Tugas Pemrograman 1 - Searching
Kecerdasan Buatan

Disusun Oleh :
Muhammad Vikhan Muharram (1302213089)
Bimo Zachriansyah Wicaksono Hermawan (1302213012)
Faris Siddiq Ramdan Putra (1302213133)

Kelas :
SE-45-01

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
UNIVERSITAS TELKOM
BANDUNG
2023

Laporan ini dibuat untuk menyelesaikan permasalahan mencari nilai x1 dan x2 sehingga diperoleh nilai minimum dari fungsi berikut.

$$f(x_1, x_2) = -\left(sin(x_1)cos(x_2) + \frac{4}{5}exp\left(1 - \sqrt{x_1^2 + x_2^2}\right)\right)$$

Dengan domain (batas nilai) untuk x1 dan x2 adalah sebagai berikut

$$-10 \leq x_1 \leq 10 \text{ dan } -10 \leq x_2 \leq 10$$

Tim melakukan penyelesaian dengan cara membuat kode program python dengan metode Algoritma Genetika (GA). Tim melakukan analisis dan desain program GA dengan melakukan implementasi berikut.

A. Ukuran Populasi, Rancangan Kromosom, dan Cara Decode

Pada pengujian, tim menentukan populasi sebanyak 10. Dalam satu populasi, tim menentukan terdapat 10 kromosom. Cara decode yang tim pilih adalah sebagai berikut :

$$x = batas\ bawah + (batas\ atas - batas\ bawah) * \frac{x}{2^n - 1}$$

```
# Fungsi generate chromosome
def generateChromosome(chromosomeSize):
    chromosome = []
    for i in range(chromosomeSize):
        chromosome.append(random.randint(0, 1))
    return chromosome

# Fungsi decode chromosome
def decodeChromosome(chromosome):
    x1 = chromosome[0:5]
    x2 = chromosome[5:10]
    x1 = int(''.join([str(i) for i in x1]), 2)
    x2 = int(''.join([str(i) for i in x2]), 2)
    x1 = -10 + (10-(-10)) * (x1/(2**5-1))
    x2 = -10 + (10-(-10)) * (x2/(2**5-1))

    return round(x1, 3), round(x2, 3)
```

B. Metode Pemilihan Orangtua

Tim menentukan metode pemilihan orangtua dengan cara memilih dua populasi yang memiliki tingkat fitness tertinggi

```
# Fungsi penghitungan fitness
def fitness(x1, x2):
    heuristics = -1 * (math.sin(x1) * math.cos(x2) + (4/5) * math.exp(1 - math.sqrt(x1**2 + x2**2)))
    return (1 / heuristics)

# Fungsi pemilihan orangtua
def selectParents(population, numParents):
    parents = []
    for i in range(numParents):
        x = random.randint(0, len(population) - 1)
        parents.append(population[x])
    return parents
```

```
sortPopulationHighest = sorted(fitnessPopulation, key=fitnessPopulation.get, reverse=True)
# Menggunakan fungsi selectParents untuk memilih orangtua
parents = selectParents(sortPopulationHighest, 2)  # Memilih 2 orangtua
parent1 = parents[0]
parent2 = parents[1]
```

C. Metode Operasi Genetik (Pindah Silang dan Mutasi)

Tim memilih metode operasi genetika pindah silang berupa single crossover dengan probabilitas (P_c). Prinsipnya adalah menukar sebagian gen antara dua kromosom parent untuk menghasilkan dua kromosom offspring yang baru. Fungsi Mutasi menggunakan metode flip bit pada fungsi mutasi dengan probabilitas (P_m). Prinsipnya adalah mengubah nilai 1 bit tertentu pada kromosom dari 0 menjadi 1 atau sebaliknya.

```
def crossover(parent1, parent2):
    offspring1 = []
    offspring2 = []
    pc = 0.8
    for i in range(10):
        if random.random() < pc:
            offspring1.append(parent1[i])
            offspring2.append(parent2[i])
        else:
            offspring1.append(parent2[i])
            offspring2.append(parent1[i])
    return offspring1, offspring2
```

```
# Fungsi mutasi
def mutasi(child1, child2):
    pm = 0.3
    for i in range(len(child1)):
        if random.random() < pm:
            child1[i] = str(1 - int(child1[i]))
    for i in range(len(child2)):
        if random.random() < pm:
            child2[i] = str(1 - int(child2[i]))
    return child1, child2
```

