Mutilobjective Evolutionary Optimization

TaskOverflow

Ngày 27 tháng 12 năm 2018

INTRODUTION

Point 1

Thế nào là bài toán tối ưu đa mục tiêu?

Các định nghĩa cơ bản

Không gian nghiệm

Với các điều kiện rằng buộc của x sẽ xác định một không gian $\Omega \subset R^n$

Các định nghĩa cơ bản

Không gian nghiệm

Với các điều kiện rằng buộc của x sẽ xác định một không gian $\Omega \subset R^n$

Hàm mục tiêu

$$y = f(x) = (f_1(x), ..., f_m(x))$$
$$x \in \Omega$$

Các định nghĩa cơ bản

Không gian nghiệm

Với các điều kiện rằng buộc của x sẽ xác định một không gian $\Omega \subset R^n$

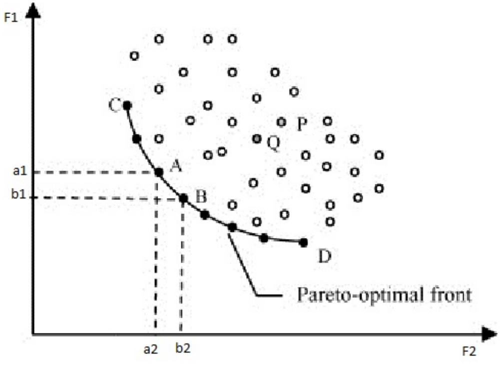
Hàm mục tiêu

$y = f(x) = (f_1(x), ..., f_m(x))$
$x \in \Omega$

Không gian mục tiêu

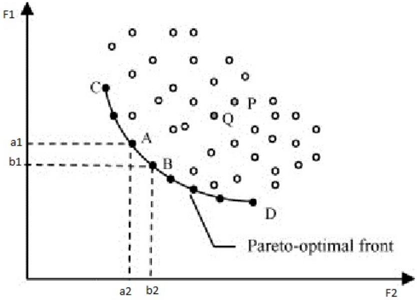
$f : \Omega \rightarrow \Theta \subset R^m$ xác định không gian mục tiêu Θ

Hình ảnh minh họa



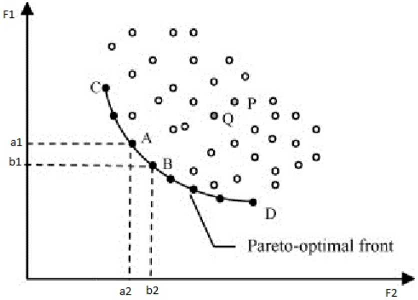
Hình: Nguồn: Semantic Scholar

Các định nghĩa cơ bản



Hình: Hình ảnh minh họa

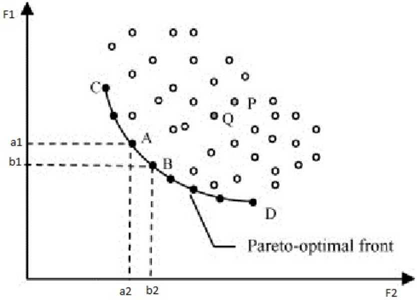
Các định nghĩa cơ bản



Hình: Hình ảnh minh họa

Cho 2 nghiệm $x, y \in \Omega$, x trội hơn y nếu như $\forall i \in 1, 2, ..., m, f_i(x) \leq f_i(y)$ và $\exists j \in 1, 2, ..., m, f_j(x) < f_j(y)$.

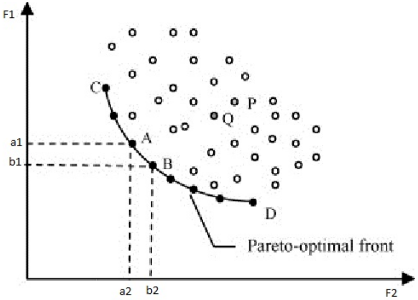
Các định nghĩa cơ bản



Hình: Hình ảnh minh họa

Một nghiệm $x^* \in \Omega$ là nghiệm tối ưu Pareto nếu không tồn tại $x \in \Omega$ trội hơn nó.

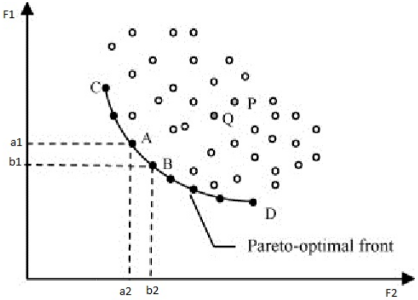
Các định nghĩa cơ bản



Hình: Hình ảnh minh họa

Tập hợp nghiệm tối ưu Pareto được gọi là tập Pareto.

