

**KECERDASAN KOMPUTASIONAL A
EVOLUTIONARY PROGRAMMING**



Oleh :

I KOMANG ARYA GANDA WIGUNA

15/388479/PPA/04918

**PROGRAM STUDI S2 ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2016**

A. Pendahuluan

Evolutionary Programming (EP) merupakan salah satu teknik atau metode untuk mengatasi masalah optimasi. Walaupun pada awal diperkenalkannya metode ini ditujukan untuk menghasilkan suatu bentuk kecerdasan (intelligence) yang dipandang sebagai suatu tingkah laku yang adaptif (adaptif behaviour). Akan tetapi dalam perkembangannya EP justru mengalami pergeseran menuju bentuk yang mirip ES (Evolutionary Strategies), digunakan untuk mengatasi masalah-masalah optimasi numerik. EP bagian dari Evolutionary Algorithms (EA) dimana teknik ini meniru proses evolusi biologia. EP mempunyai tujuan seperti Genetic Programming (GP) untuk menghasilkan rangkaian program komputer tapi prinsip kerjanya seperti ES.

Salah satu permasalahan sederhana yang bisa diselesaikan dengan EP adalah fungsi non linear dimana sebuah fungsi disusun dengan variabel bebas x_1 dan x_2 yang menghasilkan variabel tak bebas y .

X_1	X_2	Y
3	4	18
4	2	24
6	3	51
6	4	48
7	1	87
8	2	104
9	3	123
10	4	144
10	6	128
11	9	143

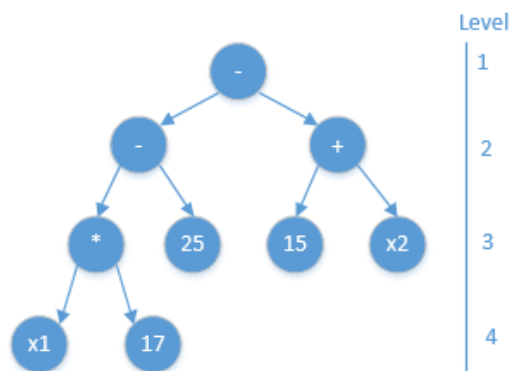
Misalkan nilai y pada tabel diatas dihasilkan menggunakan sebuah fungsi persamaan non-linear sebagai berikut :

$$y = f(x_1, x_2) = ((x_1 * 17) - 25) - (15 + x_2)$$

Dari permasalahan diatas akan diselesaikan dengan menggunakan algoritma EP untuk menyelesaikan permasalahan fungsi non linear diatas sesuai dengan constraint yang terdapat pada kasus, dalam hal ini nilai x_1 , x_2 dan y .

B. Representasi

Persamaan diatas bisa direpresentasikan dalam bentuk *binary tree*. *Tree* disusun dari sejumlah node dan link yang menunjuk sebuah node lainnya. Setiap node memiliki nilai dan 2 buah pointer yang menunjukan pada node selanjutnya. Level atau kedalaman dari tree yang digunakan dalam kasus ini adalah 4. Minimal kedalaman dari tree adalah 3 sehingga dapat memuat setidaknya 1 terminal, x_1 atau x_2 . Terminal adalah variabel x_1 , x_2 atau nilai x (random) dimana x elemen bilangan real dengan nilai maksimal 25. Operator atau fungsi yang digunakan untuk kasus yaitu $\{+, -, /\}$. Terminal hanya mempunyai nilai pada node leaf tree (daun tree), selain itu nilai dari node menggunakan operator atau fungsi.



C. Inisialisasi

Popsize (μ) yang digunakan dalam kasus ini adalah 10 sehingga akan dibangkitkan 10 tree individu atau kromosom secara acak dan posisi tree yang berbeda-beda. Nilai lamda sebesar 1 dan generasi sampai 200 generasi. Nilai generasi ini akan digunakan untuk kondisi berhenti pada EP. Semakin tinggi ukuran dan kompleksitas masalah maka nilai generasi akan semakin besar. Nilai berhenti generasi ini ditentukan sedemikian rupa sehingga konvergensi populasi tercapai. Namun dalam kasus ini cukup menggunakan 200 iterasi/generasi.

D. Reproduksi

Karena prinsip yang sama dengan ES, maka pada EP untuk melakukan reproduksi lebih menekankan pada mutasi untuk menghasilkan *offspring*. Nilai lamda yang digunakan pada kasus ini adalah 1 maka tiap parent akan menghasilkan 1 *offspring* ($1 * \mu = 1 * 10 = 10$).

Setiap parent akan dilakukan proses mutasi dengan menelusuri setiap node pada tree dan menggunakan posisi random untuk menentukan posisi mutasi, misalkan node yang terpilih pada node A, maka node akan dilakukan proses mutasi. Jika pada node A tersebut adalah sebuah operator atau fungsi maka akan dirandom pada operator saja yaitu $\{+, -, /\}$. Dan sebaliknya jika node A adalah sebuah terminal maka akan ditentukan nilai pengganti apakah variabel x_1 , x_2 atau nilai random dari 1 sampai 25.

Parent	$((x_1 * 17) - 22) - (2 + 14)$
Offspring	$((x_1 * 17) - 22) - (x_2 + x_1)$

E. Evaluasi

Dalam tahap evaluasi setiap individu dihitung nilai fitness. Dalam kasus fungsi non linear ini, semakin kecil nilai fitness akan semakin bagus karena jumlah selisih nilai y dengan hasil dari fungsi non linear dikatakan dekat. Nilai fitness didapatkan dengan menyusun fungsi non linear dari binary tree yang dihasilkan. Penyusunan dari binary tree menggunakan cara *inorder* dan perhitungan fitness memanfaatkan *stack* dan hasil dari *postorder*. Selanjutnya selisih antara output fungsi non linear yang dihasilkan dengan nilai y pada kasus.

x_1	x_2	y	Individu 1	
			$y_{p1} = ((x_1 * 17) - 22) - (15 + x_2)$	
			y_{p1}	$ y - y_{p1} $
3	4	18	10	8
4	2	24	29	5
6	3	51	62	11
6	4	48	61	13
7	1	87	81	6
8	2	104	97	7
9	3	123	113	10
10	4	144	129	15
10	6	128	127	1
11	9	143	141	2
Fitness individu 1				78

F. Seleksi

Metode seleksi yang digunakan adalah *elitism*. *Elitism* melakukan proses seleksi dengan mengumpulkan semua individu dalam populasi *parent* dan *offspring* menjadi satu populasi. Individu terbaik dalam populasi akan lolos untuk masuk dalam generasi selanjutnya. Metode seleksi ini menjamin individu yang terbaik yang selalu lolos.

```
=====
NILAI FITNES TERBAIK : 72.0
Generasi : 200
In order : 18*x1+x1-x2-x1+23+23
Post order : 18x1*x1x2-+x12323+-
=====
```

Hasil run program menggunakan JAVA yang ditampilkan pada console

REFERENCES

Koza, John R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press.

Introduction to Evolutionary Algorithms. Yu, X.; Gen, M. 2010, XVI, 422 p. 168 illus., Hardcover. ISBN : 978-1-84996-128-8.