

ESTIMATION OF DISTRIBUTION ALGORITHM (EDA)

 agungsetiabudi@ub.ac.id

Estimation of Distribution Algorithm

- Estimation of Distribution Algorithm (EDA) adalah kelas algoritma evolusioner yang berfokus pada pemodelan distribusi solusi yang baik selama proses pencarian, alih-alih hanya menggunakan operasi pembangkitan acak seperti pada algoritma genetik tradisional.
- EDA merupakan pendekatan yang inovatif dalam algoritma optimasi dan dapat digunakan untuk memecahkan berbagai masalah kompleks.

Konsep Dasar EDA

1. **Pengumpulan Solusi:** EDA dimulai dengan populasi awal dari solusi kandidat. Proses ini mirip dengan algoritma genetik, di mana solusi yang ada dievaluasi berdasarkan fungsi tujuan yang telah ditentukan.
2. **Pemodelan Distribusi:** Alih-alih menggunakan crossover dan mutasi untuk menghasilkan solusi baru, EDA membangun model probabilistik dari solusi-solusi terbaik yang ada. Model ini dapat berupa distribusi statistik seperti distribusi Gaussian, multinomial, atau lainnya yang sesuai dengan karakteristik solusi.

Konsep Dasar EDA

3. **Sampling:** Setelah model probabilistik dibangun, EDA menggunakan model ini untuk menghasilkan solusi baru. Proses sampling ini memungkinkan algoritma untuk mengeksplorasi ruang pencarian dengan cara yang lebih efisien dibandingkan dengan generasi acak murni.
4. **Iterasi:** Proses ini diulang, di mana solusi baru dievaluasi dan model distribusi diperbarui berdasarkan solusi yang paling sesuai. Dengan iterasi ini, EDA berusaha untuk mengeksplorasi dan mengeksploitasi ruang pencarian.

Langkah-Langkah Umum dalam EDA

1. **Inisialisasi:** Buat populasi awal dari solusi acak.
2. **Evaluasi:** Hitung nilai fungsi tujuan untuk setiap solusi.
3. **Seleksi:** Pilih solusi terbaik berdasarkan kriteria tertentu.
4. **Pembentukan Model:** Bangun model distribusi dari solusi yang terpilih.
5. **Sampling:** Ambil solusi baru dari model distribusi.
6. **Iterasi:** Ulangi proses hingga kriteria penghentian terpenuhi (misalnya, jumlah iterasi atau konvergensi solusi).

Penerapan EDA

EDA dapat diterapkan dalam berbagai bidang dan jenis masalah, antara lain:

1. **Optimasi Fungsional:** EDA sering digunakan untuk mencari minimum atau maksimum dari fungsi matematis yang kompleks, baik dalam ruang berdimensi rendah maupun tinggi.
2. **Jadwal dan Penjadwalan:** Dalam masalah penjadwalan, seperti penjadwalan mesin atau proyek, EDA dapat digunakan untuk menemukan jadwal yang optimal dengan mempertimbangkan berbagai batasan dan kriteria.

Penerapan EDA

3. **Pengoptimalan Kombinatorial:** Masalah kombinatorial seperti Traveling Salesman Problem (TSP) atau knapsack problem dapat diselesaikan menggunakan EDA untuk menemukan solusi optimal atau mendekati optimal.
4. **Pengenalan Pola:** EDA dapat diterapkan dalam pengenalan pola dan klasifikasi, terutama ketika memodelkan hubungan kompleks dalam data.
5. **Desain Sistem:** Dalam desain rekayasa, EDA dapat membantu dalam menemukan konfigurasi sistem yang paling efisien atau efektif dengan mempertimbangkan berbagai parameter desain.

Keunggulan EDA

- EDA dapat menemukan solusi yang lebih baik dalam ruang pencarian yang kompleks dibandingkan dengan metode tradisional.
- Pendekatan probabilistik memungkinkan eksplorasi yang lebih luas terhadap ruang pencarian.
- Lebih sedikit terjebak pada solusi lokal dibandingkan algoritma berbasis crossover dan mutasi.

Kelemahan EDA

- Kinerja EDA sangat tergantung pada pemilihan model distribusi yang tepat.
- Proses pembelajaran distribusi dapat memerlukan waktu komputasi yang lebih lama untuk masalah yang sangat besar.
- EDA mungkin tidak seefisien algoritma lain dalam situasi tertentu, terutama jika distribusi solusi tidak dapat dimodelkan dengan baik.

