ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương 10
Tối ưu đa mục tiêu

Huỳnh Thị Thanh Bình, Ban Hà Bằng, Phạm Quang Dũng, Nguyễn Khánh Phương, Đỗ Tuấn Anh

Nội dung

- **TỐI ƯU ĐA MỤC TIÊU**
  - **Bài toán đa mục tiêu**
  - **Hướng tiếp cận 1: Quy về đơn mục tiêu**
  - **Hướng tiếp cận 2: Pareto optimal**
- **NON-DOMINATED SORTING GENETIC ALGORITHM II**
  - **Giới thiệu**
  - **Chọn lọc trong NSGA II**
  - **Đánh giá**
- **A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION**
  - **Một số khái niệm**
  - **Cấu trúc thuật toán**
  - **Đánh giá**

Bài toán đa mục tiêu

Bài toán tối ưu đa mục tiêu (Multi-objective optimization problem):

- Bài toán yêu cầu tối ưu 2 hay nhiều hàm mục tiêu cùng lúc.
- Mô hình hóa:

$$\text{minimize } f(x) = \left(f_1(x), f_2(x), \dots, f_k(x)\right)$$
$$\text{s.t. } x \in X$$

  - (Giả sử các mục tiêu đều là cực tiểu hóa)
  - X là tập nghiệm chấp nhận được của bài toán
  - k hàm mục tiêu khác nhau: $f_i(x): X \longmapsto \mathbb{R}$

Bài toán đa mục tiêu

- Ví dụ:
  - Xây dựng hệ thống mạng:
    - Tối đa phạm vi phủ sóng
    - Tối thiểu chi phí triển khai
  - Lập kế hoạch đầu tư:
    - Tối đa lợi nhuận
    - Tối thiểu rủi ro…

Bài toán đa mục tiêu

- Trong bài toán tối ưu đa mục tiêu, các hàm mục tiêu thường xung đột lẫn nhau, do đó hiếm có một lời giải tối ưu với tất cả mục tiêu cùng lúc.
- Ví dụ: Cực tiểu hóa: $\begin{cases} f_1(x) = x^2 \\ f_2(x) = 9 - x \end{cases}, x \in \mathbb{R}$
  - $x = 3 \Rightarrow f_1(3) = 9; f_2(3) = 6$
  - $x = 5 \Rightarrow f_1(5) = 25; f_2(5) = 4$
  - $x = 3$ tốt hơn cho f_1, nhưng $x = 5$ tốt hơn cho f_2

Bài toán đa mục tiêu

Hướng tiếp cận giải bài toán đa mục tiêu:

- **Quy về đơn mục tiêu:** Đưa ra công thức ánh xạ nhiều mục tiêu về 1 mục tiêu, rồi giải như bài toán đơn mục tiêu.
- **Pareto optimal:** Dựa trên khái niệm tính trội và biên Pareto, tìm ra một số lời giải tốt với các hàm mục tiêu khác nhau, để một decision maker (ở đây có thể là con người) tự lựa chọn lời giải thích hợp nhất.

Hướng tiếp cận 1: Quy về đơn mục tiêu

Một số phương pháp quy về đơn mục tiêu:

- **Vector trọng số**
- **Tchebycheff**
- **Penalty-based boundary intersection (PBI)** (không giới thiệu)

Hướng tiếp cận 1: Quy về đơn mục tiêu

Vector trọng số

- Quy d mục tiêu về 1 mục tiêu
- Định nghĩa vector trọng số $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_d)$, sao cho $\sum_{i=1}^{d} \lambda_i = 1$
  - Trọng số 1 mục tiêu càng lớn thì mục tiêu đó càng được ưu tiên
- Mục tiêu mới:

$$g^{ws}(X|\lambda) = \sum_{i=1}^{d} \lambda_i f_j(X)$$

- Ví dụ: Hai mục tiêu: $f = (f_1, f_2) \Rightarrow f' = 0.3 \times f_1 + 0.7 \times f_2$

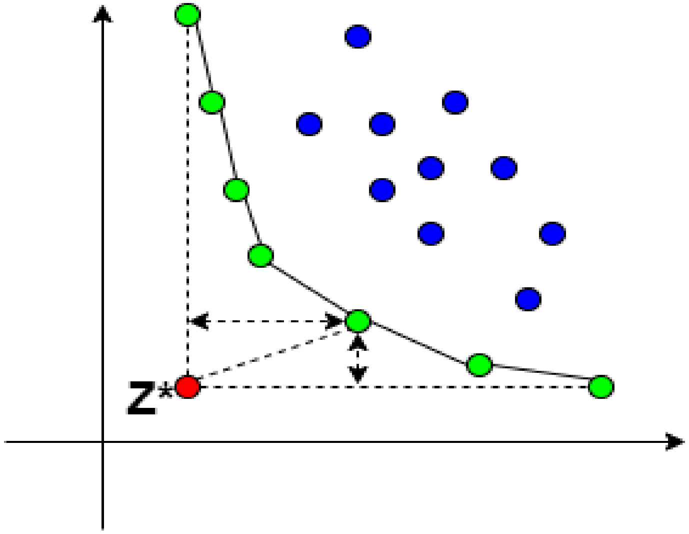
Hướng tiếp cận 1: Quy về đơn mục tiêu
Vector trọng số

Ví dụ: Cực tiểu hóa: $\begin{cases} f_1(x) = x^2 \\ f_2(x) = 9 - x \end{cases}, x \in \mathbb{R}$

- $g^{ws}(x) = \lambda_1 \times f_1(x) + \lambda_2 \times f_2(x)$
- Chọn $\lambda = (0.3, 0.7)$:
  - $x = 1 \Rightarrow g^{ws}(1) = 0.3 \times 1^2 + 0.7 \times (9 - 1) = 6.6$
  - $x = 2 \Rightarrow g^{ws}(2) = 0.3 \times 2^2 + 0.7 \times (9 - 2) = 6.1 < 6.6 = \mathrm{g}^{ws}(1)$
- Chọn $\lambda = (0.6, 0.4)$:
  - $x = 1 \Rightarrow g^{ws}(1) = 0.6 \times 1^2 + 0.4 \times (9 - 1) = 3.8$
  - $x = 2 \Rightarrow g^{ws}(2) = 0.6 \times 2^2 + 0.4 \times (9 - 2) = 5.2 > 3.8 = g^{ws}(1)$

Hướng tiếp cận 1: Quy về đơn mục tiêu Tchebycheff

- Vector trọng số $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_d)$
- Điểm tham chiếu $Z^* = (Z_1^*, Z_2^*, \dots, Z_d^*)$
  - Biên tốt nhất tìm được theo từng mục tiêu
  - Giả sử các mục tiêu đều là minimize, với mọi lời giải X tìm được:
    $$Z_i^* = \min\{f_i(X)\}$$
- Mục tiêu mới: Cực tiểu hóa:
  $$g^{te}(X|\lambda, Z^*) = \max_{1 \le i \le d}\{\lambda_i(\overline{f_i(X)} - \overline{Z_i^*})\}$$



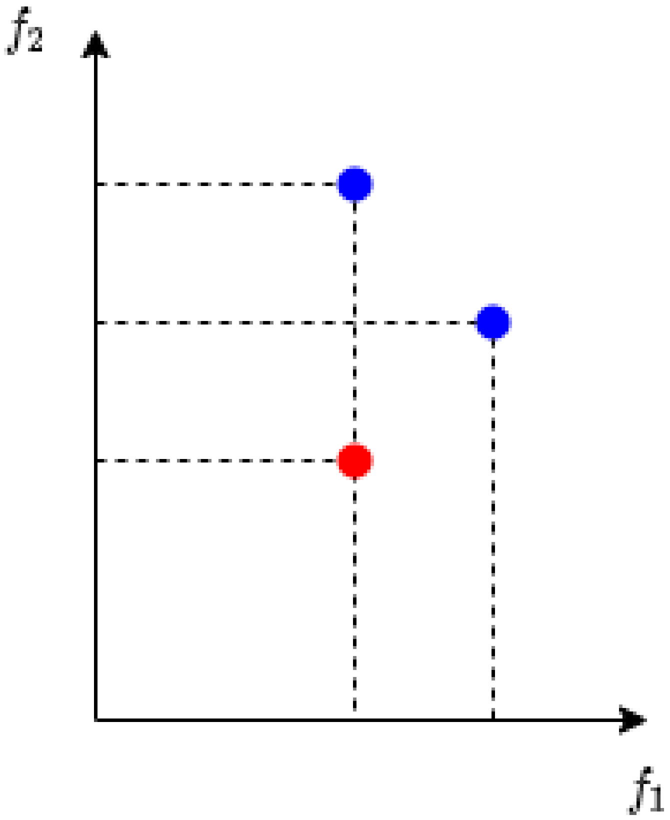
Hướng tiếp cận 2: Pareto optimal
Tính trội (Pareto dominance)

