



TUGAS AKHIR FISIKA KOMPUTASI

JUDUL:

FINDING THE MAXIMUM PROBABILITY OF MACROSTATE (NUMBER OF OCCUPYING) FROM MOLECULAR DISTRIBUTION IN A MAXWELL-BOLTZMAN STATISTIC BY GENETIC ALGORITHM (GA)

Nama Penulis:

Mohammad Sulthonul Adhim (1114100068)

JURUSAN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2017

BAB I

PENDAHULUAN

1.1 Optimasi

Optimasi adalah suatu permasalahan komputasional yang bertujuan untuk menemukan solusi terbaik dari beberapa solusi yang mungkin. Secara lebih formal, optimasi adalah menemukan solusi yang berada dalam daerah yang mungkin (*feasible region*) yang memiliki nilai minimum atau maksimum dari fungsi objektif [1].

Optimasi didefinisikan sebagai proses pemilihan sebuah solusi dari sejumlah alternatif solusi dengan memenuhi sejumlah batasan (*constraints*). Misalkan pada permasalahan pencarian rute untuk mengunjungi sejumlah kota. Pada kasus ini tentu saja terdapat banyak alternatif pilihan rute yang dapat dijadikan sebuah solusi.

1.2 Algoritma Genetika

Algoritma Genetika atau *Genetic Algorithm* (GA) dikenalkan oleh John Holland dalam menyelesaikan masalah optimasi. Algoritma Genetika mensimulasikan proses yang terjadi pada populasi alamiah yang merupakan hal yang penting dalam proses evolusi. Algoritma Genetika adalah metode pencarian yang meniru perumpamaan evolusi biologis alami untuk menentukan kromosom atau individu berkualitas tinggi dalam suatu kawasan berhingga potensial yang disebut populasi. Proses pemilihan individu dari suatu populasi dievaluasi berdasarkan fungsi *fitness*. Kromosom berwujud string tersebut merupakan calon pada setiap siklus operasi yang disebut generasi.

Struktur umum pada Algoritma Genetika yaitu :

1. Representasi kromosom.
2. Evaluasi dengan menghitung fitness.
3. Proses *crossover* untuk mendapatkan individu baru.
4. Proses mutasi yang untuk meningkatkan variasi dalam populasi.
5. Proses seleksi untuk membentuk populasi baru.

1.3 Statistika Maxwell Boltzman

Persamaan distribusi partikel dalam statistika Maxwell Boltzman adalah sebagai berikut:

$$W_{MB} = \prod_{i=1}^{\Sigma E_{level}} \frac{g_i^{n_i}}{n_i!}$$

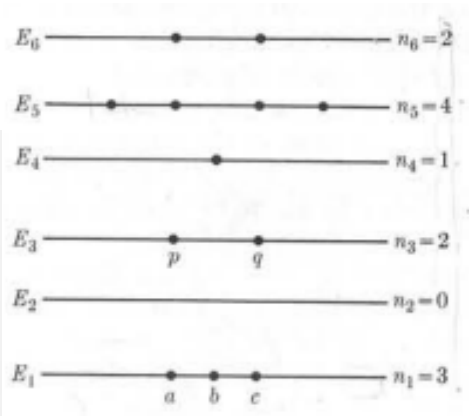
dengan:

- W_{MB} = Bilangan penempatan *microstate* dalam suatu *macrostate*
- g_i = Degenerasi (kamar) pada *Energy Level* ke-i
- n_i = Banyaknya partikel pada *Energy Level* ke-i

Persamaan ini merupakan fungsi tujuan yang akan dicari probabilitas maksimalnya dengan menggunakan algoritma genetika dengan constrain berupa N (jumlah partikel) dan E (jumlah energi).

Berikut adalah contoh *microstate* dan *macrostate*:

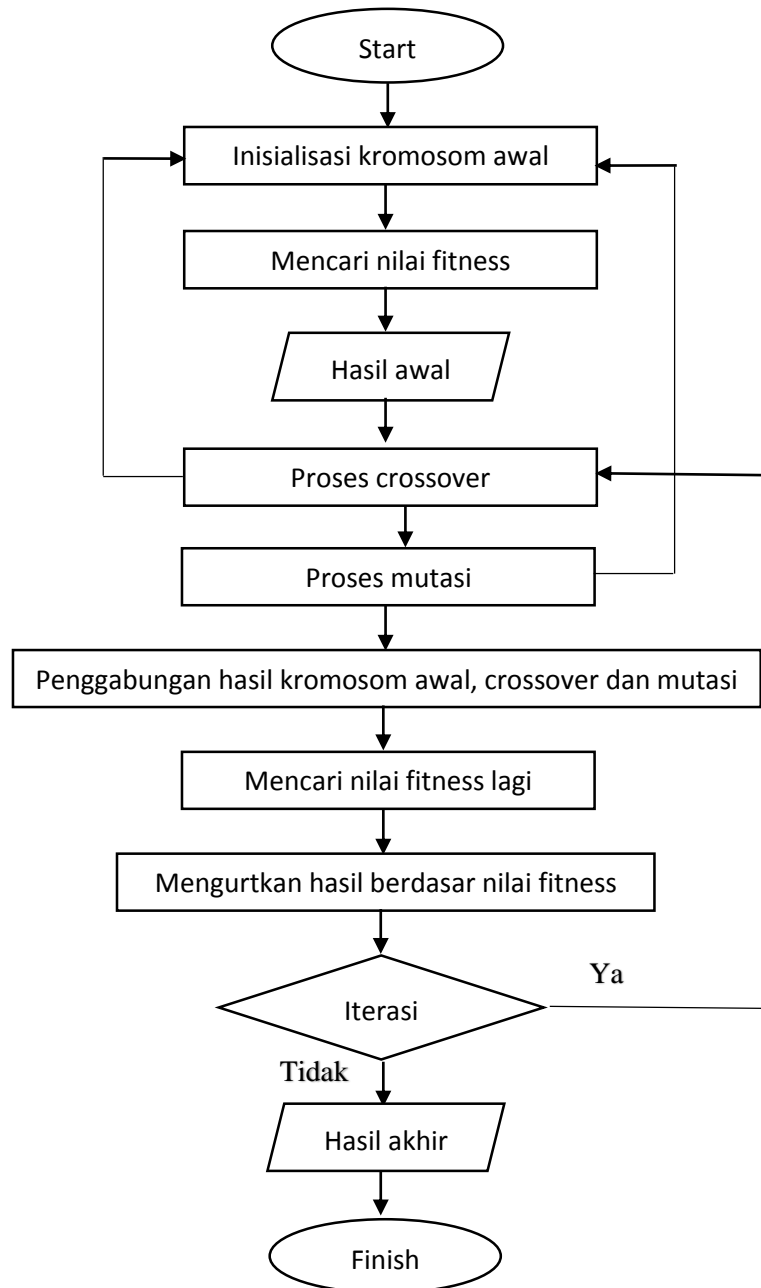
Energy Level ϵ_j	Macrostate A	Macrostate B
2	1	0
1	0	2
0	1	0



Gambar 1. *Microstate* dan *Macrostate* [2]

