Mutilobjective Evolutionary Optimization

Taskoverflow

Ngày 12 tháng 12 năm 2018

INTRODUTION

Point 1

Thế nào là bài toán tối ưu đa mục tiêu?

Bài toán tối ưu đa mục tiêu trong thực tế

Mục tiêu 1

Quãng đường

Mục tiêu 2

Thời gian

Bài toán tối ưu đa mục tiêu trong thực tế

- Với nhiêu liệu có trước, công sinh A sinh ra là cố định.

Bài toán tối ưu đa mục tiêu trong thực tế

- Với nhiêu liệu có trước, công sinh A sinh ra là cố định.
- Thời gian t giảm khi vận tốc v tăng

Bài toán tối ưu đa mục tiêu trong thực tế

- Với nhiêu liệu có trước, công sinh A sinh ra là cố định.
- Thời gian t giảm khi vận tốc v tăng
- Vận tốc v tăng thì lực cản tăng, công hao phí tăng

Bài toán tối ưu đa mục tiêu trong thực tế

- Với nhiêu liệu có trước, công sinh A sinh ra là cố định.
- Thời gian t giảm khi vận tốc v tăng
- Vận tốc v tăng thì lực cản tăng, công hao phí tăng
- Công hao phí tăng, quãng đường có thể đi được giảm

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

$$\begin{cases} \mathrm{y} = \mathrm{f(x)} = (\mathrm{f}_1(x), ..., f_m(x)) \\ x \in \Omega \end{cases}$$

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

$$\begin{cases} \mathrm{y} = \mathrm{f(x)} = (\mathrm{f}_1(x), ..., f_m(x)) \\ x \in \Omega \end{cases}$$

Không gian mục tiêu

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

Các định nghĩa cơ bản

- 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 $j \in 1, 2, ..., m, f_j(x) < f_j(y)$.

Các định nghĩa cơ bản

- 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 $j \in 1, 2, ..., m, f_j(x) < f_j(y)$.
- 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

- 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 $j \in 1, 2, ..., m, f_j(x) < f_j(y)$.
- Một nghiệm $x^* \in \Omega$ là nghiệm tối ưu Pareto nếu không tồn tại $x \in \Omega$ trội hơn nó.
- Tập hợp nghiệm tối ưu Pareto được gọi là tập Pareto.

Các định nghĩa cơ bản

- 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 $j \in 1, 2, ..., m, f_j(x) < f_j(y)$.
- Một nghiệm $x^* \in \Omega$ là nghiệm tối ưu Pareto nếu không tồn tại $x \in \Omega$ trội hơn nó.
- Tập hợp nghiệm tối ưu Pareto được gọi là tập Pareto.
- Các vector mục tiêu tương ứng của tập Pareto được gọi là Biên Parato(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

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.

Thuật toán MOEA/D

- Khởi tạo một quần thể gồm N cá thể. Mỗi cá thể gồm λ_i được khởi tạo ngẫu nhiên. $B(i)$ 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 gồm λ_i được khởi tạo ngẫu nhiên. $B(i)$ 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: $O(mNT)$

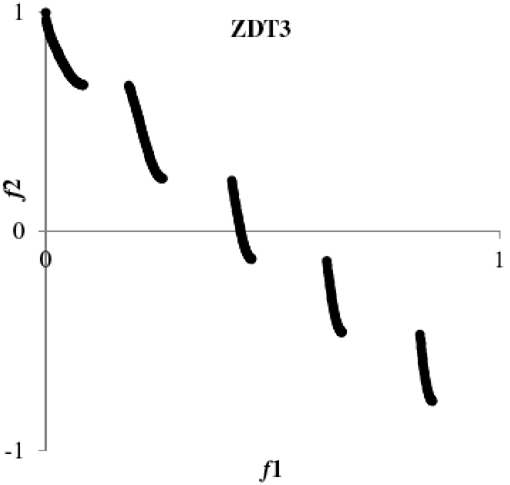
Thực nghiệm: Bài toán ZTD3

- 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 Parento: Không liên tục, lồi.

Thực nghiệm: Bài toán ZTD3



Hình: Nguồn: ResearchGate

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 Parento 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 Parento 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 Parento nhanh hơn.
- NSGA-II đảm bảo kích cỡ quần thể cố định $\Longrightarrow$ dễ quản lý và lưu trữ hơn.

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

MANY OBJECTIVE PROBLEM (MoAPs)

- Không có định nghĩa chung cho MaOPs

MANY OBJECTIVE PROBLEM (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 OBJECTIVE PROBLEM (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

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

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

KHÓ KHĂN

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

- 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

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

- 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

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

- 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 sự trade-off surface khó khăn

KHÓ KHĂN

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

- 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 sự trade-off surface khó khăn
- Tốn thời gian để so sánh hiệu suất giữa các thuật toán với nhau (Performance metrics)

KHÓ KHĂN

Các khó khăn trong quá trình giải quyết các bài toán nhiều mục tiêu (Many Objectives)

- 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 sự 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 Many Objective EMO

1.

Sử dụng nguyên tắc trội đặc biệt:-domination, mating restriction scheme or special recombination scheme (SBX với large distribution index,. . .)

Hai ý tưởng cho Many Objective EMO

1.

Sử dụng nguyên tắc trội đặc biệt:-domination, mating restriction scheme or special recombination scheme (SBX với large distribution index,. . .)

2.

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