- Phương pháp so sánh 2 lời giải trong bài toán đa mục tiêu.
- Lời giải x_1 được gọi là trội hơn x_2 nếu:
  - x_1 không tệ hơn x_2 ở mọi mục tiêu tương ứng
  - x_1 tốt hơn x_2 ở ít nhất 1 mục tiêu
- Ví dụ trong bài toán cực tiểu hóa:

$$x_1\ dominate\ x_2 \Leftrightarrow \begin{cases} \forall i, f_i(x_1) \le f_i(x_2) \\ \exists j, f_j(x_1) < f_j(x_2) \end{cases}$$

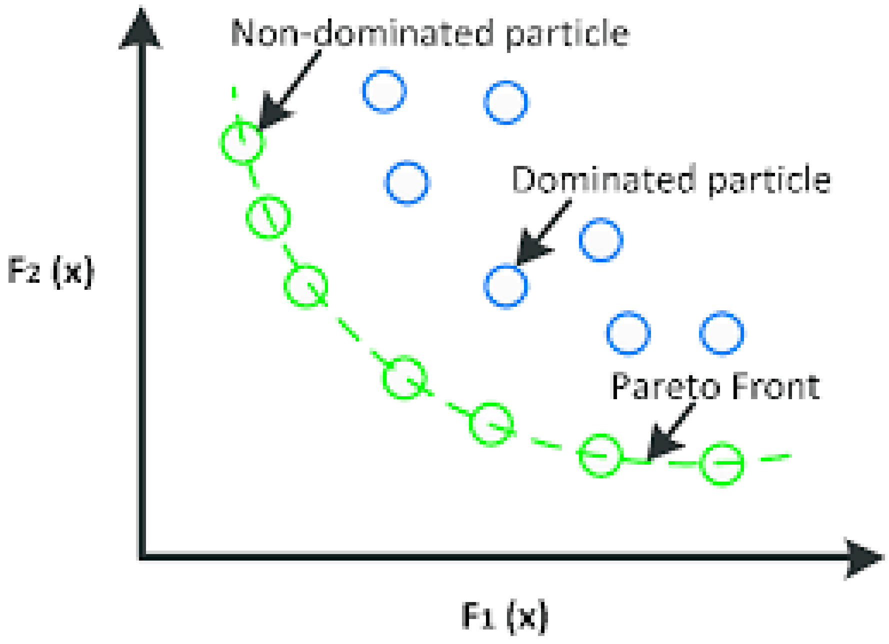
Hướng tiếp cận 2: Pareto optimal
Tính trội (Pareto dominance)

- Ví dụ trong bài toán cực tiểu hóa:
  - Cả 2 điểm màu xanh bị dominate bởi điểm màu đỏ
  - Giữa 2 điểm màu xanh, không có điểm nào dominate điểm nào



Hướng tiếp cận 2: Pareto optimal
Biên Pareto (Pareto front)

- Lời giải $x^* \in X$ được gọi là **Pareto optimal** nếu không lời giải nào trội hơn nó
- **Pareto front**: Tập các lời giải Pareto optimal
  - Ví dụ: Các điểm màu lục đều là Pareto optimal, hình thành 1 Pareto front



Hướng tiếp cận 2: Pareto optimal

- Các thuật toán áp dụng khái niệm Pareto optimal sẽ tập trung tìm kiếm biên Pareto tối ưu hoặc gần tối ưu cho bài toán
- Một số thuật toán tiến hóa điển hình:
  - Non-dominated Sorting Genetic Algorithm II (NSGA-II)
  - Strength Pareto Evolutionary Algorithm 2 (SPEA2)
  - A Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D)

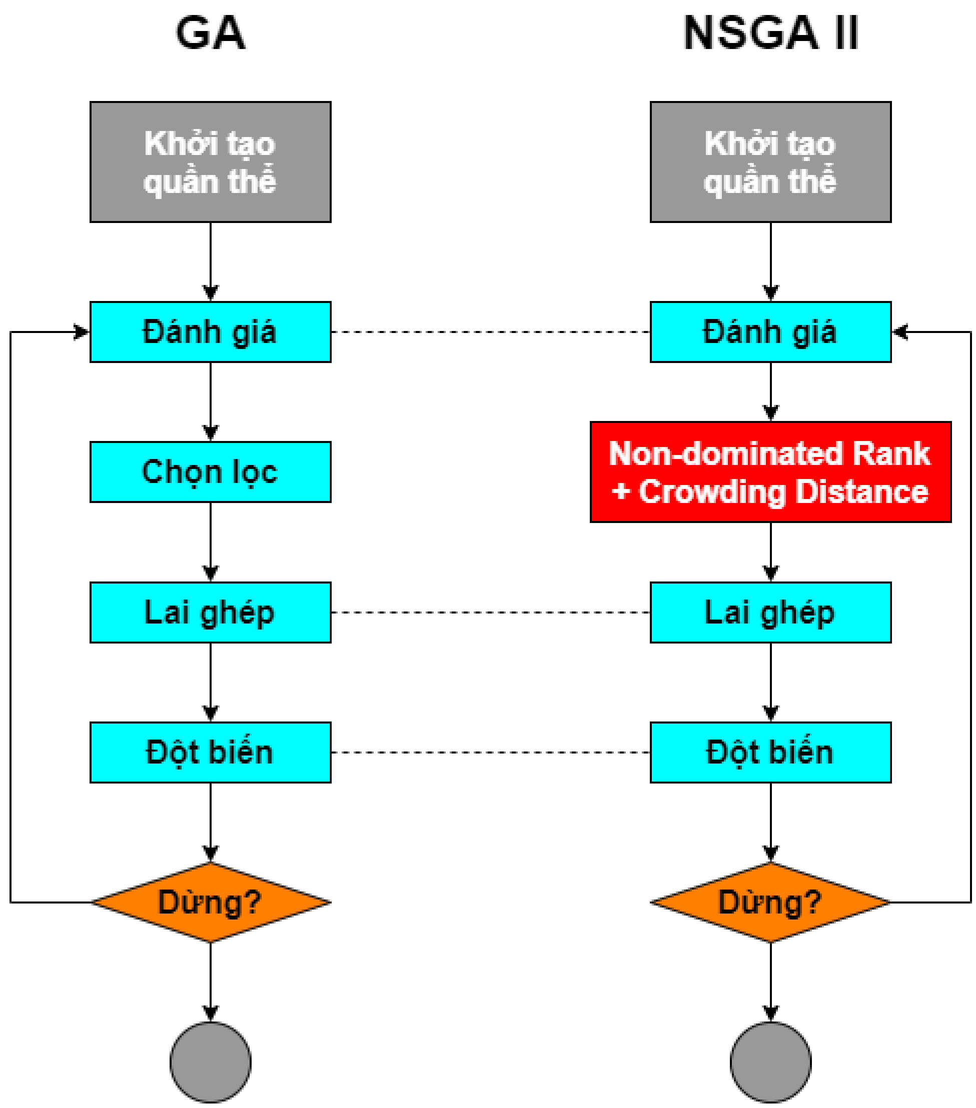
Giới thiệu

- Áp dụng giải thuật di truyền vào bài toán đa mục tiêu:
  - Đánh giá cá thể: Dựa trên hàm thích nghi => Giữ nguyên
  - Lai ghép + Đột biến: Dựa trên mã hóa => Giữ nguyên
  - Sắp xếp và chọn lọc cá thể => **???**
- Cần một giải pháp mới cho bước chọn lọc cá thể

Giới thiệu

- **Non-dominated Sorting Genetic Algorithm** (NSGA):
  - Giải thuật di truyền áp dụng sắp xếp không trội
  - Áp dụng cho bài toán đa mục tiêu
  - Chỉ khác GA thông thường ở bước chọn lọc
- **NSGA II**: Nâng cấp bước chọn lọc của NSGA:
  - Giảm độ phức tạp
  - Cải thiện độ đa dạng của quần thể

Giới thiệu


GA
NSGA II
Khởi tạo quần thể
Đánh giá
Chọn lọc
Lai ghép
Đột biến
Dừng?
Khởi tạo quần thể
Đánh giá
Non-dominated Rank + Crowding Distance
Lai ghép
Đột biến
Dừng?