D. Probabilitas Operasi Genetik (Pc dan Pm)

Tim memilih tingkat probabilitas crossover (P_c) sebesar 0,8 dan probabilitas mutasi (P_m) sebesar 0,3

```
pc = 0.8  pm = 0.3
```

E. Metode Pergantian Generasi (Seleksi Survivor)

Seleksi survivor yang dipakai oleh tim adalah elitisme. Elitisme merupakan suatu pendekatan seleksi individu terbaik dari generasi saat ini diberikan kesempatan untuk langsung "mewariskan" diri mereka ke generasi berikutnya tanpa perubahan.

```python
# Fungsi pindah generasi (elitism agar gen terbaik tidak hilang) probabilitas 0,8
def elitism(population, offspring):
    allChromosomes = population + offspring
    allFitness = [fitness(*decodeChromosome(chromosome)) for chromosome in allChromosomes]
    pc = 0.8

    if random.random() < pc:
        idx = allFitness.index(max(allFitness))
        bestChromosome = allChromosomes[idx]
    else:
        # Jika probabilitas elitisme tidak terpenuhi, maka hanya mengambil salah satu individu secara acak
        bestChromosome = random.choice(allChromosomes)

    return bestChromosome

# Menggunakan fungsi elitism untuk memutuskan apakah mempertahankan individu terbaik dari generasi sebelumnya
if previousBestChromosome is not None:
    bestChromosome = elitism([previousBestChromosome], [highestPopulation[0]])
    if bestChromosome == previousBestChromosome:
        fitnessPopulation.popitem()
        fitnessPopulation[previousBestChromosome] = fitness(*decodeChromosome(previousBestChromosome))

previousBestChromosome = highestPopulation[0]
```

F. Kriteria Penghentian Evolusi (Loop)

Tim menentukan kriteria penghentian berupa “Evolusi Tidak Menghasilkan Kemajuan”, yaitu kriteria penghentian ketika evolusi tidak menghasilkan perbaikan dalam beberapa generasi terakhir dan menghentikan GA jika tidak ada perbaikan yang terlihat dengan jumlah batasan evolusi yang ditentukan oleh tim sebanyak 100

```python
stagnationLimit = 100  # Ambang batas stagnasi (misalnya, 10 generasi tanpa peningkatan)
stagnationCount = 0  # Menghitung berapa banyak generasi tanpa peningkatan
bestFitness = -float('inf')  # Menyimpan fitness terbaik yang ditemukan
```

```python
if fitnessPopulation[highestPopulation[0]] > bestFitness:
    bestFitness = fitnessPopulation[highestPopulation[0]]
    stagnationCount = 0  # Reset stagnation count
else:
    stagnationCount += 1  # Tidak ada peningkatan, tambahkan stagnationCount

# Periksa apakah telah terjadi stagnasi
if stagnationCount >= stagnationLimit:
    print("Evolusi tidak menghasilkan kemajuan. Menghentikan evolusi.")
    break
```

Output Program

Population	Chromosome	x1	x2	Fitness
1	1101101001	7.419	-4.194	2.227625838799169
2	0110000101	-2.258	-6.774	1.4704936571346336
3	0001110010	-8.065	1.613	-23.901743058313343
4	0110000110	-2.258	-6.129	1.3146042629645462
5	1100000100	5.484	-7.419	3.313057812300181
6	1100110101	6.129	3.548	-6.9982397185352285
7	1101100010	7.419	-8.71	1.460097169039153
8	1100011100	5.484	8.065	-6.655049102279447
9	1001001001	1.613	-4.194	2.1243852814738373
10	0100000100	-4.839	-7.419	-2.3904564696251884

