

NON-DOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

 agungsetiabudi@ub.ac.id

Non-dominated Sorting Genetic Algorithm II

- NSGA-II (Non-dominated Sorting Genetic Algorithm II) adalah algoritma evolusi yang dirancang untuk menyelesaikan masalah optimisasi multi-objektif.
- Masalah optimisasi multi-objektif melibatkan pengoptimalan dua atau lebih tujuan yang sering kali saling bertentangan.
- Misalnya, dalam desain produk, kita mungkin ingin meminimalkan biaya sekaligus memaksimalkan kualitas.
- NSGA-II dikembangkan oleh Kalyanmoy Deb dan timnya pada tahun 2000 sebagai perbaikan dari algoritma NSGA pertama.

Konsep Dasar

NSGA-II menggunakan dua konsep kunci:

- **Domination:** Dalam konteks ini, satu solusi A mendominasi solusi B jika A lebih baik dari B dalam semua tujuan atau lebih baik dalam setidaknya satu tujuan dan tidak lebih buruk dalam tujuan lainnya.
- **Crowding Distance:** Metode ini digunakan untuk menjaga keragaman populasi. Dalam hal ini, solusi yang lebih jauh dari tetangga terdekatnya dalam ruang tujuan akan diprioritaskan.

Tahapan Algoritma

1. **Inisialisasi:** Buat populasi awal secara acak dari solusi potensial.
2. **Evaluasi:** Hitung nilai fungsi objektif untuk setiap individu dalam populasi.
3. **Non-dominated Sorting:** Klasifikasikan individu ke dalam lapisan (front) berdasarkan dominasi. Individu yang tidak didominasi membentuk front pertama, yang didominasi oleh individu lain membentuk front kedua, dan seterusnya.
4. **Crowding Distance Calculation:** Hitung jarak kerumunan untuk individu dalam satu front, yang akan membantu dalam menjaga keragaman populasi.

Tahapan Algoritma

5. **Pemilihan:** Gabungkan populasi saat ini dan populasi anak-anak yang dihasilkan dari operasi crossover dan mutasi. Pilih individu untuk populasi berikutnya dengan mempertimbangkan lapisan dominasi dan jarak kerumunan.
6. **Crossover dan Mutasi:** Terapkan operator crossover dan mutasi untuk menghasilkan populasi anak-anak. Crossover menggabungkan dua solusi untuk membuat solusi baru, sementara mutasi memperkenalkan variasi.

Tahapan Algoritma

7. **Iterasi:** Ulangi langkah evaluasi, non-dominated sorting, crowding distance calculation, pemilihan, dan generasi baru populasi sampai kriteria penghentian tercapai (misalnya, jumlah generasi maksimum atau konvergensi).

Keunggulan NSGA-II

- **Efisiensi:** NSGA-II memiliki kompleksitas waktu yang lebih rendah dibandingkan dengan metode optimisasi multi-objektif lainnya seperti algoritma berbasis pareto lainnya.
- **Keragaman:** Dengan menggunakan crowding distance, NSGA-II dapat menghasilkan populasi solusi yang lebih beragam, yang penting dalam menemukan trade-off yang baik antara berbagai tujuan.
- **Kualitas Solusi:** NSGA-II sering kali menghasilkan solusi yang lebih baik dalam hal keseimbangan antara berbagai tujuan.

Contoh Kasus

Optimisasi sistem penyimpanan energi terbarukan: Dalam contoh ini, kita akan mengoptimalkan penggunaan panel surya dan sistem penyimpanan baterai untuk meminimalkan biaya dan emisi karbon, dengan fokus pada dua tujuan utama.

Definisikan Masalah

Kita ingin merancang sistem energi yang mengoptimalkan penggunaan energi terbarukan. Tujuan utama adalah:

- **Tujuan 1:** Minimalkan total biaya sistem (C).
- **Tujuan 2:** Minimalkan emisi karbon (E).

