

KECERDASAN BUATAN
LAPORAN TUGAS PARALEL 1
ALGORITMA GENETIKA

1. Kromosom

Panjang kromosom yang digunakan pada tugas paralel 1 ini sepanjang 8, dengan setiap gen akan diisi dengan angka acak dari 0-10. Dipilih 8 karena untuk menyingkat waktu komputasi. Terdapat pada fungsi `chrom()` dalam file program `IF4102_1301174354.py`.

2. Populasi

Banyak kromosom dalam satu populasi adalah 10. Dipilih 10 karena agar tidak memakan banyak waktu saat proses komputasi. Terdapat pada fungsi `populesyen()` dalam file program `IF4102_1301174354.py`.

3. Fitness

Dalam tugas paralel 1 ini terdapat 2 tahap dalam mencari *fitness*.

Pertama kromosom yang di-*generate* terlebih dahulu dilakukan proses *encode* dengan cara memotong kromosom menjadi 2 bagian yang sama panjang, kemudian masing-masing bagian dimasukan ke fungsi berikut :

$$x = r_{min} + \frac{r_{max} - r_{min}}{\sum_{i=1}^N 9 * 10^{-i}} (g_1 * 10^{-1} + g_1 * 10^{-2} + \dots + g_N * 10^{-N})$$

Rmax dan Rmin adalah batas atas dan batas bawah tiap x_1, x_2 . Dalam tugas paralel 1 ini batas atas dan batas bawah tiap x adalah $-3 \leq x_1 \leq 3$ dan $-2 \leq x_2 \leq 2$.

Kemudian langkah selanjutnya adalah memasukan hasil x_1, x_2 ke dalam fungsi yang akan dicari minimasinya, yaitu :

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

Setelah mendapat hasil fungsi diatas, masukan hasil fungsi tersebut kedalam rumus berikut

$$: \frac{1}{f(x_1, x_2) + a}$$

Dengan a adalah angka positif bebas(dalam tugas ini digunakan 0.01) sehingga didapat *fitness* dari sebuah kromosom. Semua penjelasan fungsi *fitness* diatas tertuang dalam fungsi `fitness()` dalam file program `IF4102_1301174354.py`

4. Parent Selection

Dalam tugas paralel 1 ini digunakan metode *Tournament* untuk proses seleksi. Metode *Tournament* adalah proses pemilihan *parent* dengan cara membandingkan fitness tiap kromosom dengan fitness kromosom lainnya(yang dipilih secara acak) dalam populasi yang sama yang kemudian akan mendapatkan 2 *parent* terbaik dalam sebuah populasi. Terdapat pada fungsi `parent_sel()` dalam file program `IF4102_1301174354.py`.

5. Crossover

Proses *crossover* adalah proses untuk menggabung 2 *parent* yang telah diseleksi dengan cara memotong kromosom *parent* dititik tertentu kemudian menukar bagian yang terpotong tersebut ke *parent* lainnya dengan probabilitas yang ditentukan. Dalam tugas paralel 1 ini probabilitas yang digunakan adalah 0.02. Proses ini menghasilkan 2 kromosom baru yang disebut *child*. Terdapat pada fungsi `crossover()` dalam file program `IF4102_1301174354.py`.

6. Mutation

Proses untuk memutasi *child* yang didapat dari proses *crossover* dengan cara mengganti gen kromosom *child* pada index acak dengan angka acak. Terdapat pada fungsi `mutation()` dalam file program `IF4102_1301174354.py`.

7. Generate New Population

Dalam genetika algoritma proses awalnya adalah membuat populasi yang kemudian populasi itu akan dilakukan proses 3-6 hingga dalam populasi baru hanya kromosom terbaik yang akan muncul. Terdapat pada fungsi `newpop_generator()` dan Main program dalam file program `IF4102_1301174354.py`.

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
best local : [7, 1, 2, 3, 3, 5, 7, 9]
fitness : 23.929271542498327

best local : [6, 2, 8, 4, 5, 10, 7, 9]
fitness : 1.0652488383257799

best local : [6, 10, 3, 0, 7, 5, 8, 7]
fitness : 0.18555484287394555

best local : [3, 3, 2, 5, 7, 6, 4, 5]
fitness : 0.8897664285821997

best local : [2, 4, 7, 2, 7, 1, 7, 4]
fitness : 3.5037040169411444

best local : [2, 5, 2, 2, 1, 1, 7, 4]
fitness : 0.04853178381285142

best local : [2, 1, 7, 2, 1, 6, 3, 4]
fitness : 0.09000854775823555

best : [7, 1, 2, 3, 3, 5, 0, 9]
fitness : 23.929271542498327
x1,x2 : 1.2742274227422739 -0.5962596259625963
PS C:\Users\ario>
```