

TUPRO 1 Artificial Intelligence

[Simulated Annealing]

A. Studi Kasus

Artificial Intelligence adalah bagaimana mesin/komputer bertingkah layaknya manusia dan berfikir layaknya manusia. Dalam AI ada 4 pokok bahasan yaitu searching, reasoning, planning dan learning.

Dalam tugas program 1 ini, pokok bahasan yang ditugaskan adalah searching dengan algoritma Simulated Annealing pada sebuah fungsi matematis dengan 2 variable dengan range $[-10,10]$ pada setiap variable dengan tujuan untuk mencari nilai paling minimum yang dihasilkan model matematis tersebut. Simulated Annealing sendiri adalah salah satu algoritma searching dengan menerapkan analogi proses dalam bidang Metalurgi yaitu proses penguatan logam.

Penerapan SA dapat digunakan untuk mencari nilai global optimum dalam suatu domain abstrak yang luas. Pada bagian B akan dijelaskan bagaimana menerapkan algoritma SA dalam mencari nilai global minimum.

B. Analisa dan Pembahasan

a. Algoritma Simulated Annealing

Simulated Annealing adalah sebuah teknik optimasi kombinatorial, contohnya memperkecil fungsi dari sebuah fungsi yang memiliki banyak variable (Rob A. Ruternbar, 1989). Lebih jelasnya “combinatorial optimization problem” adalah untuk menemukan konfigurasi dari parameter $X = [X_1, X_2, X_3 \dots X_n]$ yang dapat meminimalkan hasil fungsi $f(x)$.

Cara kerja Simulated Annealing adalah dengan menerapkan Metropolis Algorithm dengan Boltzmann distribution. Idennya adalah pada manajemen temperature untuk bisa menemukan global optimum. Pada suhu tinggi, peluang untuk mendapat loncatan naik adalah tinggi, dan pada suhu rendah peluangnya rendah. Ada unsur randomness pada algoritma ini.

```
M = number of moves to attempt;
T = current temperature;
for m=1 to M {

    Generate a random move, e.g., move a particle;
    Evaluate the change in energy,  $\Delta E$ ;
    if (  $\Delta E < 0$  ) {
        /* downhill move: accept it */
        accept this move, and update configuration;
    }
    else {
        /* uphill move: accept maybe */
        accept with probability  $P = e^{-\Delta E/T}$ ;
        update configuration if accepted;
    }
} /* end for loop */
```

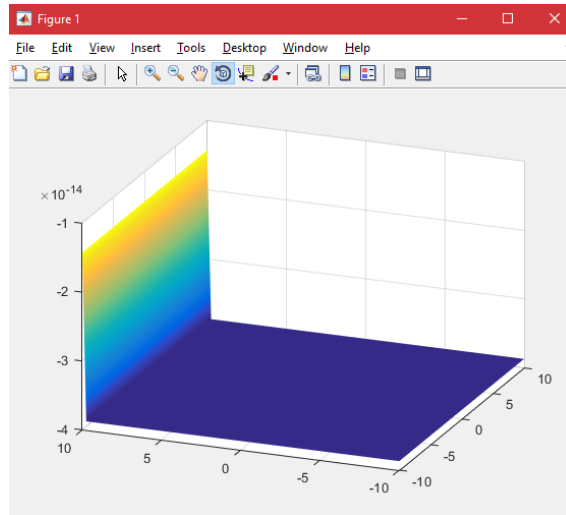
Fig 2 Metropolis algorithm.

b. Fungsi Matematis

Fungsi yang perlu diminimalkan adalah

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

Dengan X_1 dan X_2 memiliki range $[-10,10]$. Menggunakan meshgrid dan surf pada matlab dihasilkan visualisasi seperti dibawah ini.



c. Implementasi dan Hasil

Algoritma SA diimplementasikan ke dalam sebuah aplikasi menggunakan MATLAB R2015a dengan optimasi dalam Annealing Schedule. Dengan inputan random dari range $[-10,10]$ dengan hanya satu variable antara X_1 atau X_2 yang berubah dalam setiap iterasi. Dengan kondisi iterasi luar berhenti saat $T = 1$. Iterasi dalam ditujukan untuk mencari kondisi pengurangan temperature, saat masuk kondisi pertama (acceptable condition) dan kondisi kedua (acceptable with probability/failed) dengan batas max accept dan max failed yang ditentukan. Metode ini dapat menghasilkan nilai fungsi minimal $f(x,y) = -1.7312$.

