

Algoritma Genetika dan Contoh Aplikasinya

Denny Hermawanto

d_3_nny@yahoo.com

http://dennyhermawanto.webhop.org

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih “alamiah”. Salah satu aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi. Dalam tulisan ini akan dibahas teori dasar algoritma genetika beserta contoh aplikasinya dalam menyelesaikan suatu permasalahan optimasi kombinasi sederhana.

Teori Dasar Algoritma Genetika

Algoritma genetika yang dikembangkan oleh Goldberg adalah algoritma komputasi yang diinspirasi teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi aturan “yang kuat adalah yang menang”. Darwin juga menyatakan bahwa kelangsungan hidup suatu makhluk dapat dipertahankan melalui proses reproduksi, crossover, dan mutasi. Konsep dalam teori evolusi Darwin tersebut kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih “alamiah”.

Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai **chromosome**, sedangkan kumpulan chromosome-chromosome tersebut disebut sebagai **populasi**. Sebuah chromosome dibentuk dari komponen-komponen penyusun yang disebut sebagai **gen** dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Chromosome-chromosome tersebut akan berevolusi secara berkelanjutan yang disebut dengan **generasi**. Dalam tiap generasi chromosome-chromosome tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (**fungsi_objektif**) menggunakan ukuran yang disebut dengan **fitness**. Untuk memilih chromosome yang tetap dipertahankan untuk generasi selanjutnya