Chọn lọc trong NSGA II

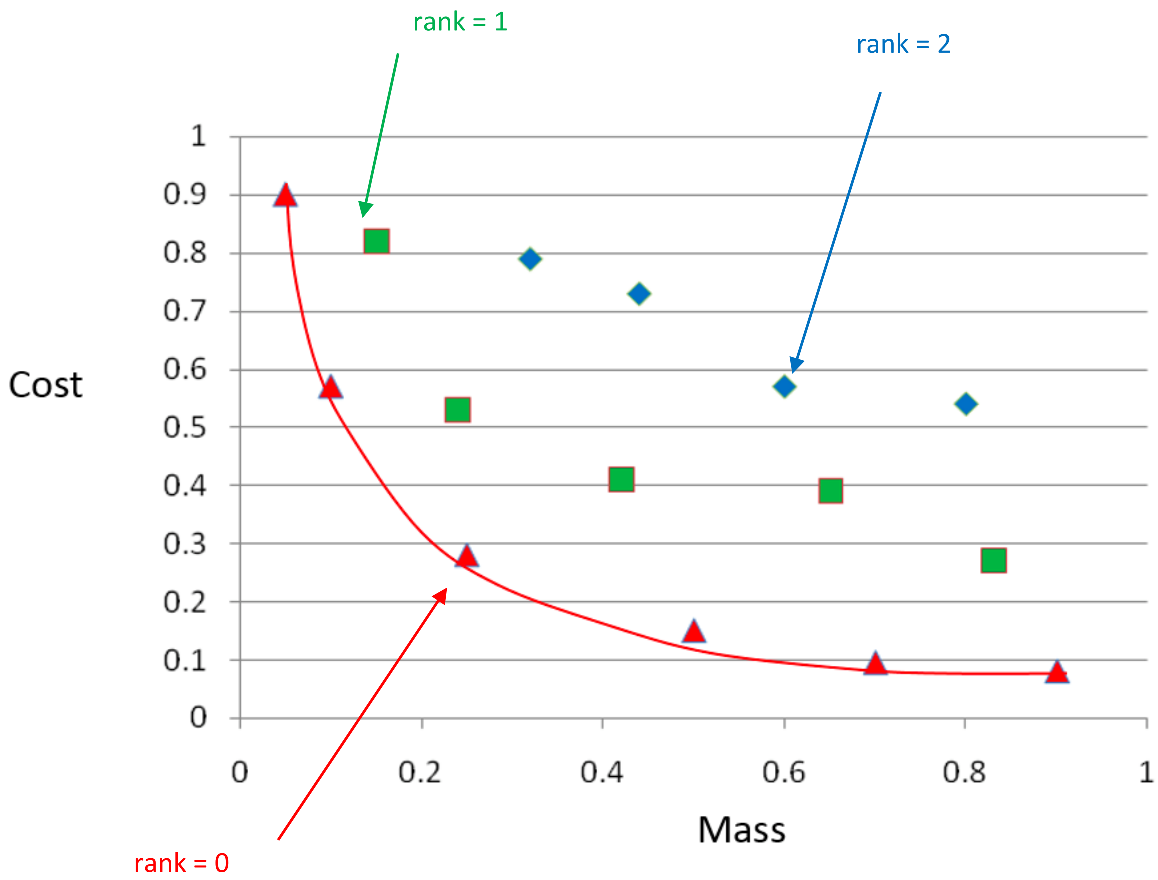
Các vấn đề ở bước chọn lọc cho bài toán đa mục tiêu:

- So sánh hai cá thể?
  - Pareto dominance → **Non-dominated Rank**
- Đảm bảo độ đa dạng của các cá thể được chọn?
  - **Crowding distance**
- Tốc độ tính toán?
  - Tốt hơn NSGA

Chọn lọc trong NSGA II
Non-dominated Rank

- Dựa trên khái niệm Pareto optimal
- "Phân lớp" quần thể thành các biên Pareto:
  - Biên thứ nhất ($rank = 0$): Các cá thể không bị cá thể nào khác trội
  - Biên thứ hai ($rank = 1$): Các cá thể chỉ bị trội bởi cá thể có $rank \leq 0$
  - Biên thứ ba…: Tương tự
- Non-dominated Rank của cá thể chính là $rank$ của biên Pareto chứa nó
  - Non-dominated rank càng nhỏ càng tốt

Chọn lọc trong NSGA II
Non-dominated Rank



Chọn lọc trong NSGA II
Non-dominated Rank

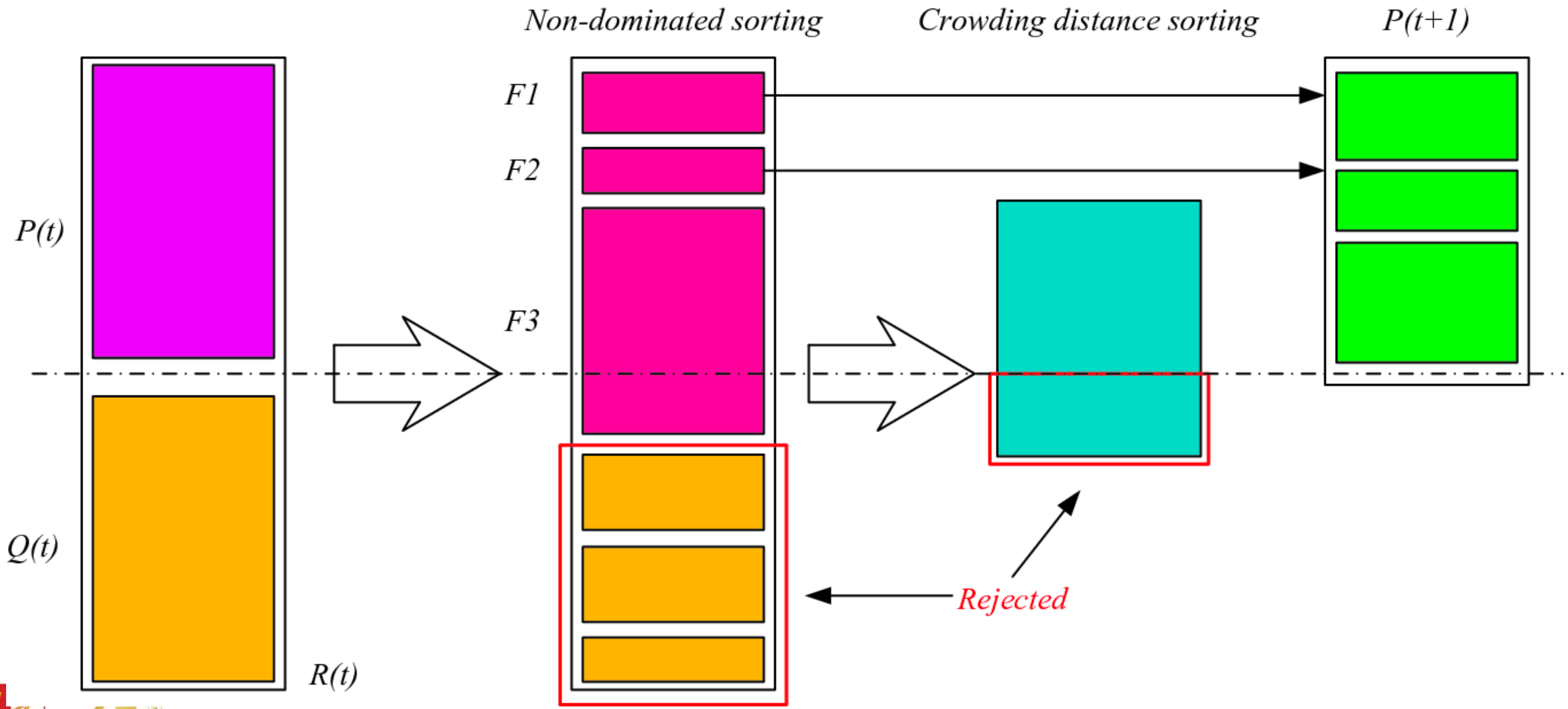
- Cài đặt: Thuật toán ngây thơ:
  - Kiểm tra 1 cá thể có bị trội không → $O(MN)$
  - Tìm biên Pareto đầu tiên → $O(MN^2)$
  - Phân lớp Pareto cho N cá thể → $O(MN^3)$
- Cải tiến: Fast Non-dominated Sorting Algorithm
  - Sử dụng mảng nhớ để lưu kết quả so sánh các cá thể:
    - n_p: Số cá thể trội hơn cá thể p trong quần thể
    - S_p: Tập các cá thể bị p trội
  - Phân lớp Pareto dựa trên các dữ kiện đã lưu
  - Độ phức tạp: $O(MN^2)$

