Tugas Matakuliah Kecerdasan Komputasional A

Maksimasi Fungsi Presisi Tertentu

Nama : Tince Etlin Tallo

NIM : 15/388504/PPA/04943

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Gadjah Mada

Studi kasus: Maksimasi fungsi dengan presisi tertentu

Max $f(x_1, x_2) = 19 + x_1 \sin(x_1\pi) + (10 - x_2)\sin(x_2\pi)$ dengan $-5{,}0 \leq x_1 \leq 9{,}8$ dan $0{,}0 \leq x_2 \leq 7{,}3$

Dalam kasus ini menggunakan siklus (μ, λ) jadi tidak menggunakan rekombinasi dalam proses reproduksi. Seleksi menggunakan *elitism selection* yaitu hanya melibatkan individu dalam offspring, individu induk dalam populasi tidak dilibatkan. Sedangkan besarnya nilai λ sebesar 7μ.

- Representasi kromosom

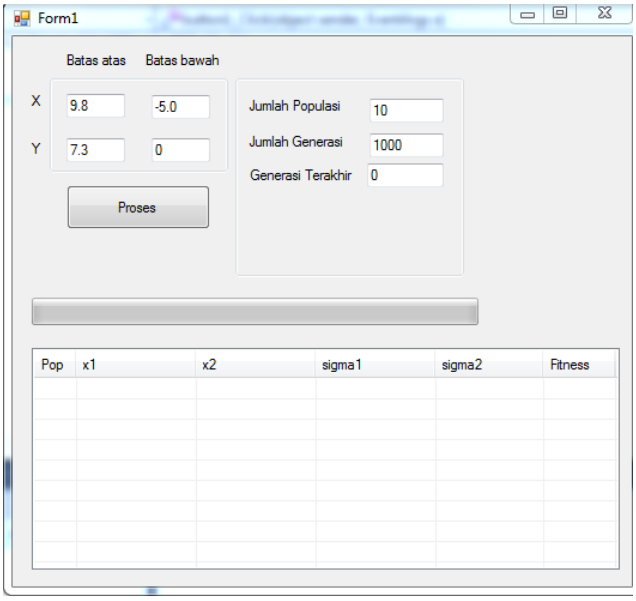
  x_1 dan x_2 adalah gen string kromosom yang menyatakan variabel keputusan, dan terdapat juga parameter tambahan yang melekat pada setiap *chromosome* yaitu σ (*sigma*). Nilai ini menyatakan level mutasi untuk *chromosome* tersebut. Nilai ini akan ikut berubah secara adaptif sepanjang generasi. Jika P adalah satu kromosom maka $P = (x_1, x_2, \sigma_1, \sigma_2)$.

- Inisialisasi

  Populasi inisial dibangkitkan secara random. Nilai *x1* dan *x2* dibangkitkan dalam rentang variabel $-5{,}0 \leq x_1 \leq 9{,}8$ dan $0{,}0 \leq x_2 \leq 7{,}3$. Dalam aplikasi ini, nilai x_1 dan x_2 dapat diinputkan oleh *user* (dinamis) seperti dapat dilihat dalam Gambar 2. Nilai σ1 dan σ2 dibangkitkan dalam rentang [0,1]. Gambar 1 merupakan inisialisasi awal kromosom.

```
for (int j = 0; j < jp;j++ ){
    //for (int i = 0; i < mx + my; i++)
    pop[j] = Math.Round(r.Next(Convert.ToInt32(xb * Math.Pow(10, k)), Convert.ToInt32(xa * Math.Pow(10, k)))/
        Math.Pow(10,k),k);
    pop1[j] = Math.Round(r.Next(Convert.ToInt32(yb * Math.Pow(10, k)), Convert.ToInt32(ya * Math.Pow(10, k)))/
        Math.Pow(10,k),k);
    pop2[j] = Math.Round(r.NextDouble(),k);
    pop3[j] = Math.Round(r.NextDouble(),k);
}
```

Gambar 1



Gambar 2.

- Reproduksi

Dalam kasus ini rekombinasi tidak digunakan maka hanya mutasi yang berperan menghasilkan *offspring*. Misalkan $P = (x_1, x_2, \sigma_1, \sigma_2)$ adalah individu yang terpilih untuk melakukan mutasi, maka dihasilkan offspring $P' = (x'_1, x'_2, \sigma'_1, \sigma'_2)$ sebagai berikut:

$x'_1 = x_1 + \sigma_1 N(0,1)$ dan $x'_2 = x_2 + \sigma_2 N(0,1)$.