Contoh Penerapan

- Kita ingin meminimalkan fungsi berikut:
 - $f(x) = (x - 3)^2 + 2$
- Fungsi ini memiliki titik minimum pada $x = 3$ dengan nilai $f(3) = 2$.
- Kita akan menggunakan EDA untuk mencari nilai x yang meminimalkan fungsi ini.

1. Inisialisasi

Kita mulai dengan membuat populasi awal dari solusi acak. Misalkan kita membuat 5 solusi acak dalam rentang $[0, 6]$:

- Populasi awal: $P = [2.0, 5.0, 1.0, 4.5, 3.5]$

```
# Inisialisasi populasi acak dalam rentang [0, 6]  
population = np.random.uniform(0, 6, population_size)
```

2. Evaluasi

Hitung nilai fungsi untuk setiap solusi:

- $f(2.0) = (2.0 - 3)^2 + 2 = 1^2 + 2 = 3$
- $f(5.0) = (5.0 - 3)^2 + 2 = 2^2 + 2 = 6$
- $f(1.0) = (1.0 - 3)^2 + 2 = 2^2 + 2 = 6$
- $f(4.5) = (4.5 - 3)^2 + 2 = 1.5^2 + 2 = 2.25 + 2 = 4.25$
- $f(3.5) = (3.5 - 3)^2 + 2 = 0.5^2 + 2 = 0.25 + 2 = 2.25$

Hasil evaluasi:

- $F = [3, 6, 6, 4.25, 2.25]$

Contoh program

```
# Evaluasi fungsi untuk setiap individu dalam populasi  
fitness_values = np.array([objective_function(x) for x in population])  
  
# Simpan hasil evaluasi  
results.append((population.copy(), fitness_values.copy()))
```

3. Seleksi

- Pilih solusi terbaik (minimal) berdasarkan nilai fungsi. Dalam hal ini, solusi dengan nilai minimum adalah $x = 3.5$ dengan nilai $f(3.5) = 2.25$.
- Kita ambil dua solusi terbaik untuk pembentukan model:
- Solusi terpilih: $S = [3.5, 2.0]$

```
# Seleksi solusi terbaik (dua terbaik)
best_indices = np.argsort(fitness_values)[:2]
selected_solutions = population[best_indices]
```

4. Pembentukan Model

- Berdasarkan solusi terpilih, kita dapat membangun model distribusi. Misalnya, kita bisa menggunakan distribusi Gaussian untuk memodelkan solusi:
- Rata-rata (mean) μ dan deviasi standar (std) σ :
 - $\mu = \frac{3.5+2.0}{2} = 2.75$
 - $\sigma = \sqrt{\frac{(3.5-2.75)^2+(2.0-2.75)^2}{2}} = \sqrt{\frac{0.5625+0.5625}{2}} = 0.5$
- Model distribusi: $\mathcal{N}(2.75, 0.5^2)$

Contoh Program

```
# Pembentukan model distribusi (Gaussian)  
mean = np.mean(selected_solutions)  
std_dev = np.std(selected_solutions)
```


5. Sampling

- Dari model distribusi yang telah dibangun, kita lakukan sampling untuk menghasilkan solusi baru. Misalkan kita ambil 5 solusi baru:
- Solusi baru (sampling): [2.5, 3.0, 3.2, 2.3, 3.7]

```
# Sampling untuk menghasilkan solusi baru  
population = np.random.normal(mean, std_dev, population_size)
```

6. Iterasi

- Hitung nilai fungsi untuk solusi baru:
 - $f(2.5) = (2.5 - 3)^2 + 2 = 0.5^2 + 2 = 0.25 + 2 = 2.25$
 - $f(3.0) = (3.0 - 3)^2 + 2 = 0 + 2 = 2$
 - $f(3.2) = (3.2 - 3)^2 + 2 = 0.2^2 + 2 = 0.04 + 2 = 2.04$
 - $f(2.3) = (2.3 - 3)^2 + 2 = 0.7^2 + 2 = 0.49 + 2 = 2.49$
 - $f(3.7) = (3.7 - 3)^2 + 2 = 0.7^2 + 2 = 0.49 + 2 = 2.49$
- Hasil evaluasi solusi baru:
 - $F' = [2.25, 2, 2.04, 2.49, 2.49]$

7. Pengulangan

Dari evaluasi solusi baru, solusi terbaik adalah $x = 3.0$ dengan nilai minimum $f(3.0) = 2$. Kita bisa melanjutkan proses ini, memilih solusi terbaik, membentuk model baru, dan sampling hingga konvergensi tercapai atau kriteria penghentian terpenuhi.