

Laporan Tugas 1
CCH3F3 Kecerdasan Buatan 2018

Oleh :
Muhammad Faris Luthfi (1301164502)
IF-40-09



S-1 Teknik Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2018

I. Analisis Masalah

Diberikan persamaan sebagai berikut :

$$f(x_1, x_2) = - \left| \sin(x_1) \cos(x_2) \exp \left(\left| 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right|$$

Persamaan tersebut adalah persamaan trigonometri, di mana terdapat kombinasi nilai x_1 dan x_2 yang menghasilkan nilai minimum pada fungsi. Salah satu metode untuk mencari nilai minimum tersebut adalah algoritma *Simulated Annealing* (selanjutnya disebut SA). Dengan algoritma SA, dapat dipetakan hasil fungsi dari x_1 dan x_2 ; di mana x_1 dan x_2 adalah nilai acak; sebagai state-state yang akan dicari nilai paling minimumnya.

II. Strategi Penyelesaian Masalah

Langkah pertama yang saya lakukan adalah memilih bahasa pemrograman. Bahasa pemrograman yang saya gunakan dalam tugas ini adalah Python, karena penggunaan fungsi-fungsi matematika (contohnya *random*) jauh lebih mudah dibandingkan bahasa yang lain yang saya ketahui. Adapun dapat digunakan library *matplotlib.pyplot* sebagai visualisasi, sehingga dapat semakin mudah dimengerti ke mana saja titik yang dicari ketika melakukan pencarian dengan menggunakan SA.

Transformasi fungsi (dengan *function* rumus()):

```
def rumus(x1, x2):  
    return -1*(abs(math.sin(x1)*math.cos(x2)*math.exp(abs(1-(math.sqrt(math.pow(x1,2)+math.pow(x2, 2))/math.pi))))))
```

Pembangkitan bilangan acak :

```
x1 = random.uniform(-10,10)  
x2 = random.uniform(-10,10)
```

Dalam sisi *annealing schedule* dipilih suhu awal sebagai *currentT*, suhu akhir sebagai *stopT*, dan *cooling rate* sebagai *coolRate*. Angka 1000 dipilih untuk *currentT* dan 1 untuk *stopT* karena saya anggap memiliki rentang jarak yang cukup untuk 'melelehkan' permasalahan, sehingga mendapat hasil akhir minimum. *Cooling rate* yang dipilih 0,999, karena saya anggap penurunan suhu tidak terlalu cepat maupun lama untuk dituntaskan program.

Inisialisasi titik awal, *best so far*, dan BSFA dan BSFB untuk *best so far* x_1 dan x_2 .

```
currentState = rumus(x1,x2)  
BSF = currentState  
BSFA = x1  
BSFB = x2
```

Algoritma perulangan yang saya gunakan :

Ulangi sampai suhu minimum :

Generate state baru

Evaluasi cost ΔE $f(\text{baru}) - f(\text{sekarang})$

Jika $\Delta E < 0$

State sekarang digantikan *state* baru

Best so far digantikan *state* baru

Else

Bandingkan probabilitas dengan variabel *random* 0 sampai 1. Jika probabilitas < angka *random*, maka terima *state* baru

Turunkan suhu dengan *cooling rate*

Dalam membangkitkan variabel baru, digunakan :

```
def generate(x1, x2):  
    x1n = random.uniform(x1-(currentT/80), x1+(currentT/80))  
    x2n = random.uniform(x2-(currentT/80), x2+(currentT/80))  
    if(x1n < -10):  
        x1n = -10  
    elif(x1n > 10):  
        x1n = 10  
    if(x2n < -10):  
        x2n = -10  
    elif(x2n > 10):  
        x2n = 10
```

di mana $x1n$ dan $x2n$ adalah $x1$ dan $x2$ baru. Pembatasan dalam *range* $x1$ dan $x2$ baru dilakukan agar pencarian tidak melebar dari pencarian sebelumnya, dan digunakan pembatas $\text{currentT}/80$ karena nilai tersebut dinamis dan konsisten. Dalam melakukan pembatasan, juga dibatasi dengan batas bawah -10, dan batas atas 10 agar tidak menyimpang dari batas rumus yang telah ditentukan.

III. Hasil Terbaik Program

```
1.0003010212192145      -19.208502318634224  
x1 : 8.055123479801372    x2 : 9.664466569871674  
Minimum Value: -19.208502318634224
```

Nilai minimum : -19,208502318634224

dengan $x1$ 8,0551234579801372 dan $x2$ 9,664466569871674

Pemetaan titik dalam pencarian :

