LAPORAN TUGAS PEMROGRAMAN

01 – SEARCHING : ALGORITMA GENETIKA

PENGANTAR KECERDASAN BUATAN

oleh:
Kelompok 19
Muhammad Fachry Gunawan / 1301204504
Galih Atha Fayi Khansa / 1301204028
Fakultas Informatika
Telkom University

I. Pendahuluan

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih “alamiah”. Algoritma genetika ini juga merupakan bagian dari Evolutionary Algorithm yaitu suatu algoritma yang mencontoh proses evolusi alami dimana konsep utamanya adalah individu-individu yang paling unggul akan bertahan hidup, sedangkan individu-individu yang lemah akan punah. Salah satu contoh aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi.

II. Soal

Lakukan analisis dan desain algoritma Genetic Algorithm (GA) serta mengimplementasikannya ke dalam suatu program komputer untuk mencari nilai x dan y sehingga diperoleh nilai minimum dari fungsi:

$$h(x,y) = \frac{(\cos x + \sin y)^2}{x^2 + y^2}$$

dengan domain (batas nilai) untuk x dan y:

$$-5 \leq x \leq 5 \text{ dan } -5 \leq y \leq 5$$

III. Pembahasan

a. Desain kromosom dan metode dekode-nya

```
def createpopulasi(jumlahpopulasi):
  populasi = []
  for i in range(jumlahpopulasi) :
    kromosom = []
    for j in range(9) :
      kromosom.append(random.randint(0,9))
    populasi.append(kromosom)

  return populasi
```

Tahap pertama, program mendesain kromosom dengan memakai representasi integer dan panjang kromosom yang digunakan pada program ini adalah sebanyak 9 gen dalam 1 kromosom

```
def decoding(rmax, rmin, g):
    x = [2**-i for i in range(1, len(g)+1)]
    return rmin + ((rmax - rmin) / sum(x) * sum([g[i] * g[i] for i in range(len(g))]))
```

Lalu kami melakukan decoding seperti yang tertera diatas

b. Ukuran populasi

```
JumlahGenerasi = int(input("Jumlah generasi yang diinginkan : "))
jumlahpopulasi = int(input("Jumlah populasi yang diinginkan : "))
```

Kami menggunakan inputan user untuk menghitung ukuran populasinya.

c. Metode pemilihan orangtua

Pada tahap ini, metode pemilihan orang tua yang digunakan pada program adalah metode tournament selection. cara kerja program ini dengan memilih secara acak calon orang tua, kemudian memilih kromosom yang memiliki nilai fitness terbesar yang akan menjadi orang tua.

```
def seleksiturnamen(populasi):
    tournament = []
    fitness = []
    max = 0
    pemenang = 0
    for i in range(5):
      tournament.append(random.choice(populasi))
      x = decoding(3.0, -2.0, tournament[i][:5])
      y = decoding(2.0, -1.0, tournament[i][5:])
      fitness.append(hitungfitness(x, y))
      if fitness[i] > max:
        max = fitness[i]
        pemenang = 1

    return pemenang
```

d. Metode operasi genetik (pindah silang dan mutasi)
Di tahap ini, Program yang dibuat menggunakan metode crossoveri satu titik dan penentuan titiknya dilakukan secara acak.

```
def crossover(ortu1, ortu2):
    pc = random.random()
    anak1 = ()
    anak2 = ()
    randint = random. uniform(0.0, 1.0)
    titikPotong = int(round(random.uniform(0, 5)))
    if randint <= pc:
        anak1 = ortu1[:titikPotong] + ortu2[titikPotong:]
        anak2 = ortu2[:titikPotong] + ortu1[titikPotong:]
    else :
     anak1 = ortu1
     anak2 = ortu2
    return anak1, anak2
```

Dan untuk implementasi mutasinya program menggunakan pengubahan satu atau beberapa nilai gen dari kromosom dengan probabilitas mutasi yang dipakai adalah 1%

```
def mutasi(a):
    pm = 0.5
    randnum = random.uniform(0.0, 1.0)
    if randnum > (1 - pm):
       posisi_mutasi = random.randint(0, len(a) - 1)
       if a[posisi_mutasi] == 1:
          a[posisi_mutasi] = 0
       else:
            a[posisi_mutasi] = 1
```

e. Probabilitas Operasi Genetik

kami menentukan pc = 1 dan pm = 0.5

f. Metode pergantian generasi (seleksi survivor)

