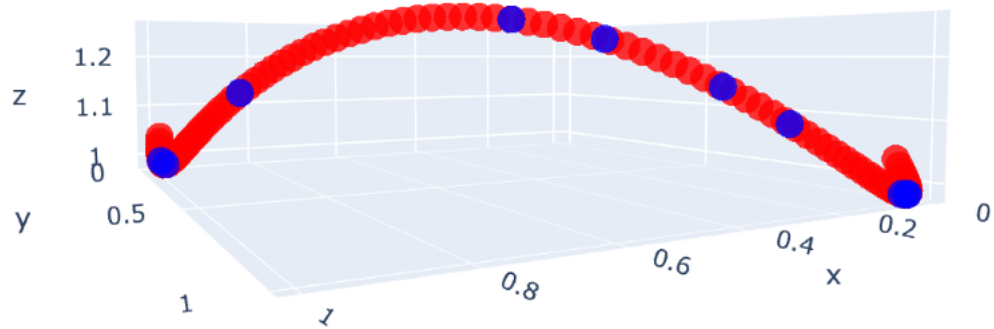
Nhận xét 2.2 Kết quả thu được cho thấy với số vòng lặp ít hơn thì Thuật toán đề xuất 4 vẫn thu được tập nghiệm có HV trung bình tốt hơn hoặc bằng so với Thuật toán PMTL có số vòng lặp nhiều hơn và do đó, tốc độ chạy của Thuật toán 4 nhanh hơn so với Thuật toán PMTL.

Ví dụ 2.3 (*Toy example (Lin et al [3])*) Xét ví dụ quy hoạch đa mục tiêu 3 chiều với các hàm không lồi sau:

$$\begin{aligned} \text{Min } F(x) = & \left\{ 1 - \exp^{-\sum_{i=1}^d (x_i - \frac{1}{d})^2}, 1 - \exp^{-\sum_{i=1}^d (x_i + \frac{1}{d})^2}, \right. \\ & \left. 2 - \exp^{-\sum_{i=1}^d (x_i - \frac{1}{d})^2} - \exp^{-\sum_{i=1}^d (x_i + \frac{1}{d})^2} \right\} \\ \text{v.đ.k } x \in \mathbb{R}^d \end{aligned}$$

Khi giải bài toán quy hoạch đa mục tiêu trong không gian 3 chiều, chúng tôi sử dụng véc tơ ưu tiên có dạng tọa độ cực $[\cos(\frac{k\pi}{K}) \sin(\frac{k\pi}{K}), \sin(\frac{k\pi}{K})^2, \cos(\frac{k\pi}{K})]$ với K là số lượng véc tơ tham chiếu và $k = 0, 1, \dots, K - 1$.

Ta tiến hành giải Ví dụ 2.3 với số chiều $d = 20$. Áp dụng Thuật toán 4 để giải ví dụ trên với 10 điểm khởi tạo ta thu được kết quả sau:



Hình 2.5: Kết quả Thuật toán 4 giải Ví dụ 2.3

Các nghiệm Pareto tìm được bởi Thuật toán 4 đã trải đều trên mặt Pareto Front.

Nhận xét 2.3 Tóm lại, các nghiệm được tìm thấy bởi thuật toán đề xuất hội tụ tốt hơn về mặt Pareto Front so với PMTL. Thuật toán đề xuất có thể coi là một phiên bản cải tiến của PMTL với khả năng tìm kiếm nghiệm Pareto nhanh hơn và thông số HV tốt hơn.

Chương 3

Thuật toán hướng giảm sâu nhất tự thích nghi cho bài toán quy hoạch đa mục tiêu có ràng buộc

3.1 Cơ sở lý thuyết của thuật toán

Điều kiện cần để một điểm $x \in U$ là nghiệm tối ưu Pareto địa phương của bài toán $(\text{MOP}(F, U))$ là:

$$JF(T(U, x)) \cap (-\mathbb{R}_{++}^m) = \emptyset, \quad (3.1)$$

tương đương với hệ

$$(\nabla F_j(x))^\top s < 0, \quad \forall j = 1, \dots, m, \quad s \in T(U, x), \quad (3.2)$$

không có nghiệm s . Tập các ràng buộc thỏa mãn chặt tại z được ký hiệu bởi $I_0(x)$,

$$I_0(x) := \{i \mid g_i(x) = 0\},$$

khi đó nón tuyến tính (*linearized cone*) tại x , ký hiệu là $C(x)$, được định nghĩa:

$$C(x) := \left\{ s \in \mathbb{R}^n \mid (\nabla g_i(x))^\top s \leq 0, \forall i \in I_0(x) \right\}.$$

Giả sử điều kiện chính quy (regular condition) $C(x) = T(U, x)$ thỏa mãn, khi đó điều kiện (3.1) (và 3.2) tương đương việc hệ

$$\begin{aligned} \nabla F_j(x)^\top s &< 0, \quad \forall j \\ \nabla g_i(x)^\top s &\leq 0 \quad \forall i \in I_0(x) \end{aligned}$$

không có nghiệm $s \in \mathbb{R}^n$. Gọi \hat{A} là ma trận được xác định bởi việc thêm các hàng $(\nabla g_i(x))^\top$ cho tất cả các ràng buộc i thỏa mãn chặt vào dưới ma trận Jacobian $JF(x)$. Điều kiện cần để hệ trên vô nghiệm là

$$R(\hat{A}) \cap -\mathbb{R}_{++}^m = \emptyset.$$

Xét tập chỉ số $I := \{1, \dots, l\}$ với $\varepsilon \in \mathbb{R}_+$ và $x \in \mathbb{R}^n$ sử dụng ý tưởng tại [8], ta định nghĩa tập I_ε như sau:

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

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

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

Ký hiệu $\Theta(x, \varepsilon)$ là giá trị tối ưu xấp xỉ tìm được. Dựa theo Bổ đề Farkas ta có $\Theta(x, 0) = 0$ với một điểm hữu hiệu x khi và chỉ khi tồn tại véc tơ $\lambda \in \mathbb{R}_+^m \setminus \{0\}, \beta \in \mathbb{R}_+^\ell$ với

$$\sum_{j=1}^m \lambda_j \nabla F_j(x) + \sum_{i=1}^\ell \beta_i \nabla g_i(x) = 0$$

và

$$\sum_{i=1}^\ell \beta_i g_i(x) = 0$$

Với $\lambda = (\lambda_j), \beta = (\beta_i) \in \mathbb{R}_{++}^m$ thì đây là điều kiện đủ cho x thành nghiệm tối ưu Pareto. Ta tiến hành giải (P_2) bằng hệ Karush-Kuhn-Tucker (KKT). Khi đó (P_2) tương đương với:

$$\left\{ \begin{array}{ll} \min & g(t, s) = t \\ \text{v.đ.k} & \nabla F_j(x)^T s + \frac{1}{2} s^T s - t \leq 0 \quad (1 \leq j \leq m) \\ & \nabla g_i(x)^T s + \frac{1}{2} s^T s - t \leq 0 \quad (i \in I_\varepsilon(x)) \\ & (t, s) \in \mathbb{R} \times \mathbb{R}^n. \end{array} \right. \tag{3.3}$$

Hàm mục tiêu Lagrangian có dạng:

$$L((t, s), \lambda) = t + \sum_{j=1}^m \lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right) + \sum_{i \in I_\varepsilon(x)} \beta_i (\nabla g_i(x)^T s + \frac{1}{2} s^T s - t).$$