Percobaan yang penulis lakukan adalah dengan mengubah nilai maxAcceptTemp, maxFailedTemp dengan tetap mempertahankan startTemperature dan coolingRate.

No	Start T	Cooling Rate	Max Accept Temp	Max Failed Temp	Uji Coba ke-	Minimal Value	Position[x,y]
1	5000	0.999	15	15	1	-1.7328	[1.2494 0.0035]
2					2	-1.7326	[1.2532 -0.0154]
3					3	-1.7329	[1.2675 0.0068]
4	5000	0.999	20	15	1	-1.7322	[1.2488 0.0234]
5					2	-1.7328	[1.2605 -0.0112]
6					3	-1.7327	[1.2470 0.0040]

7	5000	0.999	15	20	1	-1.7326	[1.2596 -0.0190]
8					2	-1.7317	[1.2928 -0.0201]
9					3	-1.7329	[1.2644 -0.0064]
7	5000	0.999	1	1	1	-1.7231	[1.1948 0.0702]
8					2	-1.7312	[1.3011 -0.0185]
9					3	-1.7144	[1.2583 0.1308]

Berdasarkan table diatas dapat disimpulkan bahwa titik global optimum yang didapat ada banyak sekali pada model ini. Adapun pengaruh dari Max Accept Temp dan Max Failed Temp adalah nilai minimum dan posisi yang dihasilkan, ketika diset ke 1 dalam artian tidak ada iterasi dalam maka nilai yang dihasilkan tidak optimal dan berubah-ubah tiap pengujiannya namun waktu iterasi luar selesai juga cepat sehingga untuk T=1 cepat dicapai. Berbeda ketika diset ke nilai 15-15, 20-15, atau 15-20, terbukti nilai yang dihasilkan adalah nilai optimum dalam hal ini -1.7312 namun untuk mencapai T=1 membutuhkan waktu yang sedikit lama.

C. Referensi

Rutenbar, R. A. (1989). Simulated Annealing Algorithms: An Overview. *IEEE Circuits and Devices Magazine*, 19-26.