Các định nghĩa cơ bản



Hình: Hình ảnh minh họa

Các vector mục tiêu tương ứng của tập Pareto được gọi là Biên Pareto (Pareto Frontier - PF).

Mục tiêu tối ưu

Mục tiêu tối ưu là đạt được một tập xấp xỉ S với tập biên Pareto thỏa mãn:

Mục tiêu tối ưu

Mục tiêu tối ưu là đạt được một tập xấp xỉ S với tập biên Pareto thỏa mãn:

Độ hội tụ

Mọi nghiệm thuộc S là gần nhất có thể với P.

Mục tiêu tối ưu

Mục tiêu tối ưu là đạt được một tập xấp xỉ S với tập biên Pareto thỏa mãn:

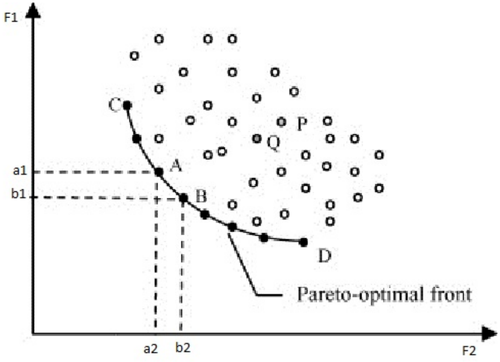
Độ hội tụ

Mọi nghiệm thuộc S là gần nhất có thể với P.

Độ đa dạng

Mọi nghiệm thuộc S phân bố đa dạng trên không gian mục tiêu

Hình ảnh minh họa



Hình: Nguồn: Science Direct

Thuật toán NSGA-II

- Sau lần chạy thứ t, duy trì một quần thể solution P_t có kích thước N cố định qua mọi lần chạy.

Thuật toán NSGA-II

- Sau lần chạy thứ t, duy trì một quần thể solution P_t có kích thước N cố định qua mọi lần chạy.
- Trong lần chạy thứ $t+1$:
  1. Sử dụng các toán tử selection, crossover và mutation để tạo ra quần thể con cháu Q_t từ P_t.
  2. Chọn ra N phần tử tốt nhất từ $P_t \cup Q_t$ để tạo ra P_{t+1}.

Thuật toán NSGA-II

- Sau lần chạy thứ t, duy trì một quần thể solution P_t có kích thước N cố định qua mọi lần chạy.
- Trong lần chạy thứ $t+1$:
  1. Sử dụng các toán tử selection, crossover và mutation để tạo ra quần thể con cháu Q_t từ P_t.
  2. Chọn ra N phần tử tốt nhất từ $P_t \cup Q_t$ để tạo ra P_{t+1}.
- Tính chất đặc trưng: Để đánh giá các solution trong bước (2), thuật toán sử dụng:
  1. Sắp xếp bất thống trị (fast nondominated sorting) để chọn ra phần tử thống trị các phần tử khác.
  2. khoảng cách quần thể (crowding distance) để đảm bảo sự đa dạng của quần thể.

Thuật toán NSGA-II

- Sau lần chạy thứ t, duy trì một quần thể solution P_t có kích thước N cố định qua mọi lần chạy.
- Trong lần chạy thứ $t+1$:
  1. Sử dụng các toán tử selection, crossover và mutation để tạo ra quần thể con cháu Q_t từ P_t.
  2. Chọn ra N phần tử tốt nhất từ $P_t \cup Q_t$ để tạo ra P_{t+1}.
- Tính chất đặc trưng: Để đánh giá các solution trong bước (2), thuật toán sử dụng:
  1. Sắp xếp bất thống trị (fast nondominated sorting) để chọn ra phần tử thống trị các phần tử khác.
  2. khoảng cách quần thể (crowding distance) để đảm bảo sự đa dạng của quần thể.
- Độ phức tạp: $O(mN^2)$ mỗi lần chạy, với m là số mục tiêu.

Các đặc tính

- **Độ hội tụ:** NSGA-II sử dụng Fast nondominated sorting để sắp xếp lại các cá thể trong $P_t \cup Q_t$, sau đó ưu tiên giữ lại các cá thể ít bị thống trị nhất.

Các đặc tính

- **Độ hội tụ:** NSGA-II sử dụng Fast nondominated sorting để sắp xếp lại các cá thể trong $P_t \cup Q_t$, sau đó ưu tiên giữ lại các cá thể ít bị thống trị nhất.
- **Độ đa dạng:** NSGA-II sử dụng Crowding distance để tính tổng khoảng cách giữa các cá thể theo từng mục tiêu của bài toán, sau đó chọn ra các cá thể cách nhau xa nhất. Như vậy các cá thể được chọn sẽ trải đều trên tập.