Dựa trên điều kiện KKT, ta tiến hành giải hệ:

$$\sum_{j=1}^m \lambda_j + \sum_{i \in I_\varepsilon(x)} \beta_i = 1, \quad \sum_{j=1}^m \lambda_j (\nabla F_j(x) + s) + \sum_{i \in I_\varepsilon(x)} \beta_i (\nabla g_i(x) + s) = 0, \quad (3.4)$$

$$\lambda_j \geq 0, \quad \nabla F_j(x)^T s + \frac{1}{2} s^T s \leq t \quad (1 \leq j \leq m), \quad (3.5)$$

$$\beta_i \geq 0, \quad \nabla g_i(x)^T s + \frac{1}{2} s^T s \leq t \quad (i \in I_\varepsilon(x)), \quad (3.6)$$

$$\lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right) = 0 \quad (1 \leq j \leq m). \quad (3.7)$$

$$\beta_i (\nabla g_i(x)^T s + \frac{1}{2} s^T s - t) = 0 \quad (i \in I_\varepsilon(x)). \quad (3.8)$$

Bài toán (3.3) có nghiệm tối ưu duy nhất, $(\Theta(x, \varepsilon), s(x))$ do đây là bài toán tối ưu lồi và giả sử điều kiện Slater thỏa mãn khi đó tồn tại bộ số KKT $\lambda = \lambda(x)$ tương ứng với $s = s(x)$ và $t = \Theta(x, \varepsilon)$ thỏa mãn điều kiện (3.4)-(3.7). Cụ thể từ (3.4) ta xác định:

$$s(x) = - \sum_{j=1}^m \lambda_j(x) \nabla F_j(x) - \sum_{i \in I_\varepsilon(x)} \beta_i(x) \nabla g_i(x). \quad (3.9)$$

Hướng giảm sâu nhất được tìm ra bằng bộ trọng số λ sau khi giải hệ KKT đưa bài toán tối ưu đa mục tiêu về bài toán tối ưu vô hướng. Giả sử điều kiện chính quy được thỏa mãn, khi đó nghiệm KKT cho bài toán quy hoạch lồi (3.3) chính là nghiệm của bài toán tối ưu ban đầu và bài toán đối ngẫu. Từ (3.9), ta có:

$$\begin{aligned} \Theta(x, \varepsilon) &= \sup_{\lambda \geq 0} \inf_{s \in \mathbb{R}^n} L((t, s), \lambda) \\ &= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} -\frac{1}{2} \left\| \sum_{j=1}^m \lambda_j(x) \nabla F_j(x) + \sum_{i \in I_\varepsilon(x)} \beta_i(x) \nabla g_i(x) \right\|^2 \end{aligned} \quad (3.10)$$

3.2 Thuật toán đề xuất

Algorithm 5 Thuật toán hướng giảm sâu nhất với cỡ bước tự thích nghi giải bài toán quy hoạch đa mục tiêu có ràng buộc

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

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

Bước 3. (Vòng lặp chính)

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

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

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

trong đó, $I_\varepsilon(x^k) = \{i \in 1, \dots, l \mid g_i(x^k) \geq -\varepsilon\}$.

Khi đó, ta xác định lời giải tối ưu cho bài toán trên:

$$\lambda^k = (\lambda_j^k), \beta^k = (\beta_i^k), s(x^k) = - \left(\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k) + \sum_{i \in I_\varepsilon(x^k)} \beta_i^k \nabla g_i(x^k) \right)$$

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

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

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

(d) Tính toán cỡ bước:

Nếu

$$F(x^{k+1}) \leq F(x^k) + \sigma \langle JF(x^k), x^{k+1} - x^k \rangle \quad (3.11)$$

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

Bước 4. Đặt $k := k + 1$ và quay trở lại **Bước 3**.

3.3 Sự hội tụ của thuật toán đề xuất

Bổ đề 3.1 [8] Đặt $\|\cdot\|$ là một chuẩn trong \mathbb{R}^n và $\|\cdot\|^*$ là chuẩn đối ngẫu cấp dưới (subordinate dual norm). Khi JF liên tục Lipschitz thì

$$\|\nabla F_i(y) - \nabla F_i(\tilde{y})\|^* \leq L \|y - \tilde{y}\|, \quad i = 1, \dots, m,$$

với mọi $y, \tilde{y} \in \mathbb{R}^n$. Nếu $JF(x)s < 0$ và với

$$\tilde{\varepsilon} := -\frac{2(1-\sigma) \max_i (JF(x)s)_i}{L (\|s\|)^2} > 0.$$

thì

$$F(x + ts) \leq F(x) + \sigma t JF(x)s$$

với mọi $t \in [0, \tilde{\varepsilon}]$.

Theo điều kiện trên nếu $\|\cdot\|$ là chuẩn Euclidian và $s = s(x)$ thì $\tilde{\varepsilon} \geq (1 - \sigma)/L$.

Bổ đề 3.2 [8] Nếu $(x^k)_k$ là một chuỗi trong G với $\lim_{k \rightarrow \infty} x^k = y$ và $(\varepsilon_k)_k$ là một chuỗi trong \mathbb{R}_+ với $\lim_{k \rightarrow \infty} \varepsilon_k = 0$. Khi đó ta có

$$\limsup_{k \rightarrow \infty} \Theta(x^k, \varepsilon_k) \leq \Theta(y, 0).$$

Bổ đề 3.3 [8] Với mọi $i \in I$ xét ∇g_i liên tục Lipshitz với hệ số Lipshitz L tương ứng với chuẩn ∞ . Xét $x \in G, s \in \mathbb{R}^n, \varepsilon \in \mathbb{R}_+$ và $\rho \in -\mathbb{R}_{++}$ thỏa mãn

$$(\nabla g_j(x))^\top s \leq \rho \quad \text{với mọi } j \in I_\varepsilon(x).$$

Khi đó

$$x + ts \in G \text{ với mọi } t \in [0, \min\{1, \varepsilon/L, -2\rho/L\}].$$

Dựa vào các kết quả trên, ta có kết quả sau

Định lý 3.1 Xét JF và $\nabla g_i (i \in I)$ liên tục Lipshitz và điểm bắt đầu $x^1 \in G$. Khi đó với $\varepsilon \in \mathbb{R}_{++}$, chuỗi $(x^k)_k \subset G$ được thiết lập dựa vào giải bài toán (P_2) và cỡ bước được thiết lập dựa vào điều kiện Adaptive (3.11) thì ta có:

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

với mọi i hoặc:

$$\lim_{k \rightarrow \infty} \Theta(x^k, \varepsilon) = 0 \tag{3.12}$$

Chứng minh: Ta có

$$F(x^{k+1}) \leq F(x^k) + \alpha_k JF(x^k)s^k + \alpha_k^2 \frac{L}{2}e,$$

với $e = (1, \dots, 1)^\top$ và L là hằng số Lipshitz của JF và g_i . Giả sử (3.12) không thỏa mãn. Nhưng ta luôn có $\Theta(x^k, \varepsilon) \leq 0$ với mọi k , nghĩa là tồn tại 1 số $\rho \in -\mathbb{R}_{++}$ và dãy con với $\Theta(x^{k_u}, \varepsilon) < \rho$. Khi đó,

$$F(x^{k_u+1}) \leq F(x^{k_u}) + \alpha_{k_u} \rho e + \alpha_{k_u}^2 \frac{L}{2}e \tag{3.13}$$

với mọi u . Giả sử $-2\rho \leq \varepsilon$ và $-2\rho \leq L$. Dựa vào Bổ đề 3.3 thì $x^k + \alpha s^k \in G$ thỏa mãn với mọi $\alpha \leq -2\rho/L$. Nhưng dựa vào Bổ đề 3.1 thì điều kiện Armijo thỏa mãn với $0 \leq \alpha \leq -2\rho(1-\sigma)/L$ do đó ta có:

$$\alpha_{k_u} \in \left[-\rho \frac{1-\sigma}{L}, -2\rho \frac{1-\sigma}{L} \right].$$

Kết hợp biến đổi từ (3.13) thì:

$$F(x^{k_u+1}) \leq F(x^{k_u}) + \rho^2 \frac{1-\sigma}{L} \max \left\{ -\frac{1}{2}(1+\sigma), -2\sigma \right\}$$

thỏa mãn. Do đó, $\lim_{k \rightarrow \infty} F_i(x^k) = -\infty$ với mọi i .