Generation	Chromosome	x1	x2	Fitness
1	1100000100	-2.258	-6.129	3.313057812300181
2	1100000100	2.903	-6.129	3.313057812300181
3	1100000100	7.419	-4.194	3.313057812300181
4	1100000100	7.419	-4.194	3.313057812300181
5	1100000100	7.419	-6.129	3.313057812300181
6	1100000100	-7.419	-6.129	3.313057812300181
7	1100000100	-0.968	-7.419	3.313057812300181
8	1100000100	-0.968	-8.71	3.313057812300181
9	1100000100	-8.065	-5.484	3.313057812300181
10	1100000100	5.484	-5.484	3.313057812300181
11	1100000100	1.613	-4.194	3.313057812300181
12	1100000100	7.419	-7.419	3.313057812300181
13	1100000100	-8.065	-7.419	3.313057812300181
14	1100000100	5.484	-7.419	3.313057812300181
15	1100000100	5.484	-7.419	3.313057812300181
16	1100000100	5.484	-7.419	3.313057812300181
17	1100000100	5.484	-7.419	3.313057812300181
.	.	.	.	.
18	1100000100	5.484	-7.419	3.313057812300181
19	1100000100	5.484	-7.419	3.313057812300181
20	1100000100	5.484	-7.419	3.313057812300181
21	1100000100	5.484	-7.419	3.313057812300181
22	1100000100	5.484	-7.419	3.313057812300181
23	1100000100	5.484	-7.419	3.313057812300181
24	1100000100	5.484	-7.419	3.313057812300181
25	1100000100	5.484	-7.419	3.313057812300181
26	1100000100	5.484	-7.419	3.313057812300181
27	1100000100	5.484	-7.419	3.313057812300181
28	1100000100	5.484	-7.419	3.313057812300181
29	1100000100	5.484	-7.419	3.313057812300181
30	1100000100	5.484	-7.419	3.313057812300181
31	1100000100	5.484	-7.419	3.313057812300181
32	1100000100	5.484	-7.419	3.313057812300181
33	1100000100	5.484	-7.419	3.313057812300181
34	1100000100	5.484	-7.419	3.313057812300181
35	1100000100	5.484	-7.419	3.313057812300181
36	1100100100	5.484	-7.419	15.486662790940798
37	1100100100	5.484	-7.419	15.486662790940798
38	1100100100	6.129	-7.419	15.486662790940798
39	1100100100	6.129	-7.419	15.486662790940798
40	1100100100	6.129	-7.419	15.486662790940798
41	1100100100	6.129	-7.419	15.486662790940798
42	1100100100	6.129	-7.419	15.486662790940798
43	1100100100	6.129	-7.419	15.486662790940798
44	1100100100	6.129	-7.419	15.486662790940798
45	1100100100	6.129	-7.419	15.486662790940798
46	1100100100	6.129	-7.419	15.486662790940798
47	1100100100	6.129	-7.419	15.486662790940798
48	1100100100	6.129	-7.419	15.486662790940798
49	1100100100	6.129	-7.419	15.486662790940798
50	1100100100	6.129	-7.419	15.486662790940798

51	1100100100	6.129	-7.419	15.486662790940798
52	1100100100	6.129	-7.419	15.486662790940798
53	1100100100	6.129	-7.419	15.486662790940798
54	1100100100	6.129	-7.419	15.486662790940798
55	1100100100	6.129	-7.419	15.486662790940798
56	1100100100	6.129	-7.419	15.486662790940798
57	1100100100	6.129	-7.419	15.486662790940798
58	1100100100	6.129	-7.419	15.486662790940798
59	1100100100	6.129	-7.419	15.486662790940798
60	1100100100	6.129	-7.419	15.486662790940798
61	1100100100	6.129	-7.419	15.486662790940798
62	1100100100	6.129	-7.419	15.486662790940798
63	1100100100	6.129	-7.419	15.486662790940798
64	1100100100	6.129	-7.419	15.486662790940798
65	1100100100	6.129	-7.419	15.486662790940798
66	1100100100	6.129	-7.419	15.486662790940798
67	1100100100	6.129	-7.419	15.486662790940798
68	1100100100	6.129	-7.419	15.486662790940798
69	1100100100	6.129	-7.419	15.486662790940798
70	1100100100	6.129	-7.419	15.486662790940798
71	1100100100	6.129	-7.419	15.486662790940798
72	1100100100	6.129	-7.419	15.486662790940798
73	1100100100	6.129	-7.419	15.486662790940798
74	1100100100	6.129	-7.419	15.486662790940798
75	1100100100	6.129	-7.419	15.486662790940798
76	1100100100	6.129	-7.419	15.486662790940798
77	1100100100	6.129	-7.419	15.486662790940798
78	1100100100	6.129	-7.419	15.486662790940798
79	1100100100	6.129	-7.419	15.486662790940798
80	1100100100	6.129	-7.419	15.486662790940798
81	1100100100	6.129	-7.419	15.486662790940798
82	1100100100	6.129	-7.419	15.486662790940798
83	1100100100	6.129	-7.419	15.486662790940798
84	1100100100	6.129	-7.419	15.486662790940798