Thuật toán MOEA/D

- Khởi tạo một quần thể gồm N cá thể. Mỗi cá thể có vector λ_i được khởi tạo ngẫu nhiên và mảng $B(i)$ chứa T chỉ số cá thể hàng xóm có khoảng các Euclid theo λ gần với i nhất.

Thuật toán MOEA/D

- Khởi tạo một quần thể gồm N cá thể. Mỗi cá thể có vector λ_i được khởi tạo ngẫu nhiên và mảng $B(i)$ chứa T chỉ số cá thể hàng xóm có khoảng các Euclid theo λ gần với i nhất.
- Sau mỗi lần chạy duy trì một một tập quần thể EP (External Population) để lưu lại các cá thể tốt nhất tính đến hiện tại.

Thuật toán MOEA/D

- Trong lần chạy thứ t:

Thuật toán MOEA/D

- Trong lần chạy thứ t:
- Duyệt $\forall$ i:
  1. Chọn 2 chỉ số ngẫu nhiên k, l từ trong các hàng xóm của i.
  2. Dùng các toán tử crossover, mutation để tạo ra cá thể con cháu/
  3. Nếu cá thể con cháu vi phạm ràng buộc thì dùng một thuật toán sửa chữa để tạo thành một cá thể con cháu mới thỏa mãn ràng buộc.

Thuật toán MOEA/D

- Trong lần chạy thứ t:
- Duyệt $\forall$ i:
  1. Chọn 2 chỉ số ngẫu nhiên k, l từ trong các hàng xóm của i.
  2. Dùng các toán tử crossover, mutation để tạo ra cá thể con cháu/
  3. Nếu cá thể con cháu vi phạm ràng buộc thì dùng một thuật toán sửa chữa để tạo thành một cá thể con cháu mới thỏa mãn ràng buộc.
  4. So sánh cá thể con cháu với các hàng xóm của i.
  5. So sánh với các cá thể trong EP.

Thuật toán MOEA/D

- Trong lần chạy thứ t:
- Duyệt $\forall$ i:
  1. Chọn 2 chỉ số ngẫu nhiên k, l từ trong các hàng xóm của i.
  2. Dùng các toán tử crossover, mutation để tạo ra cá thể con cháu/
  3. Nếu cá thể con cháu vi phạm ràng buộc thì dùng một thuật toán sửa chữa để tạo thành một cá thể con cháu mới thỏa mãn ràng buộc.
  4. So sánh cá thể con cháu với các hàng xóm của i.
  5. So sánh với các cá thể trong EP.
- Độ phức tạp: $O(mNT)$.

Các đặc tính

- **Độ hội tụ:** MOEA/D sử dụng một hàm hội tụ (Agression function) để so sánh giữa cá thể con cháu sinh ra và các hàm xóm của nó.

Các đặc tính

- **Độ hội tụ:** MOEA/D sử dụng một hàm hội tụ (Agression function) để so sánh giữa cá thể con cháu sinh ra và các hàm xóm của nó.
- **Độ đa dạng:** Bằng việc sinh ra con cháu từ việc chọn ngẫu nhiên từ đời cha là 2 hàng xóm, độ đa dạng của đời con cháu được đảm bảo.

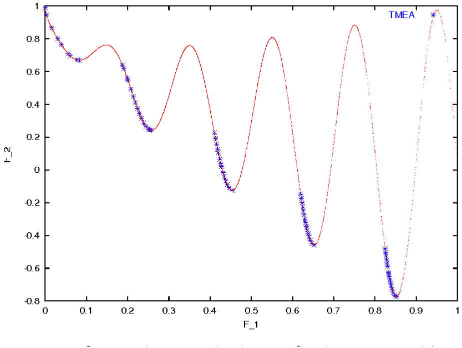
Thực nghiệm: Bài toán ZDT3

- Hàm mục tiêu:
  - $f_1(x) = x_1$
  - $f_2(x) = g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right]$, với

$$g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{(n-1)}$$

- Miền tìm kiếm: $x = (x_1, ..., x_n)^T \in [0,1]^T$.
- Biên Pareto: Không liên tục, lồi.

Thực nghiệm: Bài toán ZDT3



Hình: Nguồn: Semantic Scholar

So sánh giữa NSGA-II và MOEA/D

- Độ phức tạp mỗi lần chạy của MOEA/D nhỏ hơn NSGA-II.

So sánh giữa NSGA-II và MOEA/D

- Độ phức tạp mỗi lần chạy của MOEA/D nhỏ hơn NSGA-II.
- NSGA-II chỉ cần một hệ số ngoài là mật độ đột biến (mutation rate) khi chạy thuật toán.