Nhận xét 3.1 Khi các hàm mục tiêu không thể giảm tới $-\infty$ tức $\lim_{k \rightarrow \infty} F_i(x^k)$ hữu hạn thì dựa vào Định lý 3.1 ta có nghiệm x^k là điểm dừng Pareto.

3.4 Thuật toán đề xuất cải tiến với các véc tơ ưu tiên

Giải các bài toán quy hoạch đa mục tiêu bằng phương pháp vô hướng hóa có xu hướng tập trung các điểm tối ưu Pareto thu được vào một khu vực cụ thể trên mặt Pareto. Để phân bố đồng đều các điểm tối ưu Pareto trên toàn bộ mặt Pareto, chúng ta cần áp dụng các ràng buộc tuyến tính bổ sung để giúp phân chia không gian ảnh. Trên mỗi phân vùng của không gian ảnh, chúng ta sẽ tìm được một nghiệm tối ưu Pareto tương ứng.

Tương tự trong [28], ta xét K véc tơ ưu tiên $\{u_1, u_2, \dots, u_K\} \in \mathbb{R}_+^m$, và tiến hành chia không gian ảnh thành các không gian con subspaces Ω_k ($k = 1, \dots, K$) sử dụng ràng buộc bất đẳng thức sau:

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

Ý tưởng chính của các véc tơ ưu tiên là phân tách một bài toán MTL thành các bài toán con đa mục tiêu có ràng buộc khác nhau về sự cân đối giữa các nhiệm vụ trong bài toán gốc. Bằng cách giải quyết các bài toán con này một cách song song, thuật toán đề xuất có thể thu được một tập hợp các nghiệm tối ưu với từng tiêu chí cân bằng khác nhau. Tiếp theo, chúng ta giải bài toán tối ưu với ràng buộc trên mỗi vùng Ω_k bằng cách kết hợp các ràng buộc bất đẳng thức \mathcal{G}_p để hỗ trợ trong việc phân chia không gian. Chúng ta cần giải quyết bài toán sau:

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

Trong trường hợp này, chúng ta sẽ tích hợp các ràng buộc $\mathcal{G}_p, p = 1, \dots, K$ vào tập chấp nhận được U . Chúng tôi đề xuất phương pháp sau để tìm một điểm khởi tạo chấp nhận được:

- **Bước 1:** Khởi tạo ngẫu nhiên một điểm $x_0 \in \mathbb{R}^n$.
- **Bước 2:** Nếu x_0 là điểm chấp nhận được, **dừng lại**. Ngược lại, tiếp tục **Bước 3**.
- **Bước 3:** Chúng ta xác định tập các ràng buộc chặt là $I_\varepsilon(x_0) = \{i, p \mid \mathcal{G}_p(x_0) \geq 0, p = 1, \dots, K \text{ và } g_i(x) \geq -\varepsilon\}$. Chúng ta có thể tìm một hướng đi giảm s_r như sau:

$$(s_r, \alpha_r) = \arg \min_{s \in \mathbb{R}^n, \alpha \in \mathbb{R}} \alpha + \frac{1}{2} \|s\|^2,$$

$$\text{v.đ.k } \nabla \mathcal{G}_p(x_0)^T s \leq \alpha, p \in I_\varepsilon(x_0),$$

$$\nabla g_i(x_0)^T s \leq \alpha, i \in I_\varepsilon(x_0).$$

- **Bước 4:** Cập nhật $x_0 = x_0 + \eta_r s_r$ và quay trở lại **Bước 2**.

Sau khi đã tìm được một điểm khởi tạo chấp nhận được, chúng ta tiếp tục thuật toán đề xuất với các véc tơ ưu tiên (preference vector).

Algorithm 6 Thuật toán hướng giảm sâu nhất với cỡ bước tự thích nghi giải bài toán quy hoạch đa mục tiêu có ràng buộc với véc tơ ưu tiên

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:

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

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

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

Từ đó ta thu được nghiệm tối ưu sau:

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

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

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

(b) Nếu $\Theta(x^k, \varepsilon) = 0$ 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

$$F(x^{k+1}) \leq F(x^k) + \sigma \langle JF(x^k), x^{k+1} - x^k \rangle \quad (3.14)$$

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**.

Nhận xét 3.2 Dựa vào chứng minh ở trên Định lý 3.1 vẫn đúng.

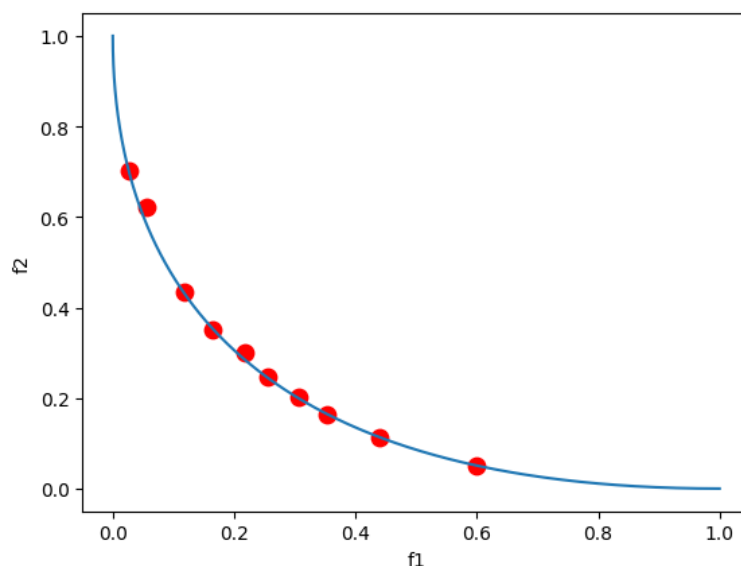
3.5 Tính toán thử nghiệm và đánh giá kết quả

Ví dụ 3.1 Xét bài toán quy hoạch đa mục tiêu lồi có ràng buộc sau

$$\text{Min } F(x) = \left\{ \frac{x_1^2 + x_2^2}{50}, \frac{(x_1 - 5)^2 + (x_2 - 5)^2}{50} \right\}$$

$$\text{v.đ.k } x_i \in [0, 5], i = 1, 2$$

Giải ví dụ trên bằng Thuật toán 6 với 10 điểm khởi tạo ngẫu nhiên khi đó kết quả thu được như sau:



Hình 3.1: Kết quả Thuật toán 6 giải ví dụ 3.1

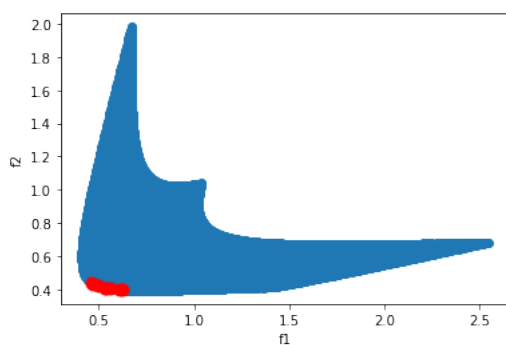
Với Mặt trận Pareto được hiển thị bằng màu xanh lam, chúng ta có thể thấy rằng các giải pháp thu được đều nằm trên Mặt trận Pareto và được phân Bố đều trên toàn bộ bề mặt.