D. Lampiran

```

1 % Initialization
2 newSAModel.position = model.start.position; %define start position
3 newSAModel.minValue = model.req.function(newSAModel.position(1),newSAModel.position(2)); %calculate the result
4
5 currentMinValue = newSAModel.minValue;
6 currentPosition = newSAModel.position;
7 T = model.start.temperature;
8
9 while (T > model.set.stopTemp)
10     acceptEnergy = 0;
11     failedEnergy = 0;
12
13     % T scheduling
14     while (acceptEnergy < model.set.maxAcceptTemp) || (failedEnergy < model.set.maxFailedTemp)
15         newPosition = randomize(currentPosition, model);
16         newMinValue = model.req.function(newPosition(1),newPosition(2));
17
18         if (newMinValue < currentMinValue)
19             % newMinValue is better, so replace the old one by the new
20             currentMinValue = newMinValue;
21             currentPosition = newPosition;
22         end
23     end
24     T = T * model.set.coolingRate;
25 end
26
27 resultSA = struct('position', currentPosition, 'minValue', currentMinValue);
28
29 % Display the result
30 display('Final position: ', currentPosition);
31 display('Final minimum value: ', currentMinValue);

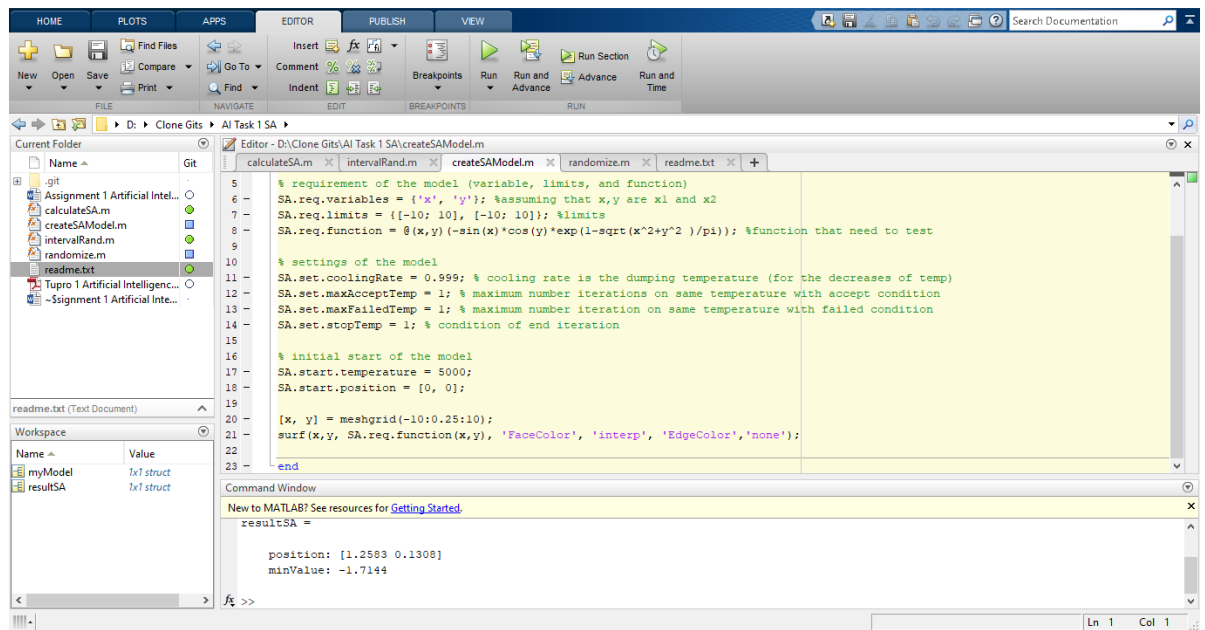
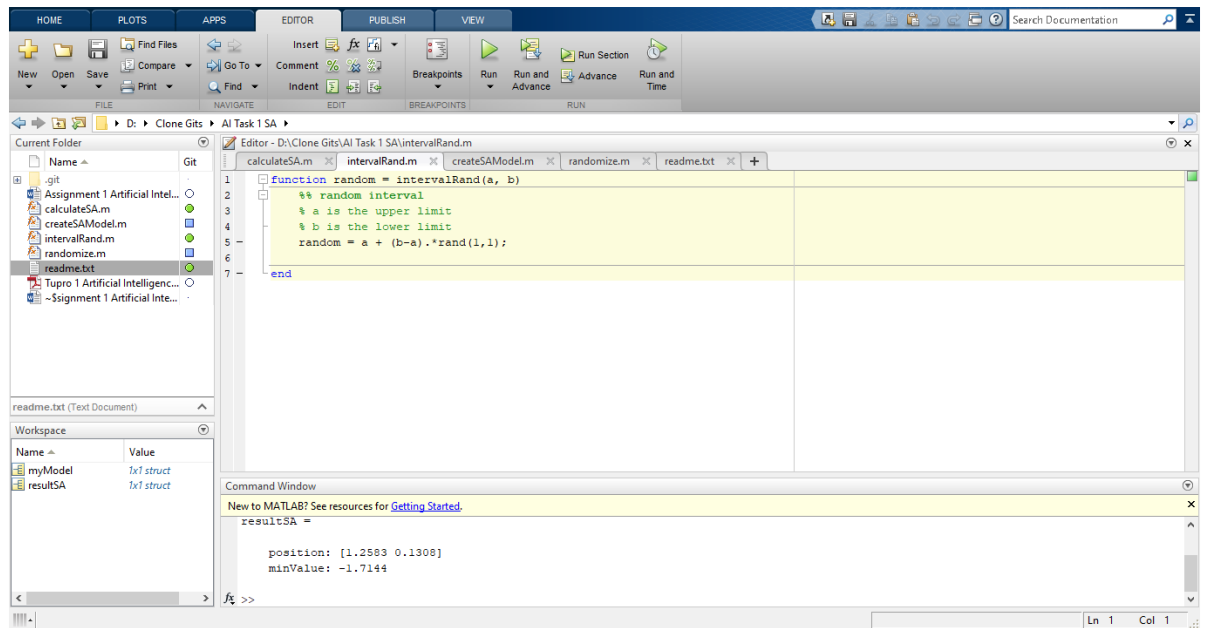
```

Command Window

```

New to MATLAB? See resources for Getting Started.
resultSA =
    position: [1.2583 0.1308]
    minValue: -1.7144

```



Thirafi Wian Anugrah
1301164069