So sánh giữa NSGA-II và MOEA/D

- Độ phức tạp mỗi lần chạy của MOEA/D nhỏ hơn NSGA-II.
- NSGA-II chỉ cần một hệ số ngoài là mật độ đột biến (mutation rate) khi chạy thuật toán.
- NSGA-II hội tụ tới biên Pareto nhanh hơn.

So sánh giữa NSGA-II và MOEA/D

- Độ phức tạp mỗi lần chạy của MOEA/D nhỏ hơn NSGA-II.
- NSGA-II chỉ cần một hệ số ngoài là mật độ đột biến (mutation rate) khi chạy thuật toán.
- NSGA-II hội tụ tới biên Pareto nhanh hơn.
- NSGA-II đảm bảo kích cỡ quần thể cố định $\Longrightarrow$ dễ quản lý và lưu trữ hơn.

So sánh giữa NSGA-II và MOEA/D

- Độ phức tạp mỗi lần chạy của MOEA/D nhỏ hơn NSGA-II.
- NSGA-II chỉ cần một hệ số ngoài là mật độ đột biến (mutation rate) khi chạy thuật toán.
- NSGA-II hội tụ tới biên Pareto nhanh hơn.
- NSGA-II đảm bảo kích cỡ quần thể cố định $\implies$ dễ quản lý và lưu trữ hơn.

Tại sao lại phải nghiên cứu và sử dụng MOEA/D?

Many Objectives Problems (MoAPs)

- Không có định nghĩa chung cho MaOPs.

Many Objectives Problems (MoAPs)

- Không có định nghĩa chung cho MaOPs.
- Động lực chính đằng sau MaOPs là làm nổi bật những thách thức đặt ra bởi một số lượng lớn các mục tiêu cho các thuật toán tiến hóa đa mục tiêu (MOEAs) hiện tại.

Many Objectives Problems (MoAPs)

- Không có định nghĩa chung cho MaOPs.
- Động lực chính đằng sau MaOPs là làm nổi bật những thách thức đặt ra bởi một số lượng lớn các mục tiêu cho các thuật toán tiến hóa đa mục tiêu (MOEAs) hiện tại.
- Hiện tại ta có thể định nghĩa MaOPs là các bài toán đa mục tiêu MOPs với $m \geq 4$.

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.
- Tính toán độ đa dạng thuật toán trở nên khó khăn.

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.
- Tính toán độ đa dạng thuật toán trở nên khó khăn.
- Quá trình tái tổ hợp có thể không hiệu quả.

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.
- Tính toán độ đa dạng thuật toán trở nên khó khăn.
- Quá trình tái tổ hợp có thể không hiệu quả.
- Việc thể hiện trade-off surface khó khăn.

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.
- Tính toán độ đa dạng thuật toán trở nên khó khăn.
- Quá trình tái tổ hợp có thể không hiệu quả.
- Việc thể hiện trade-off surface khó khăn.
- Tốn thời gian để so sánh hiệu suất giữa các thuật toán với nhau (Performance metrics).

Khó khăn trong quá trình giải quyết MoAPs

- Phần lớn các cá thể là không trội, không thể sắp xếp được.
- Tính toán độ đa dạng thuật toán trở nên khó khăn.
- Quá trình tái tổ hợp có thể không hiệu quả.
- Việc thể hiện trade-off surface khó khăn.
- Tốn thời gian để so sánh hiệu suất giữa các thuật toán với nhau (Performance metrics).
- Rất khó để biểu diễn lời giải một cách trực quan.

Hai ý tưởng cho các thuật toán Tiến hóa Tối ưu Nhiều mục tiêu - EMO

1.

Sử dụng nguyên tắc trội đặc biệt: ϵ-Domination, các chiến lược hạn chế giao phối (Mating Restriction Scheme) hoặc chiến lược tái tổ hợp đặc biệt (Special Recombination Scheme như SBX với distribution index lớn, ...).

Hai ý tưởng cho các thuật toán Tiến hóa Tối ưu Nhiều mục tiêu - EMO

1.

Sử dụng nguyên tắc trội đặc biệt: ϵ-Domination, các chiến lược hạn chế giao phối (Mating Restriction Scheme) hoặc chiến lược tái tổ hợp đặc biệt (Special Recombination Scheme như SBX với distribution index lớn, ...).

2.

Sử dụng nền tảng của các nghiên cứu đa mục tiêu trước.

Một số thuật toán tiến hóa để giải quyết các bài toái MoAPs hiện tại

- MSOPS-II
- MOEA/D
- HypE
- PICEA-g
- SDE
- GrEA
- NSGA-III
- KnEA
- RVEA
- Two-Arch 2
- ϕ - DEA
- MOEA/DD
- AnD