Chọn lọc trong NSGA II
Non-dominated Rank

- Chọn lọc bằng Non-dominated rank:
  - Lấy lần lượt các cá thể theo $rank$ từ nhỏ đến lớn
  - Lấy đến khi vừa đủ N cá thể (N là hằng số cho trước)
- Vấn đề: Biên Pareto cuối bị lẻ?
  - Không lấy biên cuối thì không đủ N, mà lấy hết thì nhiều hơn N
  - Giải pháp: **Crowding Distance**

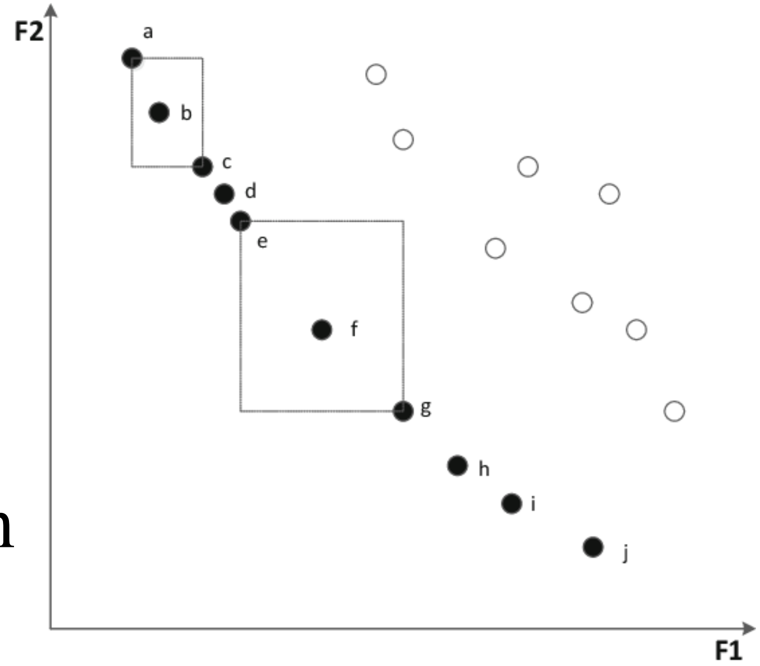
Chọn lọc trong NSGA II

Minh họa quá trình chọn lọc của NSGA II:



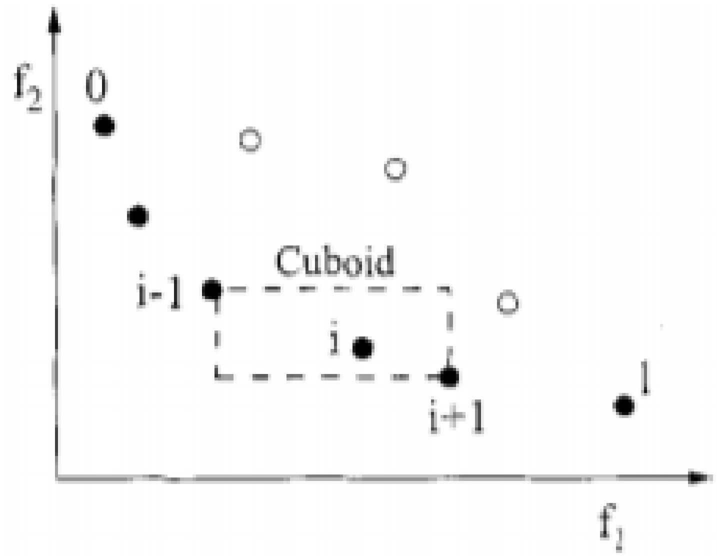
Chọn lọc trong NSGA II
Crowding Distance

- Nhìn hình bên: Nếu chỉ được chọn cá thể b hoặc f, cá thể nào sẽ tốt hơn?
  - f tốt hơn vì: f ở xa các cá thể khác, trong khi các cá thể ở gần b có thể thay thế cho b => chọn f tăng độ đa dạng quần thể
- Crowding Distance: là độ đo đánh giá tầm ảnh hưởng của cá thể đối với mức độ đa dạng của quần thể được chọn



Chọn lọc trong NSGA II
Crowding Distance

- Crowding Distance của điểm i: Tổng các khoảng cách của từng cặp 2 điểm gần i nhất theo từng chiều (từng hàm mục tiêu)
- Crowding Distance càng lớn càng tốt



$$distance[i] = \begin{cases} \infty & \text{if } f_m(i) = f_m^{max} | f_m^{min} \\ \sum_{m \in M} \frac{f_m(i+1) - f_m(i-1)}{f_m^{max} - f_m^{min}} & \text{if other cases} \end{cases}$$

Chọn lọc trong NSGA II

Bước chọn lọc trong NSGA II:

- Bước 1: Phân lớp quần thể thành các biên Pareto (Non-dominated Rank)
- Bước 2: Lấy lần lượt cá thể theo $rank$ từ nhỏ đến lớn (đến khi đủ N hoặc nếu lấy $rank$ tiếp theo sẽ vượt quá N)
- Bước 3: Nếu bước 2 chưa đủ N cá thể, tính Crowding Distance của các cá thể $rank$ tiếp rồi lấy từ to đến nhỏ tới khi đủ N

Đánh giá

- Ưu điểm:
  - Cấu trúc gần với GA truyền thống, dễ hiểu, dễ áp dụng
  - Độ phức tạp nhỏ, tốc độ chạy nhanh
  - Kết quả tốt với đa số bài toán đa mục tiêu
- Nhược điểm:
  - Không hiệu quả với một số rất nhỏ bài toán đặc thù (rời rạc, ràng buộc hay hàm mục tiêu phức tạp,…)

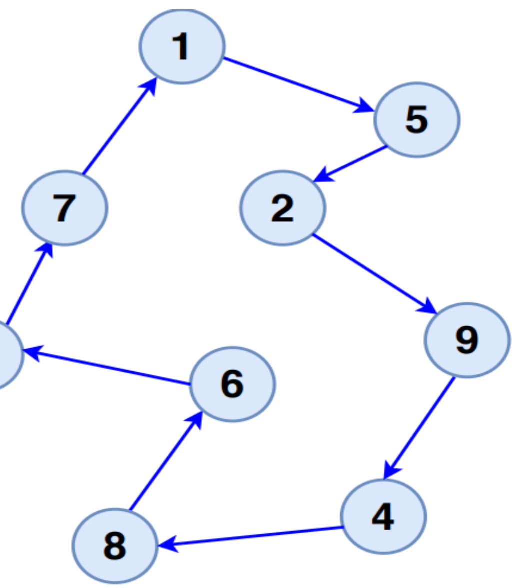
Ví dụ giải bài toán TSP đa mục tiêu
Phát biểu bài toán