Ví dụ 3.2 Xét bài toán quy hoạch không lồi có ràng buộc sau:

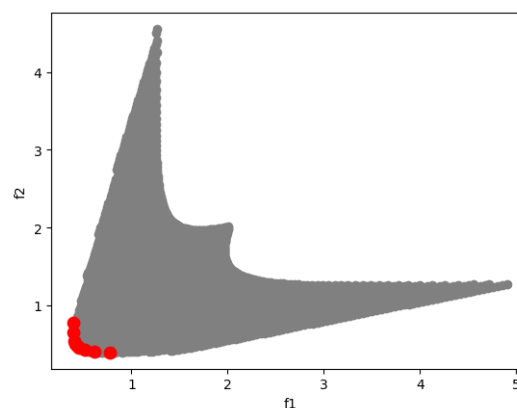
$$\begin{aligned} \text{Min} \quad & F(x) = \left\{ \frac{x_1^2 + x_2^2 + 3}{1 + 2x_1 + 8x_2}, \frac{x_2^2 + x_1^2 + 3}{1 + 2x_2 + 8x_1} \right\} \\ \text{v.đ.k} \quad & x \in U, \end{aligned}$$

trong đó, $U = \{x = (x_1, x_2)^T \in \mathbb{R}^2 | g_1(x) = -x_1^2 - 2x_1x_2 \leq -4; x_1, x_2 \geq 0\}$.

Sử dụng Thuật toán 5 và Thuật toán 6 để giải ví dụ trên với 10 điểm khởi tạo ngẫu nhiên ta thu được kết quả sau:



Hình 3.2: Kết quả Thuật toán 5 giải Ví dụ 3.2



Hình 3.3: Kết quả Thuật toán 6 giải Ví dụ 3.2

Chúng ta có thể thấy rằng cả hai thuật toán đều có thể tìm ra nghiệm Pareto trên mặt Pareto Front, nhưng với các ràng buộc tuyến tính giúp phân chia không gian hình ảnh, Thuật toán 6 tìm ra các giải pháp trải đều hơn trên mặt Pareto Front.

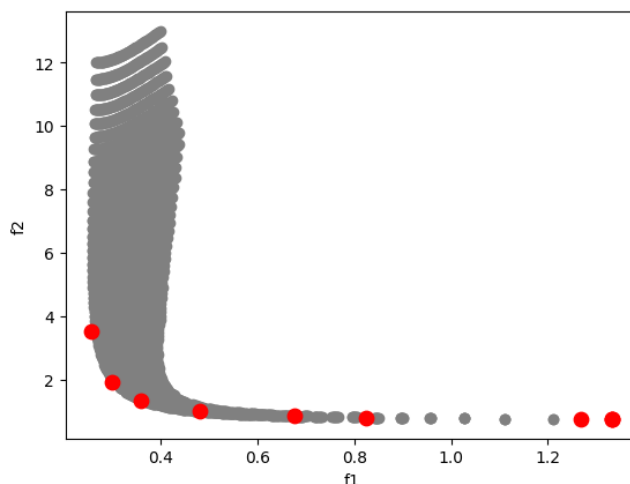
Ví dụ 3.3 Xét bài toán tối ưu đa mục tiêu có ràng buộc với 2 hàm mục tiêu không lồi như sau

$$\text{Min} F(x) = \left\{ \frac{x_1 + 1}{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.50}, \frac{x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20.00}{x_2} \right\}$$

$$\text{v.đ.k } x_1, x_2 \in U,$$

$$\text{với } U = \{2x_1 + x_2 \leq 6, 3x_1 + x_2 \leq 8, x_1 - x_2 \leq 1, x_1, x_2 \geq 1\}$$

Kết quả sử dụng Thuật toán 6 giải ví dụ 3.3 với 10 điểm khởi tạo cho kết quả như sau:



Hình 3.4: Kết quả Thuật toán 6 giải Ví dụ 3.3

Với không gian ảnh màu xám, ta thấy các nghiệm tìm Pareto tìm được (màu đỏ) đều nằm trên Pareto Front.

Ví dụ 3.4 Xét ví dụ quy hoạch đa mục tiêu lồi có ràng buộc với 3 hàm mục tiêu sau đây:

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

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

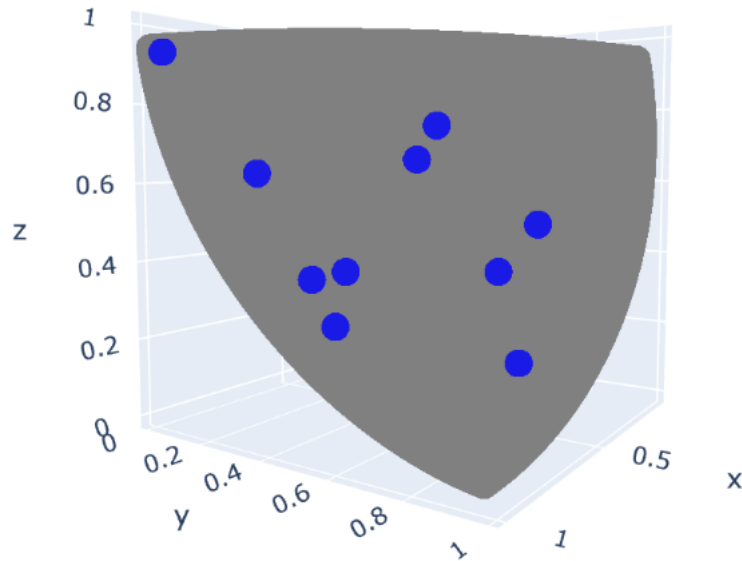
trong đó:

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

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

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

Kết quả của Thuật toán 6 giải ví dụ này với 10 điểm khởi tạo như sau:



Hình 3.5: Kết quả giải Ví dụ 3.4 của Thuật toán 6

Ta thấy các nghiệm tìm được vẫn trải đều trên Pareto Front cho thấy sự hiệu quả của thuật toán trong trường hợp giải ví dụ tối ưu 3 mục tiêu.

Chương 4

Ứng dụng vào bài toán học đa nhiệm

4.1 Thuật toán đề xuất giải bài toán MTL

Tương tự trong [3], chúng tôi sẽ sử dụng K véc tơ ưu tiên $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$ để biểu thị cho việc đánh đổi cho độ ưu tiên của các tác vụ trong bài toán MTL. Đồng thời sử dụng các ràng buộc bất đẳng thức tuyến tính $\mathcal{G}_p(\theta_t) = \langle u_p - u_k, \mathcal{L}(\theta_t) \rangle \leq 0, \forall p = 1, \dots, K$ để phân tách không gian ảnh thành các bài toán con. Sau đó, chúng tôi đề xuất sử dụng cỡ bước tự thích nghi để cập nhật cho *learning rate* của mạng Neural. Thuật toán đề xuất cho MTL được trình bày chi tiết dưới đây.

Algorithm 7 Thuật toán đề xuất cho việc giải bài toán MTL

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 θ_r^k cho mạng Neural.
- 4: Tìm bộ tham số hữu hiệu θ_0^k từ θ_r^k sử dụng phương pháp hướng giảm.
- 5: **for** $t = 1$ to T **do**
- 6: Tìm hướng giảm thông qua việc giải bài toán sau:

$$\begin{aligned} \min_{\lambda_j, \gamma_p} & \left\| \sum_{j=1}^m \lambda_j^k \nabla \mathcal{L}_j(\theta_t^k)^T + \sum_{p \in I(\theta_t^k)} \gamma_p^k \nabla \mathcal{G}_p(\theta_t^k) \right\|^2 \\ \text{v.đ.k: } & \lambda_j \geq 0, \gamma_p \geq 0, \sum_{j=1}^m \lambda_j^k + \sum_{p \in I(\theta_t^k)} \gamma_p^k = 1 \end{aligned}$$

- 7: Trong đó, $I_\varepsilon(\theta_t^k) := \{p \in 1, \dots, K | \mathcal{G}_p(\theta_t^k) \geq -\varepsilon\}$.
- 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 $\theta_{t+1}^k := \theta_t^k + \alpha_t s(\theta_t^k)$.
- 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**

Output: Bộ tham số tối ưu cho từng bài toán con $\{\theta_T^k | k = 1, \dots, K\}$.