Parameter

- Misalkan kita memiliki parameter yang mempengaruhi kedua tujuan ini:
 - x_1 : Jumlah panel surya (unit)
 - x_2 : Kapasitas baterai (kWh)

Fungsi Tujuan (*Objective Function*)

- Fungsi biaya dan emisi karbon dapat didefinisikan sebagai berikut:
 - **Fungsi biaya sistem:**
$$C = 1000x_1 + 200x_2$$
 - Di mana 1000 adalah biaya per panel surya dan 200 adalah biaya per kWh kapasitas baterai.
 - **Fungsi emisi karbon:**
$$E = 0.2x_1 + 0.1x_2$$
 - Di mana 0.2 adalah emisi karbon per panel surya dan 0.1 adalah emisi karbon per kWh kapasitas baterai.

Inisialisasi Populasi

- Kita mulai dengan populasi awal acak, misalkan kita memiliki 5 individu:
 - $P_1 = (5, 10)$
 - $P_2 = (6, 8)$
 - $P_3 = (4, 12)$
 - $P_4 = (7, 5)$
 - $P_5 = (3, 15)$

Evaluasi Fungsi Tujuan

Hitung C dan E untuk setiap individu:

- Untuk P_1 :
 - $C_1 = 1000(5) + 200(10) = 5000 + 2000 = 7000$
 $E_1 = 0.2(5) + 0.1(10) = 1 + 1 = 2$
- Untuk P_2 :
 - $C_2 = 1000(6) + 200(8) = 6000 + 1600 = 7600$
 $E_2 = 0.2(6) + 0.1(8) = 1.2 + 0.8 = 2$

Evaluasi Fungsi Tujuan

- Untuk P_3 :
 - $C_3 = 1000(4) + 200(12) = 4000 + 2400 = 6400$
 $E_3 = 0.2(4) + 0.1(12) = 0.8 + 1.2 = 2$
- Untuk P_4 :
 - $C_4 = 1000(7) + 200(5) = 7000 + 1000 = 8000$
 $E_4 = 0.2(7) + 0.1(5) = 1.4 + 0.5 = 1.9$
- Untuk P_5 :
 - $C_5 = 1000(3) + 200(15) = 3000 + 3000 = 6000$
 $E_5 = 0.2(3) + 0.1(15) = 0.6 + 1.5 = 2.1$

Hasil Evaluasi

Setelah evaluasi, kita memiliki:

- $P_1 = (7000, 2)$
- $P_2 = (7600, 2)$
- $P_3 = (6400, 2)$
- $P_4 = (8000, 1.9)$
- $P_5 = (6000, 2.1)$

Non-dominated Sorting

Identifikasi solusi yang tidak didominasi:

- **Front 1:** P_3 (6400, 2), P_4 (8000, 1.9), P_5 (6000, 2.1)
- **Front 2:** P_1 (7000, 2), P_2 (7600, 2)

Crowding Distance Calculation

Hitung jarak kerumunan untuk setiap individu dalam front yang sama.
Misalkan kita fokus pada Front 1:

- Untuk P_3 :
 - Jarak di tujuan C : $8000 - 6400 = 1600$
 - Jarak di tujuan E : $2 - 1.9 = 0.1$
 - Total crowding distance = $1600 + 0.1 = 1600.1$.
- Untuk P_4 , lakukan hal yang sama.

Pemilihan dan Crossover

Pilih individu untuk membentuk generasi baru menggunakan lapisan dominasi dan jarak kerumunan. Misalkan kita memilih 3 individu terbaik untuk melakukan crossover:

- Lakukan crossover antara individu terpilih dan mutasi.

Generasi Baru

Setelah crossover dan mutasi, kita mendapatkan generasi baru yang juga harus dievaluasi kembali menggunakan langkah-langkah di atas.

Iterasi

Ulangi langkah-langkah di atas sampai kriteria penghentian tercapai, seperti jumlah generasi atau konvergensi.