Komunitas eLearning IlmuKomputer.Com

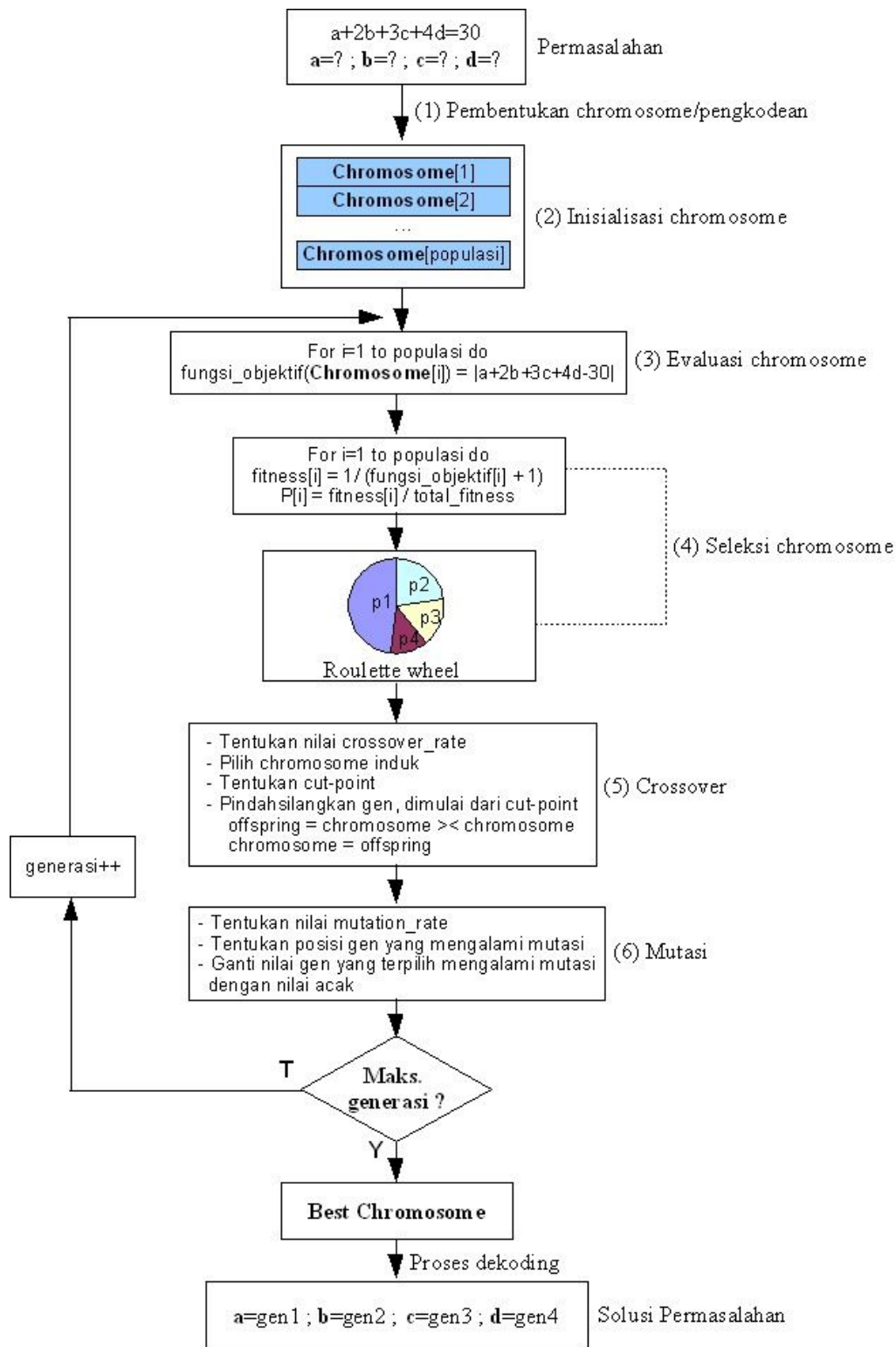
Copyright © 2003-2007 IlmuKomputer.Com

dilakukan proses yang disebut dengan *seleksi*. Proses seleksi chromosome menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu chromosome yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya.

Chromosome-chromosome baru yang disebut dengan *offspring*, dibentuk dengan cara melakukan perkawinan antar chromosome-chromosome dalam satu generasi yang disebut sebagai proses *crossover*. Jumlah chromosome dalam populasi yang mengalami crossover ditentukan oleh parameter yang disebut dengan *crossover_rate*. Mekanisme perubahan susunan unsur penyusun makhluk hidup akibat adanya faktor alam yang disebut dengan *mutasi* direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam chromosome dengan suatu nilai acak. Jumlah gen dalam populasi yang mengalami mutasi ditentukan oleh parameter yang dinamakan *mutation_rate*. Setelah beberapa generasi akan dihasilkan chromosome-chromosome yang nilai gen-gennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh algoritma genetika terhadap permasalahan yang ingin diselesaikan.

Aplikasi Algoritma Genetika

Berikut adalah contoh aplikasi algoritma genetika yang digunakan untuk menyelesaikan masalah kombinasi. Misalkan ada persamaan $a+2b+3c+4d = 30$, kita mencari nilai a, b, c, dan d yang memenuhi persamaan diatas. Kita mencoba menggunakan algoritma genetika untuk menyelesaikan permasalahan diatas. Diagram alir dari algoritma genetika untuk menyelesaikan permasalahan diatas dapat dilihat pada Gambar 1.



Gambar 1. Diagram alir algoritma genetika

Penjelasan mengenai langkah-langkah penyelesaian permasalahan diatas menggunakan algoritma genetika adalah sebagai berikut:

1. Pembentukan chromosome

Karena yang dicari adalah nilai a, b, c, d maka variabel a, b, c, d dijadikan sebagai gen-gen pembentuk chromosome. Batasan nilai variabel a adalah bilangan integer 0 sampai 30. Sedangkan batasan nilai variabel b, c, dan d adalah bilangan integer 0 sampai 10.

2. Inisialisasi

Proses inisialisasi dilakukan dengan cara memberikan nilai awal gen-gen dengan nilai acak sesuai batasan yang telah ditentukan.

Misalkan kita tentukan jumlah populasi adalah 6, maka:

Chromosome[1] = [a;b;c;d] = [12;05;03;08]

Chromosome[2] = [a;b;c;d] = [02;01;08;03]

Chromosome[3] = [a;b;c;d] = [10;04;03;04]

Chromosome[4] = [a;b;c;d] = [20;01;10;06]

Chromosome[5] = [a;b;c;d] = [01;04;03;09]

Chromosome[6] = [a;b;c;d] = [20;05;07;01]

3. Evaluasi Chromosome

Permasalahan yang ingin diselesaikan adalah nilai variabel a, b, c, dan d yang memenuhi persamaan $a+2b+3c+4d = 30$, maka fungsi_objektif yang dapat digunakan untuk mendapatkan solusi adalah **fungsi_objektif(chromosome) = | (a+2b+3c+4d) - 30 |**

Kita hitung fungsi_objektif dari chromosome yang telah dibangkitkan:

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[1]) &= \text{Abs}((12 + 2*5 + 3*3 + 4*8) - 30) \\ &= \text{Abs}((12 + 10 + 9 + 32) - 30) \\ &= \text{Abs}(63 - 30) \\ &= 33\end{aligned}$$