4.2 Bài toán MTL cho thị giác máy tính

Trong phần này, chúng tôi sẽ triển khai thuật toán được đề xuất và đánh giá hiệu quả của nó đối với vấn đề phân loại hình ảnh hai tác vụ bằng cách sử dụng bộ dữ liệu Multi-MNIST được cung cấp tại [3]. Đồng thời, chúng tôi sẽ so sánh hiệu quả của thuật toán đề xuất với Thuật toán PMTL.

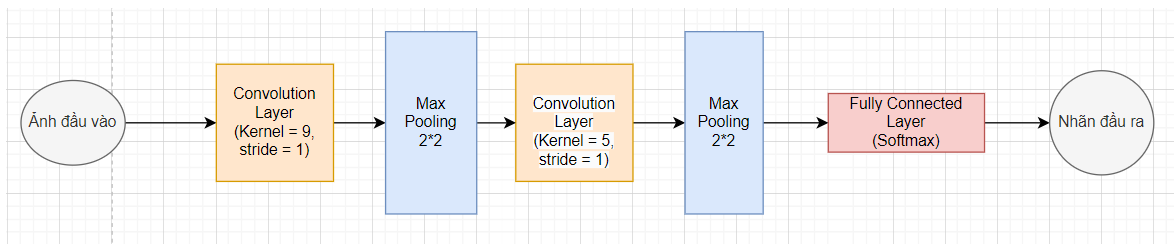
Multi-MNIST là bộ dữ liệu được xây dựng dựa trên việc ghép 2 chữ số trong bộ dữ liệu MNIST [36] vào 1 ảnh. Bài toán MTL phân loại ảnh sẽ học

đồng thời hai hàm mất mát biểu thị cho việc dự đoán 2 chữ số, 1 ở góc trên bên trái và 1 ở góc dưới bên phải lồng nhau. Tiến hành chia tập Train gồm 120000 ảnh và tập Test gồm 20000 ảnh. Hình ảnh minh họa cho bộ dữ liệu ở dưới đây:



Hình 4.1: Minh họa Multi-MNIST

Chúng tôi xây dựng mạng Lenet tương tự trong [36] với các layer như sau:



Hình 4.2: Kiến trúc mạng LeNet sử dụng trong Đồ án

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:

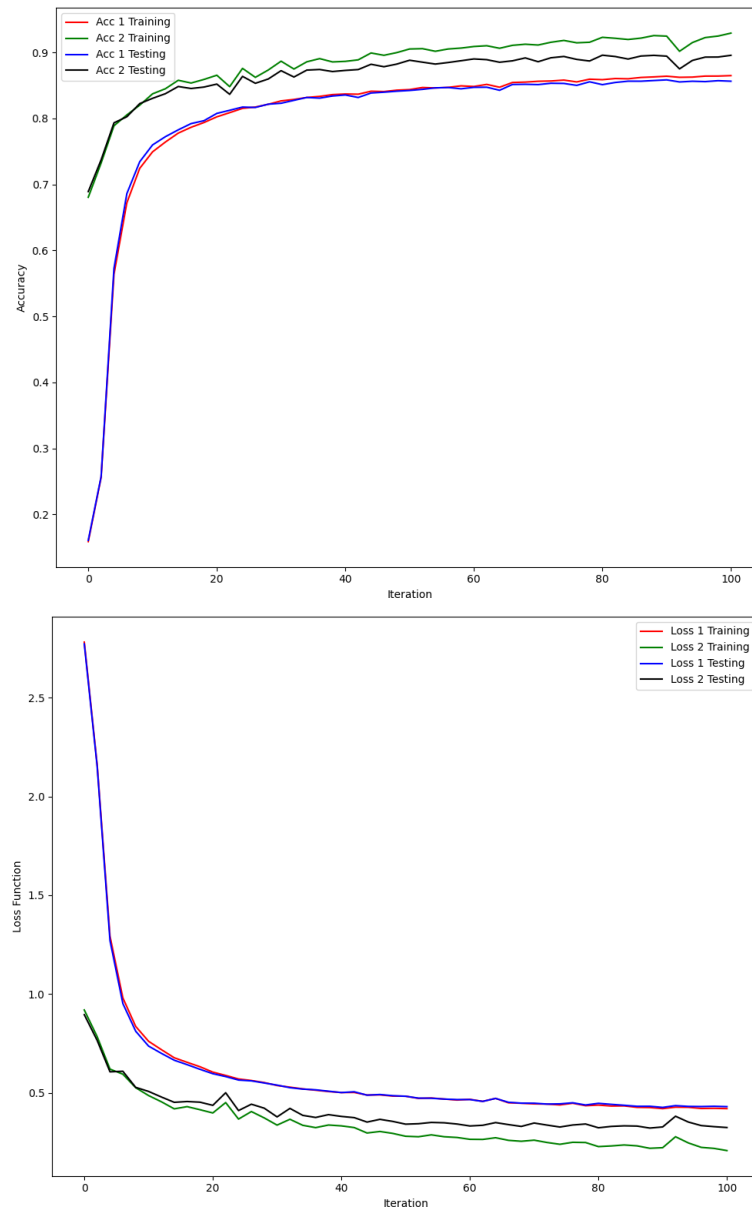
$$L_1(\hat{y}_1, y_1) = - \sum_k^K y_1^{(k)} \log \hat{y}_1^{(k)}$$

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

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. Ta cần học đồng thời cả hai hàm Cross-entropy trên.

Tiến hành quá trình huấn luyện với 10 véc tơ tham chiếu $[\cos(\frac{k\pi}{2K}), \sin(\frac{k\pi}{2K})]$, $k = 0, 1, \dots, 9$ mỗi lần huấn luyện với 100 epochs. Các siêu tham số được thiết lập

như sau: $\sigma = 0.01$, $\kappa = 0.95$ và learning rate $lr = 0.0005$. Ta minh họa kết quả huấn luyện với véc tơ tham chiếu $u_1 = (1, 0)$ như sau:

