

Analisis, Desain, dan Implementasi Algoritma Genetika Dalam Menemukan Nilai Minimum Dari Suatu Fungsi

Mochamad Gia Tama Nugraha

1301172739

IFIK-41-03

Alamat : Jalan Sukabirus NO.39

e-mail: giatama@student.telkomuniversity.ac.id

1. PENDAHULUAN

Suatu fungsi dalam istilah matematika merupakan pemetaan setiap anggota sebuah himpunan kepada anggota himpunan yang lain. Konsep fungsi adalah salah satu konsep dasar dari matematika dan setiap ilmu kuantitatif. Anggota himpunan yang dipetakan dapat berupa apa saja, namun biasanya yang dibahas adalah besaran matematika seperti bilangan riil. Dalam laporan ini, fungsi yang akan dikaji adalah

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

dengan batasan atas $-3 \leq x_1 \leq 3$ dan $-2 \leq x_2 \leq 2$.

Masalah yang terdapat pada laporan ini adalah menentukan nilai minimum dari fungsi tersebut dengan menggunakan algoritma genetika. Dengan menggunakan algoritma genetika diharapkan dapat menemukan solusi nya dengan waktu yang cepat. Tidak hanya dengan waktu

2. PENJELASAN

A. Roulette Wheel Selection

Pada AG (Algoritma Genetika) dikenal berbagai macam metode tentang bagaimana cara mencari kromosom yang akan dijadikan sebagai parent (orang tua) yaitu salah satu dari metode tersebut yaitu roulette wheel selection. Pada seleksi ini, orang tua dipilih berdasarkan nilai fitness mereka. Jika lebih baik kualitas suatu kromosom, maka lebih besar peluangnya untuk terpilih sebagai parent (orang tua). Probabilitas suatu individu terpilih untuk dilakukan penyilangan (cross over) sebanding dengan nilai fitness-nya.

B. Nilai Parameter Algoritma Genetika

Pertama, sebagai inisialisasi terdapat variabel sebagai berikut :

- 1| SumGenerasi = 100
- 2| SumPopulasi = 350
- 3| SumDNA = 10
- 4| crossRate = 0.65
- 5| MutasiRate = 0.015

Variabel *SumGenerasi* di set agar memiliki jumlah populasi sebanyak 100 generasi. Variabel *SumPopulasi* di set sebesar itu agar memiliki jumlah populasi sebanyak 350 populasi. Variabel *SumDNA* di set sebesar itu agar memiliki jumlah DNA atau GEN sebesar 10. Lalu, variabel *crossRate* dan *MutasiRate* sebesar masing-masing di set sebesar 0.65 dan 0.015.

Lalu, dilanjutkan dibaris ke-12 dan pada baris ke-29 pada source code terdapat fungsi *BinerToDesimalX*(populasi) dan fungsi *BinerToDesimal Y*(populasi).

```
def BinerToDesimalX(populasi):  
    MaxSum = 0  
    panjangDNAx = (SumDNA/2)+1  
    hasilPanjangDNAx = int(panjangDNAx)  
    x = np.zeros([SumPopulasi], dtype = float)  
    for i in range(1, hasilPanjangDNAx):  
        MaxSum = MaxSum + 2**(-i)  
    for i in range(SumPopulasi):  
        for j in range(1, hasilPanjangDNAx):  
            x[i] = x[i] + populasi[i][j-1]*(2**(-j))  
            x[i] = (-3+((3-(-3))/MaxSum)*x[i])  
    return x
```

Fungsi tersebut memiliki fungsi untuk mendecode nilai *x1* dari biner ke desimal.

Dengan memulai dari awal biner sampai dengan jumlah keseluruhan biner dibagi 2.

```
def BinerToDesimalY(populasi):
    MaxSum = 0
    panjangDNAy = SumDNA/2
    hasilPanjangDNAy = int(panjangDNAy)
    y = np.zeros([SumPopulasi], dtype = float)
    for i in range(hasilPanjangDNAy, SumDNA):
        MaxSum = MaxSum + 2**(-i)
    for i in range (SumPopulasi):
        for j in range (hasilPanjangDNAy, SumDNA):
            y[i] = y[i] + populasi[i][j] * (2**(-j))
            y[i] = (-2+((2-(-2))/MaxSum)*y[i])
    return y
```

Fungsi tersebut memiliki fungsi untuk mendecode nilai x2 dari biner ke desimal. Dengan memulai dari jumlah keseluruhan biner dibagi 2 sampai akhir dari jumlah gen.

Selanjutnya, ada fungsi nilaiDecode(). Yaitu berfungsi untuk menggabungkan hasil dari decode biner ke desimal dari nilai x1 dan nilai x2. Lalu, dijumlahkan menggunakan rumus fungsi :

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_2^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

```
def nilaiDecode(x,y):
    nilai = (4-2.1*(x**2)+((x**4)/3))*(x**2)+x*y+((-4)+(4*(y**2)))*(y**2)
    return nilai
```

Selanjutnya, ada fungsi fitness(). Yaitu berfungsi untuk mencari nilai fitness dari nilai fungsi diatas .

```
def fitness(nilai):
    fitness = abs(1/nilai)
    return fitness
```

Selanjutnya, ada fungsi RouletteWheel(). Yaitu berfungsi untuk mencari individu/kromosom yang terbaik agar dijadikan sebagai parent untuk melakukan crossover.

```
def RouletteWheel(populasi):
    panjangFitness =
    fitness(nilaiDecode(BinerToDesimalX(populasi),BinerToDesimalY(populasi)))

    jumlahFitness = panjangFitness.sum()
    kumulatifFitnes = 0
    nilaiRandom = np.random.uniform(0,1)
    i = 0
    while i <= len(populasi):
        kumulatifFitnes = kumulatifFitnes +
        panjangFitness[i]
```

```
if ((kumulatifFitnes/jumlahFitness)>nilaiRandom) :
    index = i
    break
i = i+1
return index
```

Selanjutnya, ada fungsi CrossOver(). Yaitu berfungsi untuk melakukan penyilangan antara kedua parent.

```
def CrossOver(parent1,parent2,jumlahDNA,populasi):
    Parent1 = populasi[parent1]
    Parent2 = populasi[parent2]
    nilaiRand = np.random.uniform(0,1)
    if (nilaiRand < crossRate):
        titikPotong = 1+
        int(np.ceil(np.random.randint(SumDNA-1)))
        child1 =
        np.concatenate([Parent1[0:titikPotong],Parent2[titikPotong+0:j
        umlahDNA]])
        child2 =
        np.concatenate([Parent2[0:titikPotong],Parent1[titikPotong+0:j
        umlahDNA]])
        populasi[parent1] = child1
        populasi[parent2] = child2
    return populasi
```

Selanjutnya, ada fungsi Mutasi(). Yaitu berfungsi untuk melakukan mutasi kepada kromosom yang telah di crossover.

```
def Mutasi(kromosom,jumlahDNA,MutasiRate):
    nilaiRand = np.random.uniform(0,1)
    mutasiKrom = kromosom
    for i in range (1,SumDNA):
        if (nilaiRand < MutasiRate):
            if (kromosom[i] == 0):
                mutasiKrom[i] = 1
            else:
                mutasiKrom[i] = 0
```

Selanjutnya, ada fungsi Minimum(). Yaitu berfungsi untuk mencari nilai minimum pada nilai fungsi.

```
def Minimum(nilai):
    for index in nilai:
        minimum = 99
        if index < minimum:
            minimum = index
    return minimum
```

C. Hasil Akhir

Berikut screenshot dari hasil program :

