Laporan Observasi GA
Mohammad Sani Rafsanjani
IF-42-GAB01

A. Analisis Genetic Algorithm (GA) ini adalah algoritma yang di dasar kan, atau di konsepkan berdasarkan genetika yang ada pada biologi. Sehingga tahapan-tahapan yang adapun dinamakan seperti saat ingin mengawinkan 2 genetika yang akan di sebut crossover kedepannya pada laporan ini. Pada tugas kecerdasan buatan kali ini diberikan sebuah fungsi untuk mencari nilai minimum

$$h(x_1, x_2) = \cos(x_1)\sin(x_2) - \frac{x_1}{(x_2^2 + 1)}$$

dengan batasan $-1 \leq x_1 \leq 2$ dan $-1 \leq x_2 \leq 1$.

B. Strategi penyelesaian masalah

Strategi penyelesaian masalah yang untuk mendapatkan nilai optimal pada permasalahan ini dengan menggunakan metode genetic algorithm (GA), algoritma ini banyak dipilih dikarenakan variasi yang banyak sehingga kita sebagai pengamat bisa mendapatkan sampel yang banyak dan bervariasi. Selain itu pula genetic algorithm (GA) ini dalam menyelesaikan masalahnya tidak diselesaikan per satu masalah, namun diselesaikan secara jamak/populasi. Adapun tahapan pemecahan masalahnya adalah berikut

1. Membuat kromosom dengan panjang yang telah di tertentu terlebih dahulu, dan sebanyak populasi tertentu pula.disini saya set dengan panjang 6 dengan bilangan integer random antara 0 sampai 9. Lalu membuat daftar populasi sebanyak 6 kromosom

```
#kromosom
def kromosom():
  krom=[]
  for i in range(6):
    krom.append(random.randint(0,9))
  return krom

#populasi
def populasi():
  pop=[]
  for i in range(6):
    pop.append( kromosom())
  return pop
```

Kemudian lakukan dekode kromosom untuk nilai x1 dan x2

```
#decode kromosom x1 dan x2
def penotipe(krm, max, min): #Batasan     -1 <= X <= 2  ,  -1 <= X <= 1
  b_atas = 0
  b_bwh = 0
  for i in range(len(krm)):
    g = (krm[i])
    b_atas += (g*(10**-(i+1)))
    b_bwh += (9*(10**-(i+1)))
  x = min + (((max-min)*b_atas)/b_bwh)
  return x
```

2. Melakukan perhitungan Fitness
   Agar mendapatkan nilai fitness, terlebih dahulu menghitung nilai dari fungsi x1 x2 terlebih dahulu

```
#fungsi nilai minimum
def n_min(x1, x2): #f(x1,x2)
  f = math.cos(x1) *  math.sin(x2) - (x1 / (x2**2 + 1))
  return f
```

Kemudian memasukkannya ke dalam rumus
fitness 1/(f+h), dengan h= 0.01

```
#fungsi fitness minimal
def fungsi_min(f):
   fitness = 1 / (f + 0.01)
   return fitness
```

3.Seleksi Orang Tua
Dalam pemilihan orang tua terdapat banyak cara, Namun disini saya menggunakan salah satunya yaitu dengan cara Roulette Wheel Selection

```
#seleksi orang tua
def SeleksiOrangtua(pop,fitness,total):
   r = random.random()
   i = 0
   while (r>0):
      r -= fitness[i]/total
      i += 1
      if (i == (len(pop)-1)):
            break
   orangtua = pop[i]
   return orangtua
```

4. Pindah Silang(CrossOver)
   Hasil dari pemilihan orang tua sebelumnya kemudian di crossover dengan probabilitas 0.7

```
#pindah silang
def f_pindahsilang(orangtua1,orangtua2):
  pindah_s1, pindah_s2 = [], []
  n_ps = []
  chance = random.random()
  if (chance < 0.9):
    point = random.randint(0,5)
    pindah_s1[:point] = orangtua1[:point]
    pindah_s1[point:] = orangtua2[point:]
    pindah_s2[:point] = orangtua2[:point]
    pindah_s2[point:] = orangtua1[point:]
    n_ps.append(pindah_s1)
    n_ps.append(pindah_s2)
  else:
    n_ps.append(orangtua1)
    n_ps.append(orangtua2)
  return n_ps
```

5.Hasil dari crossover sebelumnya kemudian dimutasi dengan probabilitas 0.3, kemudian akan di cek per allel dengan probabilitas 0.25 untuk nilai nya diganti dengan bilangan integer random antara 0 sampai 9

```
#mutasi
def mutasi(pindah_s1,pindah_s2):
  chance = random.random()
  if (chance < 0.3):
      for i in range (len(pindah_s1)):
        p = random.random()
        if (p < 0.25):
          pindah_s1[i] = random.randint(0,9)
      for i in range (len(pindah_s2)):
        p = random.random()
        if (p < 0.25):
          pindah_s2[i] = random.randint(0,9)
  n_ps = []
  n_ps.append(pindah_s1)
  n_ps.append(pindah_s2)
  return n_ps
```

6. Untuk mendapatkan kromosom yang terbaik dari semua generasi yaitu dengan membandingkan nilai fitness paling besar dari seluruh kromosom, dan didapatkan nilai fungsi minimal nya

```
#populasi terbaik
def bestpop(pop):
    max_fitness = -9999
    id = []
    for i in range(len(pop)):
        id = pop[i]
        a, b = pot(id)
        x1 = penotipe(a, -1, 2)
        x2 = penotipe(b, -1, 1)
        f = n_min(x1, x2)
        fitness = fungsi_min(f)
        if (fitness>max_fitness):
            max_fitness = fitness
            n_maxID = id
        return n_maxID
```

7.Program Utama,pergantian generasi

```
#main program
pop = populasi()
generasi = 1
while(generasi<4):
  id = []
  fitness = []
  daftar_f = []
  populasi_baru = []
  anak= []
  best = theBest(pop)
  total = 0
  for i in range(len(pop)):
    id = pop[i]
    a, b = split(id)
    x1 = penotipe(a, 3, -3)
    x2 = penotipe(b, 2, -2)
    f = n_min(x1, x2)
    fitness = fungsi_min(f)
    daftar_f.append(fungsi_min(f))
    total += fitness
  for j in range(len(pop)//2):
    orangtua1 = SeleksiOrangtua(pop,daftar_f,total)
    orangtua2 = SeleksiOrangtua(pop,daftar_f,total)
    anak = f_pindahsilang(orangtua1,orangtua2)
    anak = mutasi(anak[0],anak[1])
    populasi_baru.append(anak[0])
    populasi_baru.append(anak[1])
  best = theBest(populasi_baru)
  pop = populasi_baru
  generasi+=1
```

8.output program kromosom terbaik :

```
==================================
Kromosom Terbaik :
Genotype :  [9, 3, 6, 7, 9, 6]
Fenotype :  -0.810810810810811 -0.5935935935935936
Nilai Fungsi :  0.21421758911865868
Nilai Fitness :  4.459953404774091
==================================
```