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[2]) &= \text{Abs}((2 + 2*1 + 3*8 + 4*3) - 30) \\ &= \text{Abs}((2 + 2 + 24 + 12) - 30) \\ &= \text{Abs}(40 - 30) \\ &= 10\end{aligned}$$

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[3]) &= \text{Abs}((10 + 2*4 + 3*3 + 4*4) - 30) \\ &= \text{Abs}((10 + 8 + 9 + 16) - 30) \\ &= \text{Abs}(43 - 30) \\ &= 13\end{aligned}$$

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[4]) &= \text{Abs}((20 + 2*1 + 3*10 + 4*6) - 30) \\ &= \text{Abs}((20 + 2 + 30 + 24) - 30) \\ &= \text{Abs}(76 - 30) \\ &= 46\end{aligned}$$

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[5]) &= \text{Abs}((1 + 2*4 + 3*3 + 4*9) - 30) \\ &= \text{Abs}((1 + 8 + 9 + 36) - 30) \\ &= \text{Abs}(54 - 30) \\ &= 24\end{aligned}$$

$$\begin{aligned}\text{fungsi_objektif}(\text{chromosome}[6]) &= \text{Abs}((20 + 2*5 + 3*7 + 4*1) - 30) \\ &= \text{Abs}((20 + 10 + 21 + 4) - 30) \\ &= \text{Abs}(55 - 30) \\ &= 25\end{aligned}$$

Rata-rata dari fungsi objektif adalah:

$$\begin{aligned}\text{rata-rata} &= (33+10+13+46+24+25)/6 \\ &= 151 / 6 \\ &= 25.167\end{aligned}$$

4. Seleksi Chromosome

Proses seleksi dilakukan dengan cara membuat chromosome yang mempunyai fungsi_objektif kecil mempunyai kemungkinan terpilih yang besar atau mempunyai nilai probabilitas yang tinggi. Untuk itu dapat digunakan fungsi **fitness** = $(1/(1+\text{fungsi_objektif}))$, fungsi_objektif perlu ditambah 1 untuk menghindari kesalahan program yang diakibatkan pembagian oleh 0.

$$\begin{aligned}\text{fitness}[1] &= 1 / (\text{fungsi_objektif}[1]+1) \\ &= 1 / 34 \\ &= 0.0294 \\ \text{fitness}[2] &= 1 / (\text{fungsi_objektif}[2]+1) \\ &= 1 / 11 \\ &= 0.0909 \\ \text{fitness}[3] &= 1 / (\text{fungsi_objektif}[3]+1) \\ &= 1 / 14 \\ &= 0.0714 \\ \text{fitness}[4] &= 1 / (\text{fungsi_objektif}[4]+1) \\ &= 1 / 47 \\ &= 0.0212 \\ \text{fitness}[5] &= 1 / (\text{fungsi_objektif}[5]+1) \\ &= 1 / 25 \\ &= 0.0400 \\ \text{fitness}[6] &= 1 / (\text{fungsi_objektif}[6]+1) \\ &= 1 / 26 \\ &= 0.0385 \\ \text{total_fitness} &= 0.0294 + 0.0909 + 0.0714 + 0.0212 + 0.04 + 0.0385 \\ &= 0.2914\end{aligned}$$

Rumus untuk mencari probabilitas: **P[i] = fitness[i] / total_fitness**

$$\begin{aligned}\text{P}[1] &= 0.0294 / 0.2914 \\ &= 0.1009 \\ \text{P}[2] &= 0.0909 / 0.2914 \\ &= 0.3119 \\ \text{P}[3] &= 0.0714 / 0.2914 \\ &= 0.2450 \\ \text{P}[4] &= 0.0212 / 0.2914 \\ &= 0.0728 \\ \text{P}[5] &= 0.04 / 0.2914 \\ &= 0.1373 \\ \text{P}[6] &= 0.0385 / 0.2914 \\ &= 0.1321\end{aligned}$$