BAB II PEMROGRAMAN ALGORITMA GENETIKA

2.1 Diagram Pencarian dengan Algoritma Genetika



Gambar 2. *Flowchart* Pemrograman Algoritma Genetika

2.2 Script Matlab

```
clear; clc;

N = 4; % constrain jumlah partikel
El = 3; % banyaknya energy level
E = 5; % constrain total jumlah energi dari energy level yang ada
gi = 1; % degenerasi
n = 5; % banyaknya kromosom
Iteration = 100; % banyaknya iterasi

% ===== Inisialisasi Kromosom Awal ===== %
x = rand(n, El)
x = [x sum(x')']
for i = 1:n
    for j = 1:El
        x(i,j) = x(i,j)/x(i,El+1);
    end
end
x = x(:,1:El)
x = x*N
x = round(x)
x = [x(:,1:El-1) N-sum(x(:,1:El-1))']

for i = 1:El
    int(i) = i-1;
end
int
Ex = x*int';

W = ones(n,1);
for i = 1:n
    for j = 1:El
        W(i) = W(i)*((gi^x(i,j))/factorial(x(i,j)));
    end
end

Ex
W
f = W - (10^50)*abs(Ex-E) - (10^50)*abs(sum(x')'-N)

for iteration = 1 : Iteration
    % ===== Proses Crossover =====
    % x1 = zeros(n,El);
    x1 = x; % mengambil dari kromosom awal
    for i = 1:n
        x1(i,:) = round((x(randi(n),:)+x(randi(n),:))/2);
    % x1(i,1:round(El/2)) = x(randi(n),1:round(El/2));
    % x1(i,round(El/2)+1 : El) = x(randi(n),round(El/2)+1 : El);
    end

    % ===== Proses Mutasi =====
    x2 = x;
    for i = 1:n
        x2(i,randi(El)) = randi(N+1)-1;
    end
end
```

```

% ===== Proses Penggabungan Hasil & Diambil yang Terbaik =====
x3 = [x; x1;x2];
x3 = unique(x3,'rows');
for i = size(x3,1)+1 : 3*n
    for j = 1:E1
        x3(i,j) = randi(N+1)-1;
    end
end
Ex3 = x3*int';

W3 = ones(3*n,1);
for i = 1:3*n
    for j = 1:E1
        W3(i) = W3(i)*((gi^x3(i,j))/factorial(x3(i,j)));
    end
end

f3 = W3 - (10^50)*abs(Ex3-E) - (10^50)*abs(sum(x3')'-N);

% ===== Mengurutkan Kromosom Berdasarkan Nilai Fitnessnya =====
x3f3 = [x3 f3];
sortx3f3 = sortrows(x3f3,E1+1);
for i = 1 : 3*n
    sortx3f3up(i,:) = sortx3f3(3*n+1-i,:);
end
x = sortx3f3up(1:n,1:E1);
f = sortx3f3up(1:n,E1+1);
end

x1
% x2
x3
Ex3
W3
f3
% x3f3
% sortx3f3
% sortx3f3up
x
num2str(f)
x(1,:)
num2str(f(1))

```

BAB III

HASIL PEMROGRAMAN ALGORTIMA GENETIK

3.1 Hasil *Running*

a. Kromosom Awal

x =

0.5464	0.0205	0.9211
0.3989	0.9237	0.7947
0.4151	0.6537	0.5774
0.1807	0.9326	0.4400
0.2554	0.1635	0.2576

x =

0.5464	0.0205	0.9211	1.4881
0.3989	0.9237	0.7947	2.1172
0.4151	0.6537	0.5774	1.6462
0.1807	0.9326	0.4400	1.5534
0.2554	0.1635	0.2576	0.6765

x =

0.3672	0.0138	0.6190
0.1884	0.4363	0.3753
0.2522	0.3971	0.3507
0.1164	0.6004	0.2833
0.3775	0.2417	0.3808

x =

1.4689	0.0552	2.4759
0.7536	1.7451	1.5013
1.0086	1.5884	1.4030
0.4654	2.4015	1.1331
1.5100	0.9668	1.5232

x =

1	0	2
1	2	2
1	2	1
0	2	1
2	1	2

x =

1	0	3
1	2	1
1	2	1
0	2	2
2	1	1

b. Nilai Fitness Awal

f =

0,5000000000000000
0,1666666666666667
-1,000000000000000e+50
-1,000000000000000e+50
-1,000000000000000e+50

c. Hasil Proses Crossover

x1 =

2	1	2
1	3	1
2	1	2
2	1	2
2	2	2

d. Hasil Proses Mutasi

x2 =

3	1	2
0	3	1
2	3	2
3	0	2
1	3	1

e. Penggabungan Semua Hasil

x3 =

0	3	1
1	1	2
1	2	2
1	3	1
2	0	2
2	1	2
2	2	2
2	3	2

3	0	2
3	1	2
4	2	3
3	0	1
1	4	1
1	3	3
3	4	3

f. Nilai Fitness Akhir

f3 =

0,166666666666667
0,500000000000000
-2,00000000000000e+50
-1,00000000000000e+50
-1,00000000000000e+50
-1,00000000000000e+50
-3,00000000000000e+50
-5,00000000000000e+50
-2,00000000000000e+50
-2,00000000000000e+50
-8,00000000000000e+50
-3,00000000000000e+50
-3,00000000000000e+50
-7,00000000000000e+50
-1,10000000000000e+51

g. Hasil Akhir Pencarian

x =

1	1	2
0	3	1
2	1	2
2	0	2
1	3	1

ans =

0.5
0.166666666666667
-1e+50
-1e+50
-1e+50

ans =

1	1	2
---	---	---

ans =

0.5

DAFTAR PUSTAKA

- [1] Black, E, Paul. 2001. *Dictionary of Algorithms, Data Structures, and Problems*.
- [2] Alonso, Marcelo & Edward J. Finn. 1968. *Quantum and Statistical Physics*. Addison-Wesley, USA.