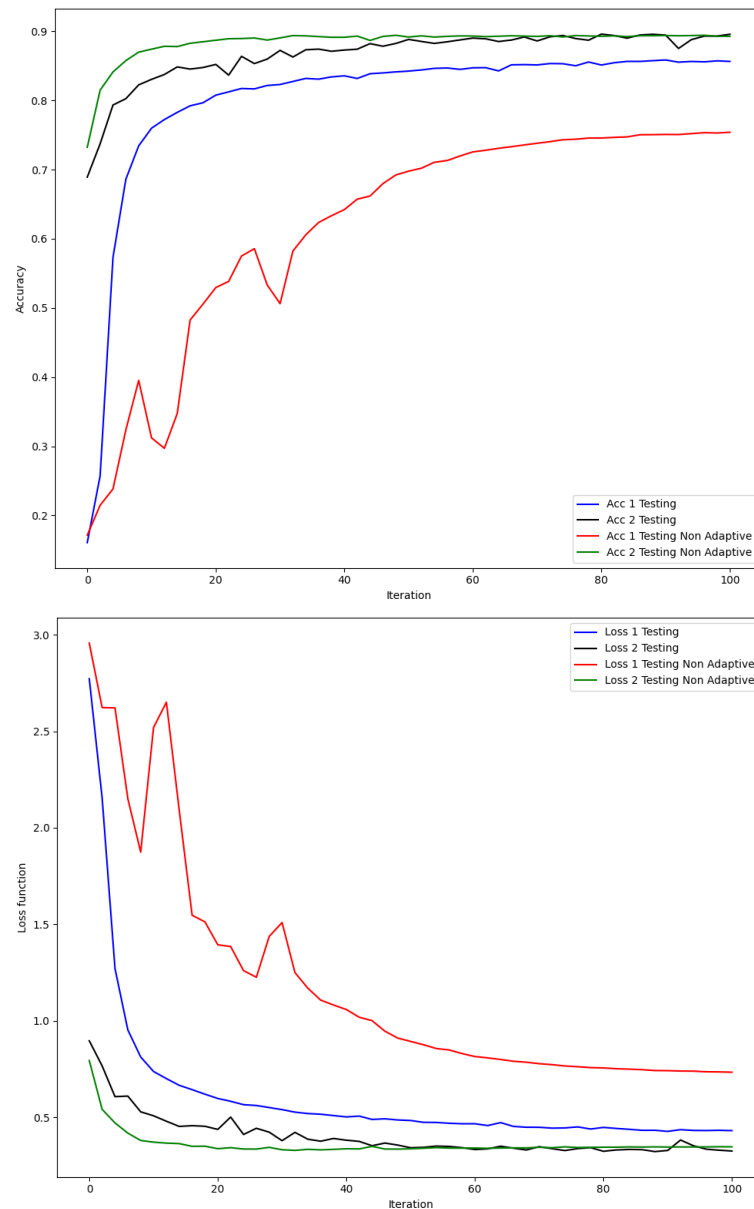


Hình 4.3: Quá trình huấn luyện Thuật toán 7 trên tập Multi-MNIST

Kết quả thực nghiệm cho thấy 2 tác vụ phân loại ảnh đều được học đồng thời và tương đối hiệu quả.

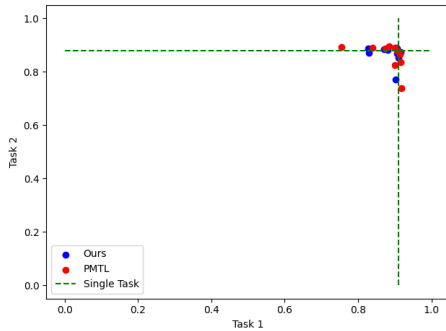
Ta tiến hành so sánh thuật toán 7 với thuật toán PMTL về tốc độ học và accuracy phân loại trên tập Test. Sau 5 lần chạy với các giá trị learning rate khác nhau, ta thiết lập learning rate tốt nhất cho PMTL là 0.0001 tương tự trong [3]. Hình ảnh so sánh quá trình huấn luyện của thuật toán đề xuất (đường 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 tham chiếu \mathbf{u}_1 như sau:



Hình 4.4: So sánh Thuật toán 7 và PMTL trên tập Multi-MNIST

Kết quả thực nghiệm cho thấy cả hai thuật toán đều có thể học đồng thời hai tác vụ với hiệu quả tương đối giống nhau, thể hiện qua so sánh giá trị độ chính xác trên tập Test. Tuy nhiên, về quá trình huấn luyện, Thuật toán 7 có lợi thế hơn PMTL ở chỗ nó sử dụng cỡ bước Thích ứng (Adaptive), dẫn đến hàm mất mát giảm đều hơn và do đó độ chính xác tăng liên tục trong quá trình đào tạo. Trong khi đó, PMTL sử dụng learning rate cố định, điều này có thể dẫn đến sự hội tụ chậm hơn trong quá trình huấn luyện. Bảng so sánh giá trị độ chính xác trên tập Test giữa thuật toán đề xuất và PMTL sẽ được minh họa dưới đây:



Hình 4.5: So sánh Pareto Front tìm được bởi Thuật toán 7 và PMTL trên tập Multi-MNIST

	Trung bình Accuracy Task 1	Trung bình Accuracy Task 2
PMTL	0.87 ± 0.01	0.85 ± 0.01
Đề xuất	0.88 ± 0.01	0.86 ± 0.01

Bảng 4.1: So sánh 2 Thuật toán 7 và PMTL khi chạy trên Multi-MNIST

Nhận xét 4.1 Kết quả cho thấy Thuật toán 7 hoạt động hiệu quả hơn thuật toán PMTL về Accuracy trung bình của 2 tác vụ trên tập Multi-MNIST.

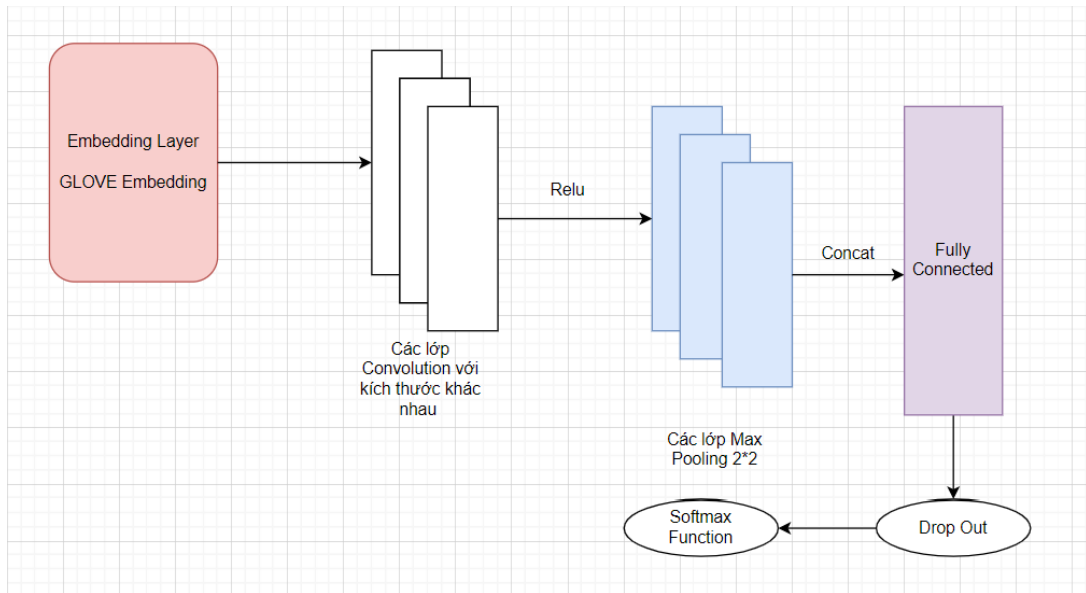
4.3 Bài toán MTL cho xử lý ngôn ngữ tự nhiên

Chúng tôi 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 [37]. Bộ dữ liệu gồm 111564 câu nhận xét. Ta sẽ sử dụng các nhận xét của khách hàng về thuốc để dự báo đồng thời điều kiện mà thuốc khách hàng mua được sử dụng và độ hài lòng của khách hàng.