Dari probabilitas diatas dapat kita lihat kalau **chromosome** ke 2 yang mempunyai fitness paling

besar maka chromosome tersebut mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari **chromosome** lainnya. Untuk proses seleksi kita gunakan *roulette wheel*, untuk itu kita harus mencari dahulu nilai kumulatif probabilitasnya:

$$C[1] = 0.1009$$

$$C[2] = 0.1009 + 0.3119 \\ = 0.4128$$

$$C[3] = 0.1009 + 0.3119 + 0.2450 \\ = 0.6578$$

$$C[4] = 0.1009 + 0.3119 + 0.2450 + 0.0728 \\ = 0.7306$$

$$C[5] = 0.1009 + 0.3119 + 0.2450 + 0.0728 + 0.1373 \\ = 0.8679$$

$$C[6] = 0.1009 + 0.3119 + 0.2450 + 0.0728 + 0.1373 + 0.1321 \\ = 1$$

Setelah dihitung cumulative probabilitasnya maka proses seleksi menggunakan roulette-wheel dapat dilakukan. Prosesnya adalah dengan membangkitkan bilangan acak **R** dalam range 0-1.

Jika $R[k] < C[1]$ maka pilih chromosome 1 sebagai induk, selain itu pilih chromosome ke-k sebagai induk dengan syarat $C[k-1] < R < C[k]$. Kita putar roulette wheel sebanyak jumlah populasi yaitu 6 kali (bangkitkan bilangan acak **R**) dan pada tiap putaran, kita pilih satu chromosome untuk populasi baru. Misal:

$$R[1] = 0.201$$

$$R[2] = 0.284$$

$$R[3] = 0.009$$

$$R[4] = 0.822$$

$$R[5] = 0.398$$

$$R[6] = 0.501$$

Angka acak pertama **R**[1] adalah lebih besar dari **C**[1] dan lebih kecil daripada **C**[2] maka pilih chromosome[2] sebagai chromosome pada populasi baru, dari bilangan acak yang telah dibangkitkan diatas maka populasi chromosome baru hasil proses seleksi adalah:

$$\text{chromosome}[1] = \text{chromosome}[2]$$

$$\text{chromosome}[2] = \text{chromosome}[2]$$

$$\text{chromosome}[3] = \text{chromosome}[1]$$

$$\text{chromosome}[4] = \text{chromosome}[5]$$

$$\text{chromosome}[5] = \text{chromosome}[2]$$

$$\text{chromosome}[6] = \text{chromosome}[3]$$

Chromosome baru hasil proses seleksi:

$$\text{chromosome}[1] = [02;01;08;03]$$

$$\text{chromosome}[2] = [02;01;08;03]$$

$$\text{chromosome}[3] = [12;05;03;08]$$

$$\text{chromosome}[4] = [01;04;03;09]$$

$$\text{chromosome}[5] = [02;01;08;03]$$

$$\text{chromosome}[6] = [10;04;03;04]$$

5. Crossover

Setelah proses seleksi maka proses selanjutnya adalah proses crossover. Metode yang digunakan salah satunya adalah one-cut point, yaitu memilih secara acak satu posisi dalam chromosome induk kemudian saling menukar gen. Chromosome yang dijadikan induk dipilih secara acak dan jumlah chromosome yang mengalami crossover dipengaruhi oleh parameter *crossover_rate* (**pc**).

Pseudo-code untuk proses crossover adalah sebagai berikut:

begin

```
k ← 0;
while(k < populasi) do
  R[k] ← random(0-1);
  if (R[k] < pc) then
    select Chromosome[k] as parent;
  end;
  k = k + 1;
end;
```

end;

Misal kita tentukan crossover probability adalah sebesar 25%, maka diharapkan dalam satu generasi ada 50% Chromosome (3 chromosome) dari satu generasi mengalami proses crossover.

Prosesnya adalah sebagai berikut:

Pertama kita bangkitkan bilangan acak **R** sebanyak jumlah populasi

R[1] = 0.191

R[2] = 0.259

R[3] = 0.760

R[4] = 0.006

R[5] = 0.159

R[6] = 0.340

Maka Chromosome ke k akan dipilih sebagai induk jika **R**[k] < **pc**, dari bilangan acak **R** diatas maka yang dijadikan induk adalah **chromosome**[1], **chromosome**[4] dan **chromosome**[5].