84	1100100100	6.129	-7.419	15.486662790940798
85	1100100100	6.129	-7.419	15.486662790940798
86	1100100100	6.129	-7.419	15.486662790940798
87	1100100100	6.129	-7.419	15.486662790940798
88	1100100100	6.129	-7.419	15.486662790940798
89	1100100100	6.129	-7.419	15.486662790940798
90	1100100100	6.129	-7.419	15.486662790940798
91	1100100100	6.129	-7.419	15.486662790940798
92	1100100100	6.129	-7.419	15.486662790940798
93	1100100100	6.129	-7.419	15.486662790940798
94	1100100100	6.129	-7.419	15.486662790940798
95	1100100100	6.129	-7.419	15.486662790940798
96	1100100100	6.129	-7.419	15.486662790940798
97	1100100100	6.129	-7.419	15.486662790940798
98	1100100100	6.129	-7.419	15.486662790940798
99	1100100100	6.129	-7.419	15.486662790940798
100	1100100100	6.129	-7.419	15.486662790940798
101	1100100100	6.129	-7.419	15.486662790940798
102	1100100100	6.129	-7.419	15.486662790940798
103	1100100100	6.129	-7.419	15.486662790940798
104	1100100100	6.129	-7.419	15.486662790940798
105	1100100100	6.129	-7.419	15.486662790940798
106	1100100100	6.129	-7.419	15.486662790940798
107	1100100100	6.129	-7.419	15.486662790940798
108	1100100100	6.129	-7.419	15.486662790940798
109	1100100100	6.129	-7.419	15.486662790940798
110	1100100100	6.129	-7.419	15.486662790940798
111	1100100100	6.129	-7.419	15.486662790940798
112	1100100100	6.129	-7.419	15.486662790940798
113	1100100100	6.129	-7.419	15.486662790940798
114	1100100100	6.129	-7.419	15.486662790940798
115	1100100100	6.129	-7.419	15.486662790940798
116	1100100100	6.129	-7.419	15.486662790940798
117	1100100100	6.129	-7.419	15.486662790940798

117	1100100100	6.129	-7.419	15.486662790940798
118	1100100100	6.129	-7.419	15.486662790940798
119	1100100100	6.129	-7.419	15.486662790940798
120	1100100100	6.129	-7.419	15.486662790940798
121	1100100100	6.129	-7.419	15.486662790940798
122	1100100100	6.129	-7.419	15.486662790940798
123	1100100100	6.129	-7.419	15.486662790940798
124	1100100100	6.129	-7.419	15.486662790940798
125	1100100100	6.129	-7.419	15.486662790940798
126	1100100100	6.129	-7.419	15.486662790940798
127	1100100100	6.129	-7.419	15.486662790940798
128	1100100100	6.129	-7.419	15.486662790940798
129	1100100100	6.129	-7.419	15.486662790940798
130	1100100100	6.129	-7.419	15.486662790940798
131	1100100100	6.129	-7.419	15.486662790940798
132	1100100100	6.129	-7.419	15.486662790940798
133	1100100100	6.129	-7.419	15.486662790940798
134	1100100100	6.129	-7.419	15.486662790940798
135	1100100100	6.129	-7.419	15.486662790940798
136	1100100100	6.129	-7.419	15.486662790940798

Evolusi tidak menghasilkan kemajuan. Menghentikan evolusi.
Kromosom terbaik : 1100100100
x1 : 6.129
x2 : -7.419
Fitness : 15.486662790940798

Kode Program
https://github.com/vikhanmuhammad/TubesAiAlJabbar

Nama	Bagian/Peran
Muhammad Vikhan Muharram	Metode Pergantian Generasi (Seleksi Survivor) & Kriteria Penghentian Evolusi (Loop)
Bimo Zachriansyah Wicaksono Hermawan	Ukuran Populasi, Rancangan Kromosom, dan Cara Decode & Metode Pemilihan Orangtua
Faris Siddiq Ramdan Putra	Metode Operasi Genetik (Pindah Silang dan Mutasi), Probabilitas Operasi Genetik (Pc dan Pm)