

Representasi Chromosome Algoritma Genetika dalam Bentuk Biner

Denny Hermawanto

d_3_nny@yahoo.com

http://dennyhermawanto.webhop.org

Lisensi Dokumen:

Copyright © 2003-2006 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.

Pendahuluan

Sebuah nilai solusi yang dihasilkan dalam algoritma genetika disebut dengan *chromosome* [1]. *Chromosome* tersusun atas gen-gen yang merupakan representasi nilai variabel dalam suatu fungsi yang ingin dicari solusinya. *Chromosome* dapat dibentuk dari bilangan numerik, biner ataupun karakter. Representasi *chromosome* dalam bentuk biner adalah bentuk yang banyak digunakan karena mempermudah pemrograman didalam proses algoritma genetika seperti crossover dan mutasi. Untuk memudahkan dalam mengingat selanjutnya *chromosome* yang tersusun atas gen-gen dalam bentuk biner dapat kita sebut sebagai *chromosome* biner.

Sebelum diproses dalam algoritma genetika, solusi yang dibangkitkan baik dalam bentuk integer atau float harus dirubah dulu ke dalam bentuk biner. Demikian pula sebaliknya setelah proses-proses dalam algoritma genetika selesai maka nilai *chromosome* biner tersebut harus dirubah kembali ke bentuk semula yaitu berupa bilangan integer atau float.

Berikut adalah contoh proses membentuk *chromosome* biner dari suatu bilangan integer. Misalkan diketahui fungsi $f(x)=x^3+2x+1$, algoritma genetika digunakan untuk mendapatkan nilai x yang optimal dalam range solusi $9 \leq x \leq 30$. Maka *chromosome* biner x dapat dibentuk dengan bilangan biner yang merepresentasikan nilai integer antara 9 sampai 30. Langkah pertama adalah menentukan panjang bilangan biner yang dapat menjangkau nilai integer mulai dari 9 sampai 30 atau dapat juga ditulis $2^n > 30$ dimana n = panjang biner. Yang ingin kita tentukan adalah nilai n yang jika 2 dipangkatkan dengan n nilainya mendekati 30. Jika $n = 4$, maka $2^4=16$, nilai integer yang dapat direpresentasikan hanya mulai dari 0 sampai 15. Apabila $n = 6$ maka integer yang dapat direpresentasikan adalah dari 0 sampai 63, tetapi hal ini tidak efektif dalam penggunaan memori karena yang dibutuhkan hanya untuk mengkodekan desimal dari 0-30. Sehingga nilai n yang paling sesuai adalah 5 karena $2^5=32$, range nilai integer yang dapat direpresentasikannya adalah dari 0 sampai 31.

Setelah diketahui panjang biner proses selanjutnya adalah mengubah bilangan integer menjadi biner. Algoritma untuk mengubah bilangan integer ke biner sangat sederhana yaitu dengan mendapatkan nilai modulus (sisa hasil pembagian) dari pembagian bilangan dengan 2 sebanyak panjang biner (n kali).

Sebagai contoh adalah merubah bilangan integer 23 ke dalam bentuk biner dengan panjang biner $n = 5$:

Langkah-langkah:

1. Lakukan proses pembagian bilangan dengan 2 sebanyak 5 kali iterasi, dan dapatkan nilai modulusnya:

23 dibagi 2 sama dengan 11 sisa 1
11 dibagi 2 sama dengan 5 sisa 1
5 dibagi 2 sama dengan 2 sisa 1
2 dibagi 2 sama dengan 1 sisa 0
1 dibagi 2 sama dengan 0 sisa 1

2. Dapatkan nilai binernya dengan cara menyimpan nilai modulus dimulai dari bawah atau dari pembagian yang terakhir.

Maka biner untuk 23 adalah: 10111

Pengkodean Bilangan Floating

Prinsip untuk merubah bilangan float ke biner adalah dengan merubah bilangan float tersebut menjadi bilangan integer terlebih dahulu, setelah itu bilangan integer tersebut baru dirubah dalam biner. Untuk mengubah dari bilangan float ke integer adalah dengan mengalikan bilangan tersebut dengan 10^m dimana m = jumlah angka dibelakang koma dari bilangan float yang akan dirubah ke biner. Jika ruang solusi dari algoritma genetika adalah antara $a \leq x \leq b$ dimana x adalah bilangan float, maka [2]:

$$2^{n-1} < (b-a) * 10^m \leq 2^n - 1 \quad (1)$$

Demikian pula proses sebaliknya untuk merubah dari bilangan biner ke bentuk bilangan float maka proses pertama kali adalah dengan mendapatkan nilai integer dari bilangan biner tersebut baru kemudian dihitung nilai floatnya.

Rumus yang dapat digunakan untuk merubah bilangan biner ke float adalah:

$$x = a + integer * \frac{(b-a)}{2^n - 1} \quad (2)$$

dimana *integer* adalah nilai integer dari bilangan biner yang akan dirubah.

Sebagai contoh range solusi adalah $-3.2 < x < 12.4$ dengan nilai x yang diinginkan adalah bilangan float dengan kepresisian 4 angka dibelakang koma. Untuk merepresentasikan dalam bentuk chromosome biner langkahnya adalah sebagai berikut:

Pertama adalah menentukan panjang biner yang diperlukan menggunakan persamaan (1):

$$2^{n-1} < (b-a) * 10^m \leq 2^n - 1$$

karena jumlah angka dibelakang koma yang diinginkan adalah 4 digit maka $m = 4$, sehingga:

$$= 2^{n-1} < (12,4 - (-3,2)) * 10^4 \leq 2^n - 1$$

$$= 2^{n-1} < (15,6 * 10^4) \leq 2^n - 1$$

$$= 2^{n-1} < 156000 \leq 2^n - 1$$

Nilai n yang memenuhi persamaan diatas adalah $n = 18$ karena $2^{17} < 156000 \leq 2^{18}$

Setelah diketahui panjang biner yang diperlukan, langkah-langkah berikutnya sama seperti proses pengubahan integer ke biner.

Sedangkan contoh untuk merubah dari biner menjadi float adalah berikut:

Misalnya diketahui bilangan binernya adalah: 000010010101000110

Untuk merubah bilangan biner tersebut menjadi bilangan float prosedurnya adalah:

Langkah pertama adalah mendapatkan nilai integer dari bilangan biner tersebut:

Nilai integer dari 000010010101000110 adalah 9542

Selanjutnya adalah mendapatkan nilai bilangan float berdasarkan nilai integer diatas menggunakan persamaan (2):

Maka nilai dalam bilangan float:

$$= -3.2 + 9542 * \frac{12.4 - (-3.2)}{2^{15} - 1}$$

$$= -3.2 + (9542) * (15.6 / 32767)$$

$$= 1.3428388317514572588274788659322$$

$$= 1.343$$

Kesimpulan

Chromosome digunakan untuk merepresentasikan variabel dari permasalahan yang akan diselesaikan ke bentuk yang dapat diproses oleh algoritma genetika. Salah satu bentuk representasi chromosome yang banyak digunakan adalah dalam bentuk biner. Proses untuk merubah bilangan ke bentuk biner dan dari biner ke bentuk asalnya dilakukan dengan mengetahui terlebih dahulu nilai bilangan yang akan dirubah dalam bentuk integer.

Referensi

- [1] Denny Hermawanto, "Algoritma Genetika dan Contoh Aplikasinya", <http://ilmukomputer.com/2007/03/29/algoritma-genetika-dan-contoh-aplikasinya/>
- [2] Mitsuo Gen, Runwei Cheng, "Genetic Algorithm and Engineering Design", John Willey & Son Inc., 1997