Pada seleksi survivor program yang dibuat menggunakan metode steady state dimana metode ini akan mengganti kromosom dengan nilai fitness terkecil dengan anak dari hasil operasi genetik yang sudah dilakukan tadi.

```
[10] def Steadystate(gabungan):
         fitness = []
         for i in range(len(gabungan)):
            if gabungan[i] != None:
              x = decoding(2, -1, gabungan[i][:5])
              y = decoding(1, -1, gabungan[i][5:])
              fitness.append(hitungfitness(x, y))

         steadystate_list = sort_fitness(fitness)
         population = []

         for j in range(jumlahpopulasi):
             population.append(gabungan[steadystate_list[j]])
         return population, fitness
```

g. Kriteria penghentian evolusi (loop)
Pada program yang dibuat, user akan diminta menginputkan berapa banyak generasi dan populasi yang diinginkan. Kemudian loop pada program akan berhenti jika kriteria sudah terpenuhi sesuai dengan angka yang user inputkan sebelumnya.

```
JumlahGenerasi = int(input("Jumlah generasi yang diinginkan : "))
jumlahpopulasi = int(input("Jumlah populasi yang diinginkan : "))
population = createpopulasi(jumlahpopulasi)
for i in range(JumlahGenerasi):
    fitness = []
    anak = []
    for j in range(jumlahpopulasi):
        orangtual = seleksiturnamen(population)
        orangtua2 = seleksiturnamen(population)
        anak1 = population[orangtual]
        anak2 = population[orangtua2]
        crossover(anak1, anak2)
        anak1 = mutasi(anak1)
        anak2 = mutasi(anak2)
        anak. append(anak1)
        anak. append (anak2)
    gabungan = population + anak
    population, fitness = Steadystate(gabungan)

    print("---------------------------------------")
    print("Generasi ke", i+1)
    print('Nilai fitness: ', fitness[0])
    print('x:', decoding(5, -5, population[0][:5]))
    print('y:', decoding(5, -5, population[0][5:]))
    print()
    print('Kromosom terbaiknya adalah : ', population[0])
    print('Nilai fitness: ', fitness[0])
    print("---------------------------------------")
    print("\n")
```

h. Run program

Pada percobaan ini, user menginputkan 3 populasi dan 250 generasi dan untuk function tournament selection dibuat 6 calon orang tua. Hasilnya dapat dilihat sebagai berikut :

```
Jumlah generasi yang diinginkan : 3
Jumlah populasi yang diinginkan : 250
----------------------------------------
Generasi ke 1
Nilai fitness:  0.00012741152928921894
x: 36.29032258064516
y: 5.666666666666666

Kromosom terbaiknya adalah :  [1, 1, 1, 0, 1, 0, 0, 1, 0]
Nilai fitness:  0.00012741152928921894
----------------------------------------



----------------------------------------
Generasi ke 2
Nilai fitness:  0.7709750963835936
x: 36.29032258064516
y: 5.666666666666666

Kromosom terbaiknya adalah :  [1, 1, 1, 0, 1, 0, 0, 1, 0]
Nilai fitness:  0.7709750963835936
----------------------------------------



----------------------------------------
Generasi ke 3
Nilai fitness:  0.7709750963835936
x: 36.29032258064516
y: 5.666666666666666

Kromosom terbaiknya adalah :  [1, 1, 1, 0, 1, 0, 0, 1, 0]
Nilai fitness:  0.7709750963835936
----------------------------------------
```

IV. Kesimpulan

Algoritma Genetika paling sering digunakan dalam masalah optimasi di mana kita harus memaksimalkan atau meminimalkan nilai fungsi tujuan yang diberikan di bawah serangkaian kendala yang diberikan. Panjang kromosom akan berpengaruh terhadap nilai maksimumnya, semakin panjang kromosomnya maka nilai maksimumnya juga akan semakin besar.

https://drive.google.com/file/d/1Vmbgd_ppKQi0u2MxMCa0LwtuhcP7jy6D/view?usp=sharing