Setelah melakukan pemilihan induk proses selanjutnya adalah menentukan posisi crossover. Ini dilakukan dengan cara membangkitkan bilangan acak dengan batasan 1 sampai (panjang **chromosome**-1), dalam kasus ini bilangan acak yang dibangkitkan adalah 1 – 3. Misalkan didapatkan posisi crossover adalah 1 maka **chromosome** induk akan dipotong mulai gen ke 1 kemudian potongan **gen** tersebut saling ditukarkan antar induk.

chromosome[1] >< **chromosome**[4]

chromosome[4] >< **chromosome**[5]

chromosome[5] >< **chromosome**[1]

Posisi *cut-point* crossover dipilih menggunakan bilangan acak 1-3 sebanyak jumlah crossover yang terjadi, misal

C[1] = 1

C[2] = 1

C[3] = 2

```
offspring[1] = chromosome[1] >< chromosome[4]
              = [02;01;08;03] >< [01;04;03;09]
              = [02;04;03;09]
offspring[4] = Chromosome[4] >< Chromosome[5]
              = [01;04;03;09] >< [02;01;08;03]
              = [01;01;08;03]
offspring[5] = Chromosome[5] >< Chromosome[1]
              = [02;01;08;03] >< [02;01;08;03]
              = [02;01;08;03]
```

Dengan demikian populasi Chromosome setelah mengalami proses crossover menjadi:

```
chromosome[1] = [02;04;03;09]
chromosome[2] = [02;01;08;03]
chromosome[3] = [12;05;03;08]
chromosome[4] = [01;01;08;03]
chromosome[5] = [02;01;08;03]
chromosome[6] = [10;04;03;04]
```

6. Mutasi

Jumlah chromosome yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation_rate*. Proses mutasi dilakukan dengan cara mengganti satu gen yang terpilih secara acak dengan suatu nilai baru yang didapat secara acak. Prosesnya adalah sebagai berikut. Pertama kita hitung dahulu panjang total **gen** yang ada dalam satu populasi. Dalam kasus ini panjang total gen adalah **total_gen** = (jumlah **gen** dalam chromosome) * jumlah populasi

$$= 4 * 6$$
$$= 24$$

Untuk memilih posisi **gen** yang mengalami mutasi dilakukan dengan cara membangkitkan bilangan integer acak antara 1 sampai **total_gen**, yaitu 1 sampai 24. Jika bilangan acak yang kita bangkitkan lebih kecil daripada variabel *mutation_rate* (**pm**) maka pilih posisi tersebut sebagai sub-chromosome yang mengalami mutasi. Misal **pm** kita tentukan 10% maka diharapkan ada 10% dari **total_gen** yang mengalami mutasi:

```
jumlah mutasi = 0.1 * 24
              = 2.4
              = 2
```

Misalkan setelah kita bangkitkan bilangan acak terpilih posisi **gen** 12 dan 18 yang mengalami mutasi. Dengan demikian yang akan mengalami mutasi adalah **chromosome** ke-3 gen nomor 4 dan **Chromosome** ke-5 **gen** nomor 2. Maka nilai **gen** pada posisi tersebut kita ganti dengan bilangan acak 0-30.

Misalkan bilangan acak yang terbangkitkan adalah 2 dan 5. Maka populasi chromosome setelah mengalami proses mutasi adalah:

```
chromosome[1] = [02;04;03;09]
chromosome[2] = [02;01;08;03]
chromosome[3] = [12;05;03;02]
chromosome[4] = [01;01;08;03]
chromosome[5] = [02;05;08;03]
chromosome[6] = [10;04;03;04]
```


Setelah proses mutasi maka kita telah menyelesaikan satu iterasi dalam algoritma genetika atau disebut dengan satu generasi. Maka fungsi_objective setelah satu generasi adalah:

$$\begin{aligned}\text{chromosome}[1] &= [02;04;03;09] \\ \text{fungsi_objektif}[1] &= \text{Abs}((2 + 2*4 + 3*3 + 4*9) - 30) \\ &= \text{Abs}((2 + 8 + 9 + 36) - 30) \\ &= \text{Abs}(55 - 30) \\ &= 25\end{aligned}$$

$$\begin{aligned}\text{chromosome}[2] &= [02;01;08;03] \\ \text{fungsi_objektif}[2] &= \text{Abs}((2 + 2*1 + 3*8 + 4*3) - 30) \\ &= \text{Abs}((2 + 2 + 24 + 12) - 30) \\ &= \text{Abs}(40 - 30) \\ &= 10\end{aligned}$$

$$\begin{aligned}\text{chromosome}[3] &= [12;05;03;02] \\ \text{fungsi_objektif}[3] &= \text{Abs}((12 + 2*5 + 3*3 + 4*2) - 30) \\ &= \text{Abs}((12 + 10 + 9 + 8) - 30) \\ &= \text{Abs}(39 - 30) \\ &= 9\end{aligned}$$