	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 4.6: Minh họa tập dữ liệu Drug

Sau khi tiền xử lý các câu nhận xét ta sẽ tiến hành Embedding các câu nhận xét bằng phương pháp GLOVE [38] để chuyển từ không gian các từ, các câu sang không gian ma trận số, sau đó truyền ma trận đã được embed qua mô hình TextCNN [39]. Chi tiết mô hình được xây dựng như sau:



Hình 4.7: Mô hình TextCNN xây dựng

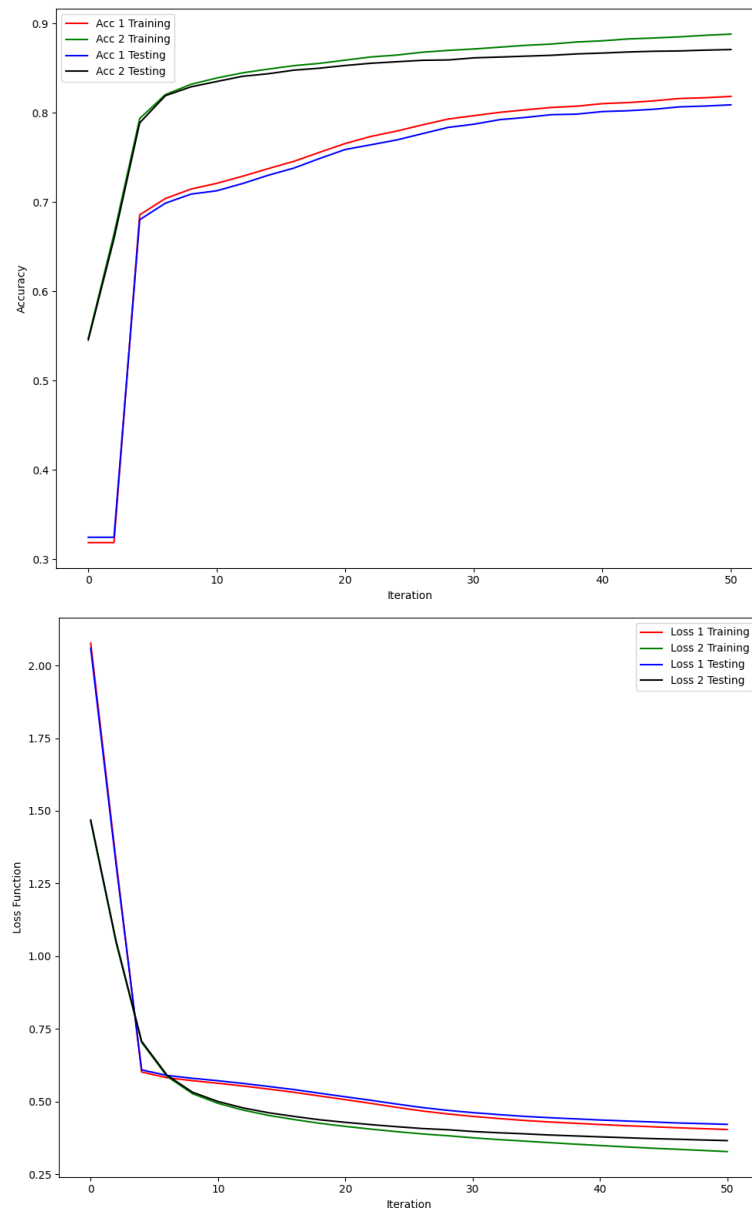
Ta xây dựng hàm mất mát bằng hai hàm Cross Entropy cho bài toán phân loại đa nhãn.

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

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

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.

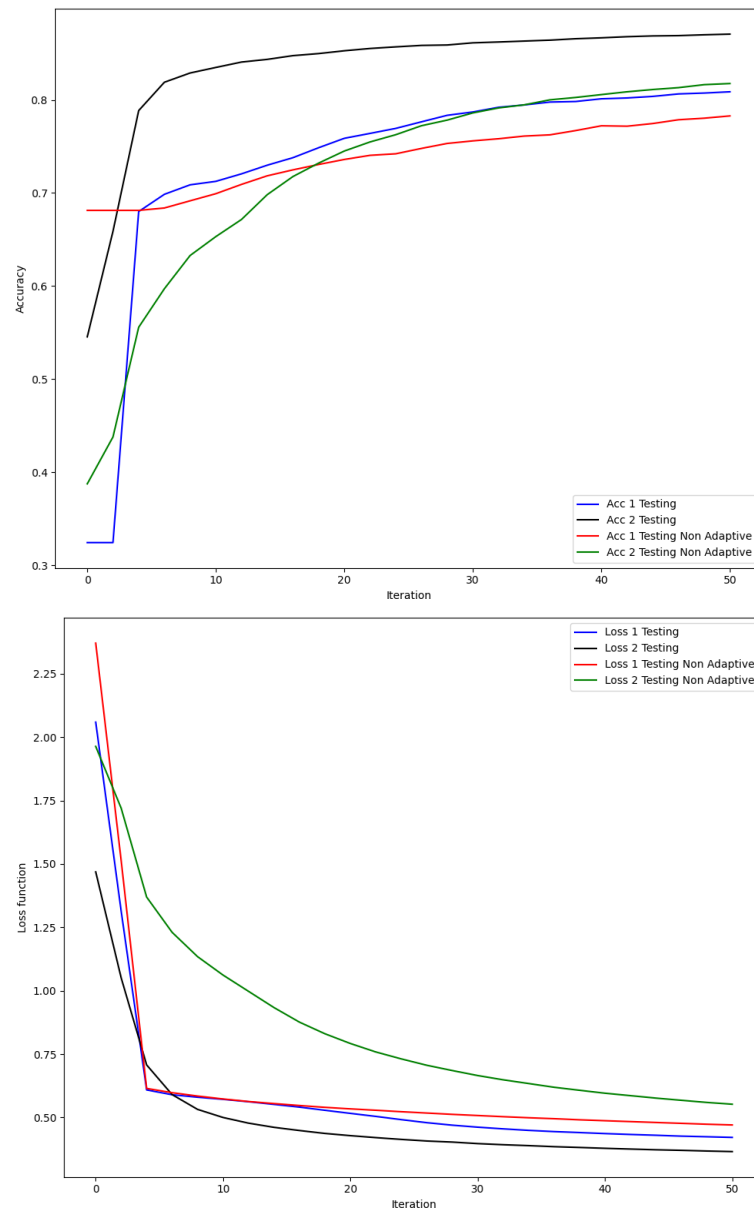
Tiến hành huấn luyện mô hình bằng thuật toán 7 với 5 véc tơ ưu tiên $[\cos(\frac{k\pi}{2K}), \sin(\frac{k\pi}{2K})]$, $k = 0, 1, \dots, 5$ mỗi véc tơ ưu tiên trải qua 50 epochs với các siêu tham số sau khi tuning được cài đặt là $\sigma = 0.05$, $\kappa = 0.95$ và learning rate $lr = 0.001$. Tỷ lệ tập Train/Test là 3 : 1. Kết quả quá trình huấn luyện với các thông số của hàm mất mát và Accuracy trên tập Test như sau:



Hình 4.8: Quá trình huấn luyện Thuật toán 7 trên tập Drug Review

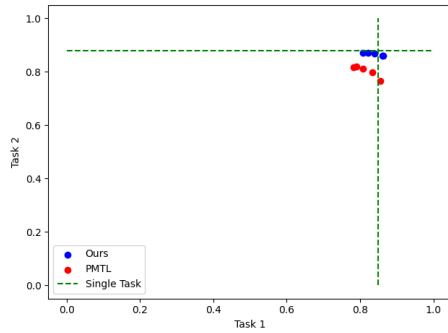
Ta thấy giá trị hàm mất mát giảm đều trên cả tập Train và tập Test và Accuracy thì tăng đều chứng tỏ mô hình học đồng thời cả 2 tác vụ khá hiệu quả.