- Đầu vào:
  - Đồ thị đầy đủ K_n với n là số thành phố
  - Ma trận khoảng cách $D = \{\mathrm{d_{ij}}\}_{\mathrm{n}\times n}$
  - Ma trận thời gian $T = \{t_{ij}\}_{n\times n}$
- Đầu ra: Chu trình độ dài n đi qua cả n thành phố và quay về thành phố xuất phát
- Mục tiêu:
  - Cực tiểu hóa tổng khoảng cách fd
  - Cực tiểu hóa tổng thời gian ft

Ví dụ giải bài toán TSP đa mục tiêu
Mã hóa cá thể

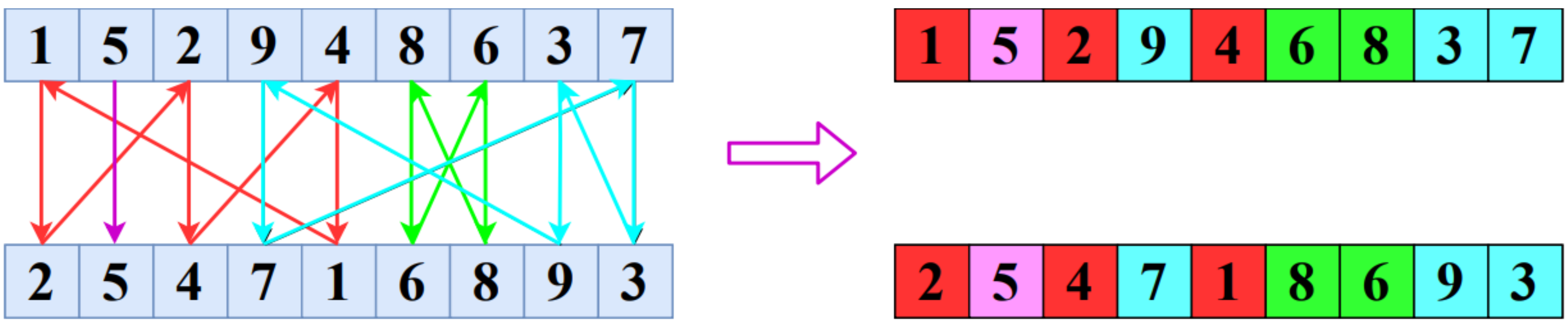
- Biểu diễn cá thể: Mã hóa hoán vị. Mỗi cá thể là 1 hoán vị n phần tử.

1	5	2	9	4	8	6	3	7



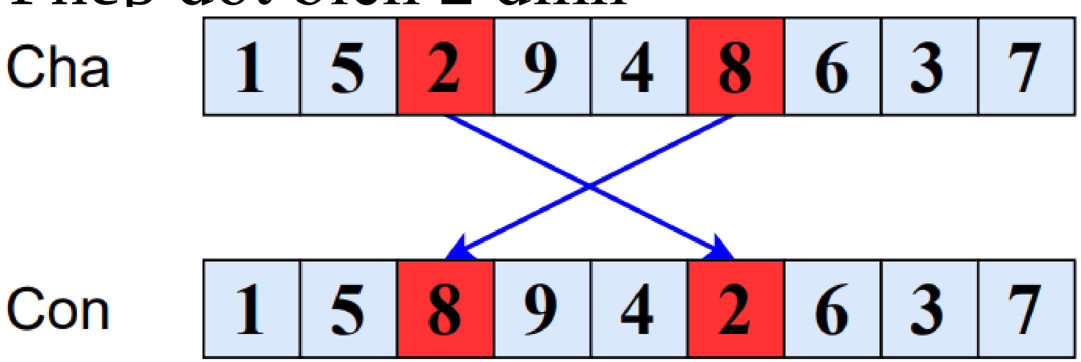
Ví dụ giải bài toán TSP đa mục tiêu
Lai ghép và đột biến

- Lai ghép: Phép lai ghép chu trình



(a) (b)

- Đột biến: Phép đột biến 2 đỉnh



Ví dụ giải bài toán TSP đa mục tiêu
Chọn lọc cá thể

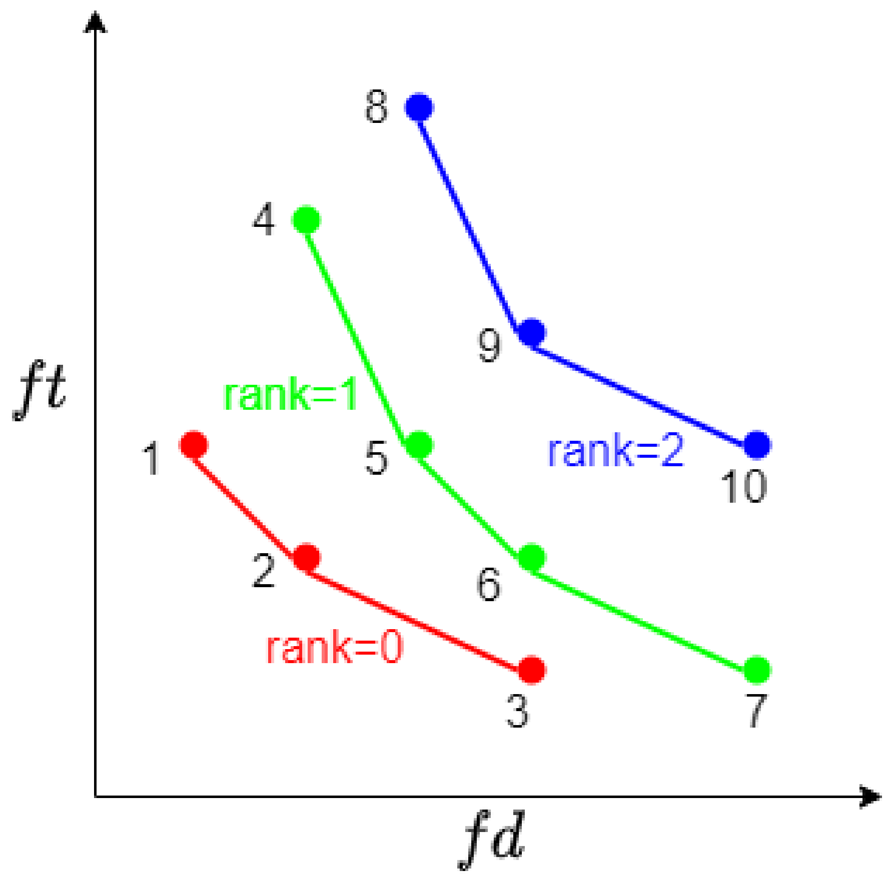
- Yêu cầu:
  - Bảng các cá thể i thu được sau lai ghép và đột biến (bao gồm quần thể ban đầu):
    - fd và ft là giá trị 2 hàm mục tiêu (khoảng cách và thời gian)

Cá thể i	fd_i	ft_i	Cá thể i	fd_i	ft_i
1	1	3	6	4	2
2	2	2	7	6	1
3	4	1	8	3	6
4	2	5	9	4	4
5	3	3	10	6	3

  - Cần chọn ra 5 cá thể

Ví dụ giải bài toán TSP đa mục tiêu
Chọn lọc cá thể

- Trình tự thực hiện:
  - Bước 1: Phân lớp quần thể bằng Non-dominated Rank (như hình)
  - Bước 2: Chọn 3 cá thể $rank = 0$ (vì chọn đến $rank = 1$ sẽ vượt quá 5 cá thể)
  - Bước 3: Tính Crowding Distance của 4 cá thể $rank = 1$. Chọn cá thể số 4 và 7 vì có $distance = \infty$
- Kết quả: Quần thể mới gồm 5 cá thể số 1, 2, 3, 4 và 7



Một số khái niệm

- **Phân hoạch**
- **Hàng xóm**
- **A multi-objective evolutionary algorithm based on decomposition**

Một số khái niệm

Phân hoạch (Decomposition):

- Nhắc lại: Phương pháp vector trọng số → Quy bài toán đa mục tiêu về đơn mục tiêu
- Cho N vector λ khác nhau: 1 bài toán đa mục tiêu → N bài toán đơn mục tiêu khác nhau → N lời giải tối ưu khác nhau
- Phân hoạch: Phương pháp quy bài toán đa mục tiêu thành N bài toán đơn mục tiêu khác nhau, kết hợp giải N bài toán con này để được tập lời giải tạo thành biên Pareto của bài toán gốc

Một số khái niệm

Phân hoạch (Decomposition):

- Ví dụ phân hoạch bài toán 2 mục tiêu:

Bài toán gốc:
$f = (f_1, f_2)$

Bài toán con 1:
$g_1^{ws} = 0.1 \times f_1 + 0.9 \times f_2$

Bài toán con 2:
$g_2^{ws} = 0.2 \times f_1 + 0.8 \times f_2$

...

