

Laporan Tugas 1 Simulated Annealing

Disusun untuk memenuhi tugas Mata Kuliah Artificial Intelligence

Oleh :

Muhamad Restu Assegaf (NIM 1301164198)



**TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2018**

1. Simulated Annealing

Simulated Annealing (SA) adalah salah satu algoritma untuk optimisasi yang bersifat generik. Berbasiskan probabilitas dan mekanika statistik, algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan.

2. Analisis Masalah

Disini Simulated Annealing digunakan untuk menganalisis data yang mendekati suhu yang ditentukan yaitu -19. Analisis terhadap data pengukuran berulang tersebut dilakukan untuk menyelidiki apakah akan mencapai nilai yang ditargetkan. Bahasa pemrograman yang digunakan pada perancangan ini adalah Python untuk mencari data nilai target yang akan dicari.

3. Strategi Pengujian

- a. Menentukan data yang akan di uji

Data yang diberikan :

- Suhu awal (T) : 100
- Parameter pengali T (c) : 0.99999 (akan berubah-ubah setiap 0,99999)
- x1,x2 & y1,y2 : (-10,10)

Maximum dan Minimumnya dengan menggunakan rumus:

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

Bentuk algoritma :

```
-1*(math.fabs(math.sin(x1)*math.cos(x2)*math.exp(math.fabs(1-  
(math.sqrt(pow(x1,2) + pow(x2,2))/ math.pi))))))
```

- b. Mendapatkan solusi awal dengan rumus diatas.
- c. Melakukan pengulangan semasih T lebih besar dari 0,0001 (batas perulangan yang saya berikan).
- d. Mendapatkan solusi baru dengan rumus diatas
- e. Jika solusi baru lebih kecil dari solusi awal maka nilai solusi awal di ganti dengan nilai solusi baru. Sebaliknya, jika solusi baru lebih besar dari solusi awal maka cari selisih dengan solusi baru dikurangi solusi awal, setelah itu menampung nilai dari $\text{math.exp}((-1*\text{selisih})/T)$, kemudian memberi nilai random.

- f. Jika nilai random lebih kecil dari nilai yang di tamping maka nilai solusi awal di ganti dengan nilai solusi baru.
- g. T(suhu) akan di ganti dengan nilai T*c, dan melakukan pengulangan sampai selesai.
- h. Terakhir, akan menampilkan nilai tujuan yang dicari.

4. Source Code

```
tugas ai.py - C:\Users\USER\Documents\tugas ai.py (3.7.0)
File Edit Format Run Options Window Help

import math
import random

T =100
c=0.99999

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

solusiAwal =-1*(math.fabs(math.sin(x1)*math.cos(x2)*math.exp(math.fabs(1-(math.sqrt(pow(x1,2) + pow(x2,2))/ math.pi))))))

while (T>0.0001) :
    y1=random.uniform(-10,10)
    y2=random.uniform(-10,10)
    solusiBaru=-1*(math.fabs(math.sin(y1)*math.cos(y2)*math.exp(math.fabs(1-(math.sqrt(pow(y1,2)+pow(y2,2))/math.pi))))))
    if(solusiBaru<solusiAwal):
        solusiAwal = solusiBaru
    else:
        selisih = solusiBaru - solusiAwal
        tamping =math.exp((-1*selisih)/T)
        r=random.random()
        if (r< tamping):
            solusiAwal = solusiBaru

    T=T*c
print(solusiAwal)
```

5. Hasil Pengujian

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\USER\Documents\tugas ai.py =====
-19.207742515639385
>>> |
```

6. Akurasi Simulated Annealing

Akurasi simulated annealing dapat dihitung dengan rumus :

$$\text{Akurasi SA} = \left(1 - \left| \frac{f_A - f_R}{f_R} \right| \right) \times 100\%$$

f_A : -19.2007

f_R : -19.1365

Maka Akurasi SA = $\left(1 - \left| \frac{-19.2007 - (-19.1365)}{-19.1365} \right| \right) \times 100\%$

Akurasi SA = 21.2007

0.9966451545

19.2007