Tiến hành so sánh hiệu suất của thuật toán đề xuất với thuật toán PMTL được giữ nguyên $lr = 0.0001$ (là siêu tham số tối ưu địa phương sau khi tuning) để kiểm chứng độ hiệu quả, ta thu được kết quả sau:



Hình 4.9: So sánh 2 Thuật toán 7 và PMTL khi chạy trên Drug Review

Kết quả cho thấy chỉ sau 20 epochs thì Accuracy của thuật toán 7 đã cao hơn hẳn so với thuật toán PMTL. Điều này là kết quả của việc điều chỉnh hệ số học phù hợp trong quá trình huấn luyện dựa trên điều kiện cập nhật Adaptive. Tiến hành đo kết quả Accuracy trung bình 2 tác vụ của 2 thuật toán PMTL và thuật toán 7 sau 10 lần chạy khác nhau với cùng bộ thông số khởi tạo ta thu được kết quả dưới đây:



Hình 4.10: So sánh Pareto Front của PMTL và Thuật toán 7 trên tập Drug Review

	Trung bình Accuracy Task 1	Trung bình Accuracy Task 2
PMTL	0.82 ± 0.01	0.83 ± 0.01
Đề xuất	0.84 ± 0.01	0.86 ± 0.01

Bảng 4.2: So sánh 2 Thuật toán 7 và PMTL khi chạy trên Drug Review

Nhận xét 4.2 Kết quả cho thấy độ hiệu quả của thuật toán 7 so với Thuật toán PMTL. Pareto Front tìm được bởi Thuật toán 7 trội hơn hẳn Pareto Front tìm được bởi PMTL.

Kết luận

Đóng góp chính của đề án

1. Đề xuất kết hợp phương pháp hướng giảm sâu nhất đa mục tiêu với cách cập nhật cỡ bước tự thích nghi để giải bài toán quy hoạch đa mục tiêu có ràng buộc và không ràng buộc.
2. Chứng minh được tính chất hội tụ mạnh của thuật toán đề xuất trong trường hợp giải bài toán quy hoạch đa mục tiêu không ràng buộc.
3. Đề xuất áp dụng cách thức cập nhật hệ số học theo điều kiện Adaptive khi huấn luyện mô hình Deep Learning giải bài toán MTL.
4. Thực nghiệm độ hiệu quả hơn của thuật toán đề xuất so với thuật toán PMTL trong các trường hợp giải bài toán quy hoạch đa mục tiêu và bài toán MTL trong lĩnh vực CV và NLP.
5. Kết quả của Đề 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 của đề án trong tương lai

Trong tương lai, tác giả sẽ nghiên cứu các cách thiết lập cỡ bước khác nhau giúp nâng cao hiệu quả của thuật toán tìm hướng giảm sâu nhất giải bài toán tối ưu đa mục tiêu. Đồng thời, áp dụng thuật toán đề xuất vào nhiều bài toán MTL khác như xử lý tiếng nói hay dịch máy,...

Tài liệu tham khảo

- [1] Bauschke, H.H., Combettes, P.L.: Convex analysis and monotone operator theory in hilbert spaces. Springer, New York (2011)
- [2] Matthias Ehrgott, 2005. Multicriteria Optimization, Springer Books, Springer, edition 0, number 978-3-540-27659-3, July.
- [3] Lin, Xi and Zhen, Hui-Ling and Li, Zhenhua and Zhang, Qing-Fu and Kwong, Sam: Pareto multi-task learning, Advances in neural information processing systems (2019)
- [4] Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)
- [5] Rockafellar, R.T.: Convex analysis. Princeton University Press, Princeton, New Jersey (1970)
- [6] W. Bian, L. Ma, S. Qin, X. Xue: Neural network for nonsmooth pseudo-convex optimization with general convex constraints, Neural Networks 101, 1-14 (2018).
- [7] Cevher, V., Becker, S., Schmidt, M.: Convex optimization for big data. Signal Process. Magaz. 31, 32–43 (2014).
- [8] Fliege and Svaiter, Benar Fux: Steepest descent methods for multicriteria optimization, Mathematical methods of operations research (2000).
- [9] J.E. Dennis, R.B. Schnabel: Numerical methods for unconstrained optimization and nonlinear equations, Prentice-Hall, New Jersey, 1983.
- [10] O. P. Ferreira, W. S. Sosa, On the Frank–Wolfe algorithm for non-compact constrained optimization problems, Optimization, 71:1, 197-211 (2022).

- [11] Y. Hu , J. Li, C. K. Yu, Convergence Rates of Subgradient Methods for Quasiconvex Optimization Problems, Computational Optimization and Applications, 77, 183–212 (2020).
- [12] K. C. Kiwiel, Convergence and efficiency of subgradient methods for quasiconvex minimization, Math. Program., Ser. A 90: 1–25 (2001).
- [13] I. V. Konnov, Simplified versions of the conditional gradient method, Optimization, 67(12), 2275-2290 (2018).
- [14] Guanghui Lan, First-order and Stochastic Optimization Methods for Machine Learning, Springer Series in the Data Sciences, Springer Nature Switzerland (2020)
- [15] N. Liu, J. Wang, S. Qin: A one-layer recurrent neural network for non-smooth pseudoconvex optimization with quasiconvex inequality and affine equality constraints, Neural Networks 147, 1-9 (2022).
- [16] Yura Malitsky, Konstantin Mishchenko, Adaptive Gradient Descent without Descent, Proceedings of Machine Learning Research, 119:6702-6712 (2020).
- [17] Sener, O., Koltun, V. (2018). Multi-task learning as multi-objective optimization. Advances in neural information processing systems, 31.
- [18] Désidéri, Jean-Antoine. "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization." Comptes Rendus Mathématique 350.5-6 (2012): 313-318.
- [19] Luc, Dinh. (2005). Generalized Convexity in véc tơ Optimization. Handbook of Generalized Convexity and Generalizes Monotonicity. 76. 10.1007/0-387-23393-85.
- [20] O. Mangasarian, Pseudo-convex functions, Siam Control, 8, 281-290 (1965)
- [21] D.T. Luc, Theory of véc tơ optimization, Lecture Notes in Economics and Mathematical Systems, vol. 319, Springer, Berlin, 1989, pp. 101–109.
- [22] Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.

- [23] Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4): 257–271.
- [24] Xu, H.K.: Iterative algorithms for nonlinear operators. *J. London Math. Soc.* 66, 240-256 (2002)
- [25] C.K. Yu, Y. Hu, X. Yang & S. K. Choy, Abstract convergence theorem for quasi-convex optimization problems with applications, *Optimization*, 68(7), 1289-1304, 2019.
- [26] Yura Malitsky, Konstantin Mishchenko, Adaptive Gradient Descent without Descent, *Proceedings of Machine Learning Research*, 119:6702-6712 (2020).
- [27] O. Mangasarian, Pseudo-convex functions, *Siam Control*, 8, 281-290 (1965)
- [28] Lin, Xi, et al. "Pareto multi-task learning." *Advances in neural information processing systems* 32 (2019).
- [29] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [30] Xu, H.K.: Iterative algorithms for nonlinear operators. *J. London Math. Soc.* 66, 240-256 (2002)
- [31] C.K. Yu, Y. Hu, X. Yang & S. K. Choy, Abstract convergence theorem for quasi-convex optimization problems with applications, *Optimization*, 68(7), 1289-1304, 2019.
- [32] Cruz, J. B., Pérez, L. L., Melo, J. G. (2011). Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Analysis: Theory, Methods and Applications*, 74(16), 5268-5273.
- [33] Liu, Hai-Lin, Fangqing Gu, and Qingfu Zhang. "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems." *IEEE transactions on evolutionary computation* 18.3 (2013): 450-455.

- [34] Kiwiel, Krzysztof C. "Convergence and efficiency of subgradient methods for quasiconvex minimization." *Mathematical programming* 90 (2001): 1-25.
- [35] Horst, R., Muu, L.D., Nast, M. Branch-and-bound decomposition approach for solving quasiconvex-concave programs. *J Optim Theory Appl* 82, 267–293 (1994).
- [36] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
- [37] Felix Graber, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning. In *Proceedings of the 2018 International Conference on Digital Health*. ACM, New York, NY, USA, 121-125.
- [38] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
- [39] Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).