Bài toán con 9:
$g_9^{ws} = 0.9 \times f_1 + 0.1 \times f_2$

Một số khái niệm

Hàng xóm:

- Trong phân hoạch ở trên, một số bài toán con có hàm mục tiêu (hay vector λ) gần nhau hơn các bài toán khác
  - Ví dụ: Bài toán gốc có 2 mục tiêu, bài toán con đang xét có $\lambda_1 = (0.3, 0.7)$. Có 2 bài toán con khác: $\lambda_2 = (0.4, 0.6)$ và $\lambda_3 = (0.7, 0.3)$. Dễ thấy bài toán λ_1 gần với λ_2 hơn.
- Định nghĩa: "Hàng xóm" của 1 bài toán con là T bài toán con gần với nó nhất trong N bài toán con (bao gồm chính nó)
  - T là hằng số cho trước ($1 \leq T \leq N$)
  - Khoảng cách 2 bài toán con là khoảng cách Euclid giữa 2 vector trọng số tương ứng
  - Ký hiệu: Hàng xóm của bài toán con i: $B^i = \{b_1, b_2, \ldots, b_T\}$

Một số khái niệm

Hàng xóm:

- Ví dụ xây dựng tập hàng xóm cho bài toán con 2:
  - $T = 3$: Mỗi bài toán con có 3 hàng xóm
  - 3 bài toán con gần nhất là 1, 2, 3 (bao gồm chính nó)
  - Hàng xóm của 2: $B^2 = \{1, 2, 3\}$

Hàng xóm?

Bài toán con 1:	Bài toán con 2:	Bài toán con 3:
$g_1^{ws} = 0.1 \times f_1 + 0.9 \times f_2$	$g_2^{ws} = 0.2 \times f_1 + 0.8 \times f_2$	$g_3^{ws} = 0.3 \times f_1 + 0.7 \times f_2$

Một số khái niệm

A multi-objective evolutionary algorithm based on decomposition (MOEA/D)

- Giải thuật tiến hóa đa mục tiêu dựa trên phân hoạch
- Ý tưởng:
  - Phân hoạch bài toán đa mục tiêu thành N bài toán đơn mục tiêu con $P^1, \dots, P^N$
    - Tạo N vector trọng số khác nhau
  - Duy trì 1 lời giải tốt nhất cho từng bài toán con
  - Tối ưu lời giải bài toán con bằng cách kết hợp lời giải của các "hàng xóm" (các bài toán gần nhau thì thường có lời giải tốt gần nhau)
- Đánh giá lời giải:
  - 1 trong các phương pháp quy về đơn mục tiêu đã nêu
  - Trong slide này dùng phương pháp Tchebycheff

Cấu trúc thuật toán

Thuật toán MOEA/D gồm 3 bước chính:

- **Bước 1: Khởi tạo**
- **Bước 2: Tiến hóa theo thế hệ**
  - Lặp lại nhiều lần tới khi thỏa mãn điều kiện dừng
- **Bước 3: Điều kiện dừng**

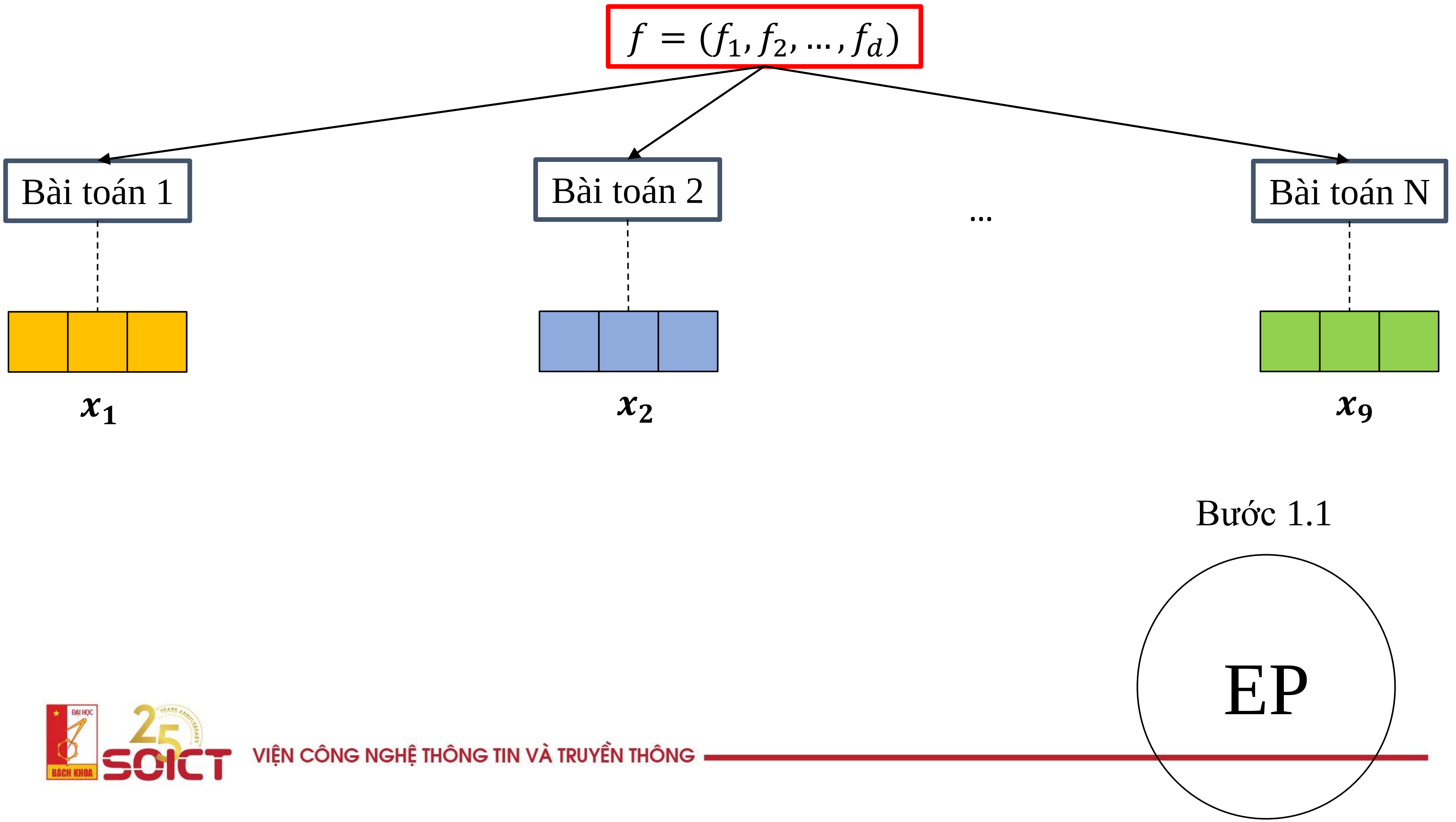
Cấu trúc thuật toán
Đầu vào

- Các tham số:
  - N vector trọng số
  - T: Số hàng xóm của 1 bài toán con
- Các đối tượng được duy trì và cập nhật qua từng thế hệ:
  - Quần thể: N lời giải của N bài toán con
  - Điểm tham chiếu Z^*: Dùng cho phương pháp đánh giá Tchebycheff
    - Không cần thiết nếu không dùng Tchebycheff
  - Quần thể ngoài EP: Biên Pareto của tất cả lời giải tìm được tính đến thế hệ hiện tại
    - Không cần thiết nếu trực tiếp dùng quần thể làm đầu ra