Nilai $N(0,1)$ diperoleh dengan menggunakan rumus $N(0,1) = \sqrt{-2.\ln r_1}\sin 2\pi r_2$ dengan nilai r_1 dan r_2 di-*random* pada interval [0,1].

```
double N() {
    double r1,r2;
    r1 = r.NextDouble();
    r2 = r.NextDouble();
    return Math.Round(Math.Sqrt(-2*Math.Log(r1))*Math.Sin(2*Math.PI*r2),k);
}
```

Sedangkan untuk nilai σ dinaikkan jika paling sedikit 20% hasil mutasi yang menghasilkan individu yang lebih baik dari induknya ($\sigma' = \sigma \times 1{,}1$). Jika tidak maka nilai σ diturunkan ($\sigma' = \sigma \times 0{,}9$). Misalnya di dalam aplikasi ini nilai $\lambda = 7\mu$ dan dimasukkan nilai $\mu = 4$ maka setiap individu dalam populasi akan dihasilkan 7 *offspring*. Pada kasus ini, nilai σ akan dinaikkan jika ada setidaknya 2 *offspring* yang lebih baik.

```
for (int i = 0; i < jp;i++)
{
    int z = 0;
    for (int j = 0; j < 7; j++)
    {
        pop4[i * 7 + j] = pop[i] + pop2[i] * N();
        pop5[i * 7 + j] = pop1[i] + pop3[i] * N();
        fit1[i * 7 + j] = fitkrom(pop4[i * 7 + j], pop5[i * 7 + j]);
        if (fit1[i * 7 + j] > fit[i])
        {
            z++;
        }

    }
    if (z >= 2)
    {
        pop2[i] = Math.Round(1.1 * pop2[i], k); ;
        pop3[i] = Math.Round(1.1 * pop3[i], k);
    }
```

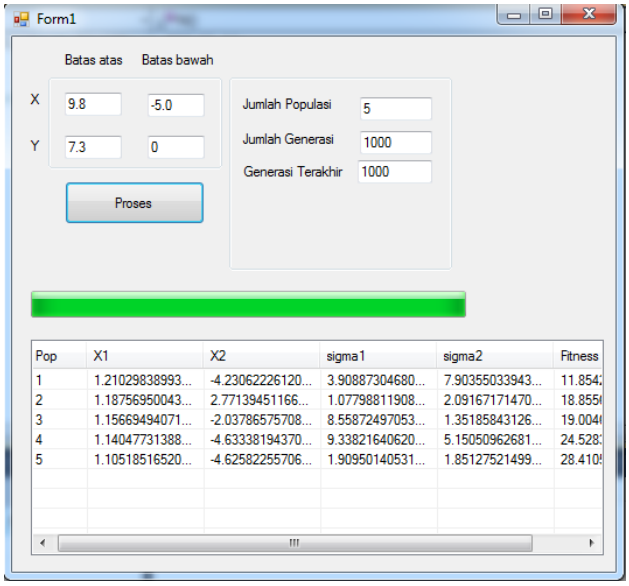
```
    }
    else {
        pop2[i] = Math.Round(0.9 * pop2[i], k);
        pop3[i] = Math.Round(0.9 * pop3[i], k);
    }
}
Array.Sort(fit1);

for (int i = 0; i < jp*7;i++)
{
   int j=Array.IndexOf(fit1,fitkrom(pop4[i],pop5[i]));
   if (j>jp*6-1) {
       pop[jp * 7 - j-1] = pop4[i];
       pop1[jp * 7 - j-1] = pop5[i];
   }
}
cg++;
pb1.Value++;
```

- Seleksi

  Seleksi menggunakan *elitism selection* hanya melibatkan individu dalam *offspring*, individu induk dalam populasi tidak dilibatkan.

- *Terminal Condition*

  Dalam kasus ini, iterasi berhenti sampai generasi *n*. Nilai n bisa ditentukan oleh user dengan diinputkan ke dalam aplikasi. Tampilan aplikasi yang dibuat dapat dilihat pada Gambar 2.


Form1
Batas atas
Batas bawah
X
9.8
-5.0
Y
7.3
0
Proses
Jumlah Populasi
5
Jumlah Generasi
1000
Generasi Terakhir
1000
Pop
X1
X2
sigma1
sigma2
Fitness
1
1.21029838993...
-4.23062226120...
3.90887304680...
7.90355033943...
11.854
2
1.18756950043...
2.77139451166...
1.07798811908...
2.09167171470...
18.855
3
1.15669494071...
-2.03786575708...
8.55872497053...
1.35185843126...
19.004
4
1.14047731388...
-4.63338194370...
9.33821640620...
5.15050962681...
24.528
5
1.10518516520...
-4.62582255706...
1.90950140531...
1.85127521499...
28.410

Gambar 2