$$\begin{aligned}\text{chromosome}[4] &= [01;01;08;03] \\ \text{fungsi_objektif}[4] &= \text{Abs}((1 + 2*1 + 3*8 + 4*3) - 30) \\ &= \text{Abs}((1 + 2 + 24 + 12) - 30) \\ &= \text{Abs}(39 - 30) \\ &= 9\end{aligned}$$

$$\begin{aligned}\text{chromosome}[5] &= [02;05;08;03] \\ \text{fungsi_objektif}[5] &= \text{Abs}((2 + 2*5 + 3*8 + 4*3) - 30) \\ &= \text{Abs}((2 + 10 + 24 + 12) - 30) \\ &= \text{Abs}(48 - 30) \\ &= 18\end{aligned}$$

$$\begin{aligned}\text{chromosome}[6] &= [10;04;03;04] \\ \text{fungsi_objektif}[6] &= \text{Abs}((10 + 2*4 + 3*3 + 4*4) - 30) \\ &= \text{Abs}((10 + 8 + 9 + 16) - 30) \\ &= \text{Abs}(43 - 30) \\ &= 13\end{aligned}$$

Rata-rata fungsi objektif setelah satu generasi adalah:

$$\begin{aligned}\text{rata-rata} &= (25 + 10 + 9 + 9 + 18 + 13) / 6 \\ &= 84 / 6 \\ &= 14.0\end{aligned}$$

Dapat dilihat dari hasil perhitungan fungsi objektif diatas bahwa setelah satu generasi, nilai hasil rata-rata fungsi_objektif lebih menurun dibandingkan hasil fungsi_objektif pada saat sebelum mengalami seleksi, crossover dan mutasi. Hal ini menunjukkan bahwa chromosome atau solusi

yang dihasilkan setelah satu generasi lebih baik dibandingkan generasi sebelumnya. Maka pada generasi selanjutnya chromosome-chromosome yang baru adalah:

chromosome[1] = [02;04;03;09]
chromosome[2] = [02;01;08;03]
chromosome[3] = [12;05;03;02]
chromosome[4] = [01;01;08;03]
chromosome[5] = [02;05;08;03]
chromosome[6] = [10;04;03;04]

Chromosome-chromosome ini akan mengalami proses yang sama seperti generasi sebelumnya yaitu proses evaluasi, seleksi, crossover dan mutasi yang kemudian akan menghasilkan chromosome-chromosome baru untuk generasi yang selanjutnya. Proses ini akan berulang sampai sejumlah generasi yang telah ditetapkan sebelumnya.

Setelah 50 generasi didapatkan chromosome yang terbaik adalah:

Chromosome = [07;05;03;01]

Jika didekode maka:

a=7 ; b=5 ; c=3 ; d=1

Jika dihitung terhadap persamaan $f = a+2b+3c+4d$:

$7 + (2*5) + (3*3) + (4*1) = 30$

Penutup

Pada contoh diatas telah diberikan contoh mengenai penerapan algoritma genetika untuk mencari solusi suatu persamaan matematis sederhana. Source code dalam bahasa Java untuk contoh diatas dapat didownload di milis **sc-ina** yahoogroups: <http://groups.yahoo.com/group/sc-ina/> atau <http://soft-computing.org>

Referensi

- [1] David E. Goldberg, "Genetic Algorithms in Search, Optimization & Machine Learning", Addison- Wesley, 1989.
- [2] Mitsuo Gen, Runwei Cheng, "*Genetic Algorithms And Engineering Design*", John Wiley & Sons, 1997.
- [3] Forrest, S., "Genetic Algorithms: principles of natural selection applied to computation", Science, vol.261, pp.872-878, 1993.
- [4] <http://www.generation5.org>

Biografi Penulis



Denny Hermawanto, lahir di Desa Brenggolo Kabupaten Kediri tahun 1980. Kuliah di Politeknik Elektronika Negeri Surabaya (PENS-ITS) dan lulus tahun 2001. Melanjutkan S1 di Teknik Elektro ITS dan mengambil bidang peminatan Teknik Komputer. Saat ini bekerja di Lab. Akustik & Getaran, Puslit KIM-LIPI. Bidang yang diminati adalah mengenai pengolahan sinyal akustik & getaran, otomatisasi pengukuran dan *bio-inspired algorithm*. Penulis dapat dihubungi via YM: **d_3_nny**