Cấu trúc thuật toán
Bước 1: Khởi tạo

- Bước 1.1: Đặt quần thể ngoài $EP = \emptyset$
- Bước 1.2: Xây dựng tập hàng xóm:
  - $B^i = \{b_1, b_2, \dots, b_T\}$: Tập hàng xóm của bài toán thứ i
  - $\lambda^{b_1}, \lambda^{b_2}, \dots, \lambda^{b_T}$ - T vector trọng số gần với vector λ^i nhất
- Bước 1.3: Khởi tạo ngẫu nhiên quần thể: $\{x^1, \dots, x^N\}$
- Bước 1.4: Khởi tạo điểm tham chiếu $Z^* = (Z_1^*, Z_2^*, \dots, Z_d^*)$:
  - $Z_k^* = \min_{1 \le i \le N} \{f_k(x^i)\}$

Cấu trúc thuật toán
Bước 1: Khởi tạo



Cấu trúc thuật toán
Bước 2: Tiến hóa theo thế hệ (1)

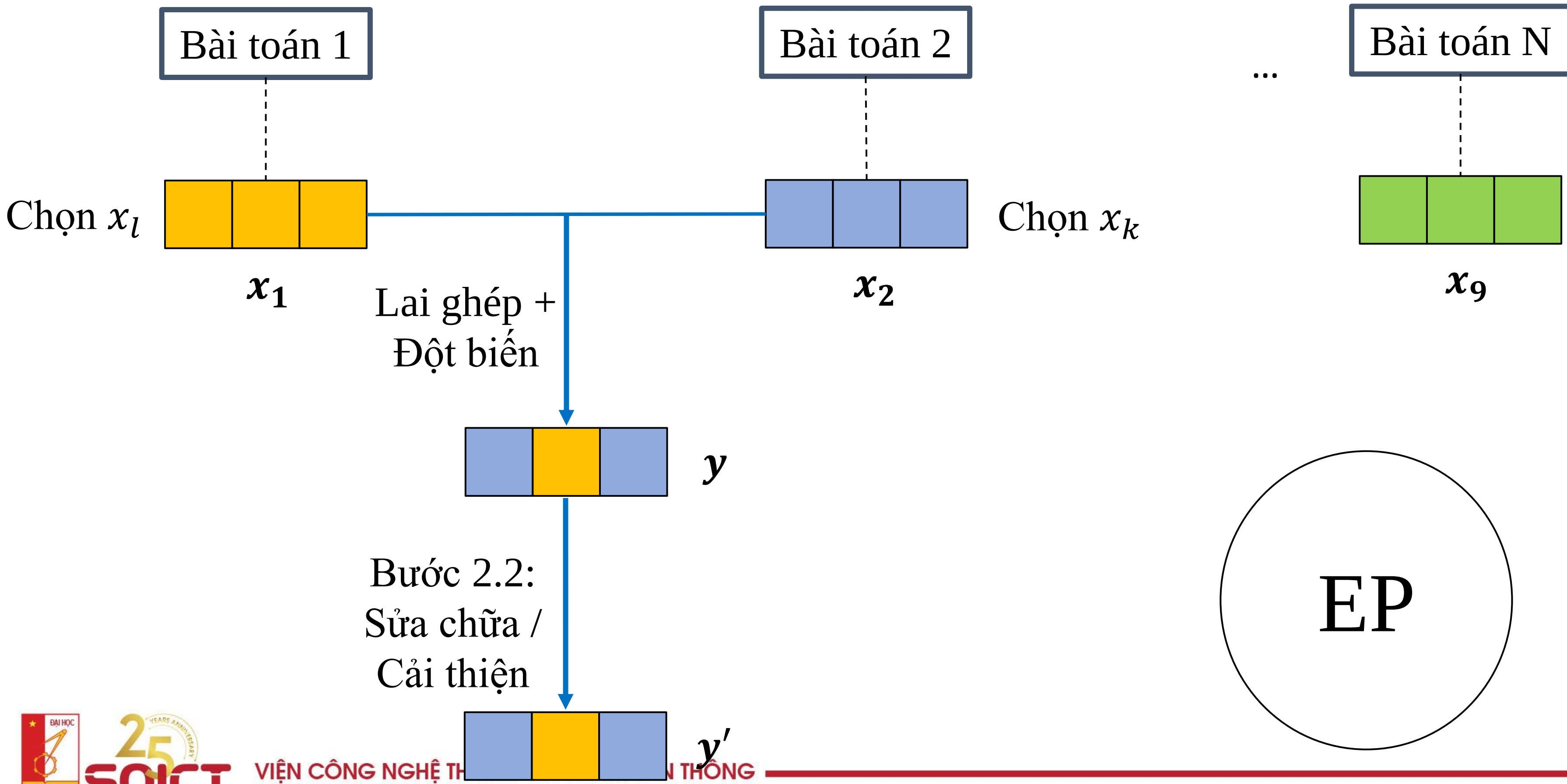
Với mỗi bài toán con P^i:

- Bước 2.1: Tạo lời giải mới
  - Chọn ngẫu nhiên 2 lời giải hàng xóm x^k, x^l của P^i $(k, l \in B^i)$
  - Lai ghép: $x^k \oplus x^l$ → lời giải mới y
  - Đột biến y → lời giải mới y'
- Bước 2.2: Sửa chữa/Cải thiện lời giải
  - Nếu y' không là lời giải hợp lệ, cần sửa chữa hoặc loại bỏ
  - Cải thiện y' nếu có thể (tùy bài toán)

Cấu trúc thuật toán
Bước 2: Tiến hóa theo thế hệ (1)

Bước 2.1: Tạo lời giải mới

Xét bài toán 2



Cấu trúc thuật toán
Bước 2: Tiến hóa theo thế hệ (2)

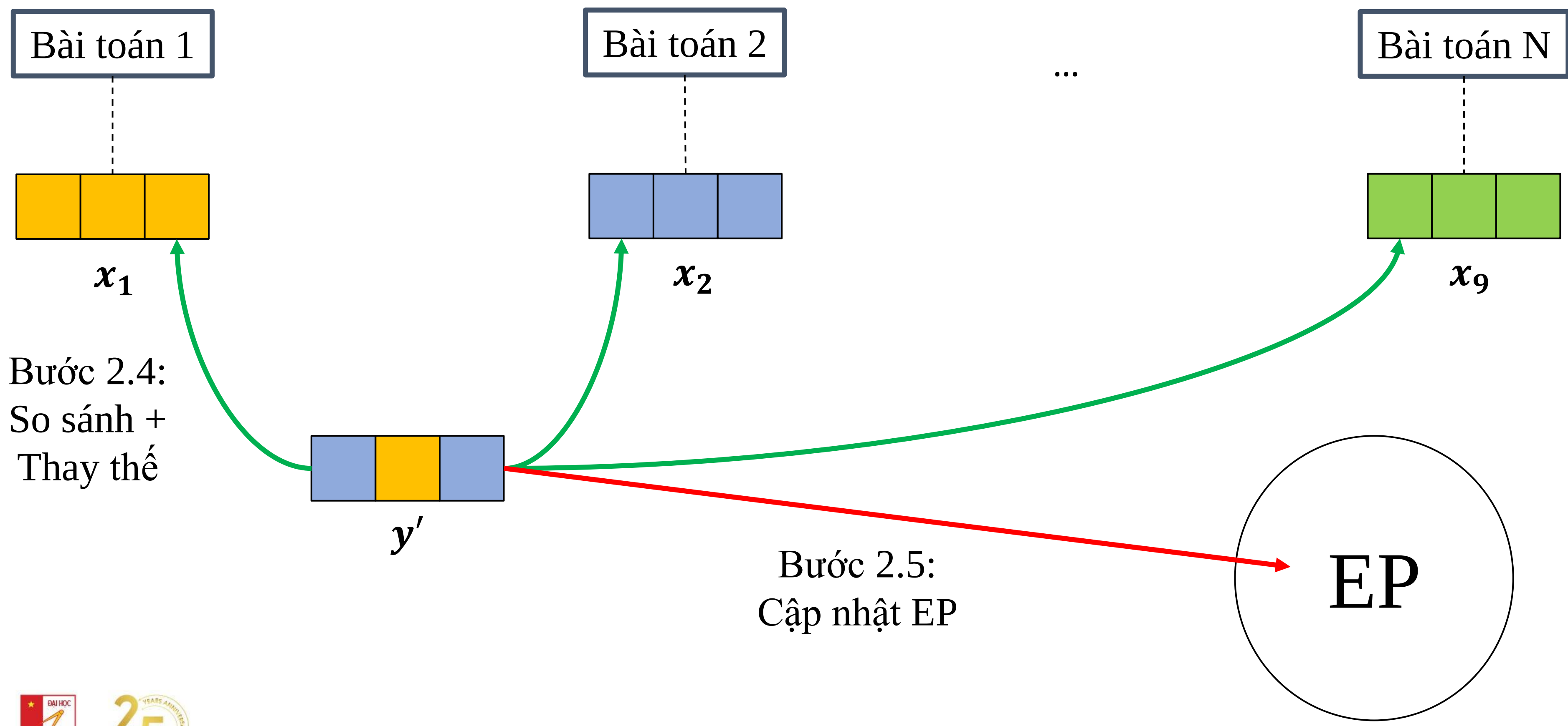
Với mỗi bài toán con P^i:

- Bước 2.3: Cập nhật Z^*:
  - $Z_k^* = \min\left(Z_k^*, f_k(y')\right) \forall k = \overline{1, d}$
- Bước 2.4: Cập nhật các hàng xóm:
  - Với mọi $b \in B^i$:

  →nếu $g^{te}(y'|\lambda^b, Z^*) \leq g^{te}(x^b|\lambda^b, Z^*)$ (theo Tchebycheff)

  →thay x^b bằng y'
- Bước 2.5: Cập nhật EP:
  - Loại mọi lời giải bị trội bởi y'
  - Thêm y' vào EP nếu nó không bị trội bởi lời giải nào

Cấu trúc thuật toán
Bước 2: Tiến hóa theo thế hệ (2)



Cấu trúc thuật toán
Bước 3: Điều kiện dừng

- Điều kiện dừng: Tương tự các thuật toán GA thông thường
  - Số thế hệ tối đa
  - Số thế hệ tối đa mà EP không được cập nhật
  - …
- Đầu ra: EP (Biên Pareto)

Đánh giá

- Ưu điểm:
  - Nhanh, độ phức tạp tương đương GA thông thường
  - Biên Pareto đều
  - Khả năng duy trì cân bằng các hàm mục tiêu (ví dụ: tránh hội tụ sớm khi áp dụng tìm kiếm cục bộ quá mạnh lên 1 mục tiêu)
- Nhược điểm:
  - Hiệu quả phụ thuộc lớn vào tham số người dùng tự đặt (đặc biệt là cách chia vector trọng số)
  - Không hiệu quả bằng các thuật toán như NSGA-II trong nhiều bài toán thông thường (khả năng tối ưu các mục tiêu tương đối đồng đều)

Ví dụ giải bài toán TSP đa mục tiêu

Phát biểu bài toán

- Đầu vào:
  - Đồ thị đầy đủ K_n với n là số thành phố
  - Ma trận khoảng cách $D = \{\mathrm{d_{ij}}\}_{\mathrm{n}\times n}$
  - Ma trận thời gian $T = \{t_{ij}\}_{n\times n}$
- Đầu ra: Chu trình độ dài n đi qua cả n thành phố và quay về thành phố xuất phát
- Mục tiêu:
  - Cực tiểu hóa tổng khoảng cách fd
  - Cực tiểu hóa tổng thời gian ft

Ví dụ giải bài toán TSP đa mục tiêu

Phát biểu bài toán

- Đầu vào:
  - Đồ thị đầy đủ K_n với n là số thành phố
  - Ma trận khoảng cách $D = \{\mathrm{d_{ij}}\}_{\mathrm{n}\times n}$
  - Ma trận thời gian $T = \{t_{ij}\}_{n\times n}$
- Đầu ra: Chu trình độ dài n đi qua cả n thành phố và quay về thành phố xuất phát
- Mục tiêu:
  - Cực tiểu hóa tổng khoảng cách fd
  - Cực tiểu hóa tổng thời gian ft

Ví dụ giải bài toán TSP đa mục tiêu
Tham số cài đặt

- Số bài toán con (kích thước quần thể): $N = 5$
  - Tập vector trọng số: $\lambda^1 = (0.0,\ 1.0),\ \lambda^2 = (0.25, 0.75),\ \lambda^3 = (0.5, 0.5),\ \lambda^4 = (0.75, 0.25),\ \lambda^5 = (1.0, 0.0)$
- Kích thước hàng xóm: $K = 3$

Ví dụ giải bài toán TSP đa mục tiêu
Bước 1: Khởi tạo

- Bước 1.1: $EP = \emptyset$
- Bước 1.2: Xây dựng tập hàng xóm

Vector	Giá trị	Tập hàng xóm
λ^1	0.0, 1.0	$B^1 = \{\lambda^1, \lambda^2, \lambda^3\}$
λ^2	0.25, 0.75	$B^2 = \{\lambda^1, \lambda^2, \lambda^3\}$
λ^3	0.5, 0.5	$B^3 = \{\lambda^2, \lambda^3, \lambda^4\}$
λ^4	0.75, 0.25	$B^4 = \{\lambda^3, \lambda^4, \lambda^5\}$
λ^5	1.0, 0.0	$B^5 = \{\lambda^3, \lambda^4, \lambda^5\}$

Ví dụ giải bài toán TSP đa mục tiêu
Bước 1: Khởi tạo

- Bước 1.3: Khởi tạo ngẫu nhiên quần thể

Cá thể i	Bài toán con	fd_i	ft_i
x^1	P^1	3	6
x^2	P^2	3	5
x^3	P^3	**2**	3
x^4	P^4	4	**1**
x^5	P^5	3	2

- Bước 1.4: Khởi tạo điểm tham chiếu:

$$Z^* = (2, 1)$$

Ví dụ giải bài toán TSP đa mục tiêu
Bước 2: Tìm kiếm theo thế hệ

Xét bài toán con P^1:

- Bước 2.1: Tạo lời giải mới
  - Tập hàng xóm: $B^1 = \{P^1, P^2, P^3\}$ → Lấy ngẫu nhiên 2 lời giải x^2, x^3
  - Lai ghép và đột biến: $x^2 \oplus x^3 \Rightarrow y \Rightarrow y'$
    - Đánh giá y' được: $fd(y') = 1,\ ft(y') = 5$
- Bước 2.2: Sửa chữa y'
  - Trong ví dụ này y' luôn hợp lệ, không cần sửa chữa
- Bước 2.3: Cập nhật Z^*
  - Hiện tại: $Z^* = (2,1)$; $f(y') = \left(fd(y'), ft(y')\right) = (1,5)$
  - Cập nhật: $Z^* = (1,1)$

Ví dụ giải bài toán TSP đa mục tiêu
Bước 2: Tìm kiếm theo thế hệ

Xét bài toán con P^1:

- Bước 2.4: Cập nhật các hàng xóm (trong tập B^1)
  - Điểm tham chiếu: $Z^* = (1, 1)$
  - Cá thể mới y': $f(y') = (1, 5)$

Cá thể	Vector trọng số	Hàm mục tiêu	g_i^{te}	$g^{te}(y')$	Thay thế?
x^1	$\lambda^1 = (0.0, 1.0)$	$(3, 6)$	5	4	Có
x^2	$\lambda^2 = (0.25, 0.75)$	$(3, 5)$	3.5	3	Có
x^3	$\lambda^3 = (0.5, 0.5)$	$(2, 3)$	1.5	2	Không

- Bước 2.5: Cập nhật EP
  - Hiện tại: $EP = \emptyset$ ➔ Thêm y' vào EP

Ví dụ giải bài toán TSP đa mục tiêu
Bước 3: Điều kiện dừng

- Lặp lại quá trình trên với cả N bài toán con, trong từng thế hệ
- Dừng thuật toán khi điều kiện dừng thỏa mãn
- Trả về tập lời giải trong EP