

PENERAPAN ALGORITMA GENETIKA DENGAN CROSSOVER CUT AND SPLICE DALAM OPTIMASI ROUTING JARINGAN

Inge Martina

Departemen Teknik Informatika Institut Teknologi Harapan Bangsa

Jl. Dipati Ukur 80-84 Bandung

Telp. +62 66 250 6636 / Fax. +62 22 250 7901

inge@ithb.ac.id

Abstrak— Routing jaringan adalah pemilihan jalur dalam jaringan yang dipakai untuk mengirimkan data. Di dalam jaringan komputer, data akan dikirimkan dari sumbernya ke tujuan akhir melalui titik-titik perantara. Jalur yang dilalui akan mempengaruhi kecepatan pengiriman (delay) dan banyaknya paket data yang hilang. Pencarian routing jaringan adalah masalah pencarian jalur terpendek (shortest path). Paper ini bertujuan untuk mengusulkan sebuah algoritma untuk pencarian routing jaringan dengan algoritma genetika yang memakai crossover cut and splice..

Kata kunci— routing jaringan, shortest path, algoritma genetika, crossover cut and splice

Abstract— Network routing is the process of selecting paths in a network along which to send data. In computer network, the data will be sent from the source toward the ultimate destination through intermediate nodes. The path will influence the transmission speed (delay) and data packet losses. Network routing finding can be classified as shortest path problem. This paper proposes an algorithm for finding network routing by adapting genetic algorithm with cut and splice crossover.

Keywords— network routing, shortest path, genetic algorithm, cut and splice crossover

I. PENDAHULUAN

Tujuan pembangunan jaringan komputer adalah mendukung komunikasi antar pemakai melalui pengiriman data yang tepat waktu dan tepat isi. Tetapi sering terjadi kualitas jaringan tidak bagus karena data yang diterima tidak tepat isi atau tidak tepat waktu.

Routing jaringan adalah pemilihan jalur dalam jaringan yang dipakai untuk mengirimkan data. Di dalam jaringan komputer, data akan dikirimkan dari sumbernya ke tujuan akhir melalui titik-titik perantara. Jalur yang dilalui akan mempengaruhi kecepatan pengiriman (delay) dan banyaknya paket data yang hilang. Pencarian routing jaringan adalah masalah pencarian jalur terpendek (shortest path).

Algoritma genetika mensimulasikan proses evolusi makhluk hidup yang melakukan perkawinan antar dua individu, seleksi individu berdasarkan nilai terbaik untuk bertahan hidup serta mutasi.

Dalam masalah ini, sebuah individu adalah sebuah routing jaringan dengan nilai fitnessnya adalah total packet loss. Banyaknya gen dalam sebuah individu bervariasi karena titik perantara dapat bervariasi. Proses crossover harus menghasilkan individu yang merupakan routing jaringan valid, sehingga crossover yang cocok adalah *cut and splice*, yaitu pemilihan titik crossover pada individu orang tua secara acak.

II. JARINGAN KOMPUTER

Jaringan komputer adalah sekelompok komputer yang saling dihubungkan dengan menggunakan suatu protokol komunikasi sehingga antara satu komputer dengan komputer yang lain dapat berbagi data atau berbagi sumber daya (*sharing resources*).

Di dalam jaringan komputer data dikirimkan dalam bentuk paket-paket. Pengiriman data dari satu titik ke titik lainnya sering kali memerlukan titik-titik perantara yang berupa sebuah perangkat keras yang bernama router. Router bekerja pada layer 3 OSI yakni layer network yang berarti melewatkan paket ke alamat yang dituju. Jalur yang dipakai pengiriman data akan mempengaruhi kualitas pengiriman dalam hal packet yang hilang (*packet loss*) dan waktu tunda (*delay*). Pencarian jalur tersebut disebut routing jaringan. Pencarian routing jaringan adalah masalah pencarian jalur terpendek (shortest path).

III. ALGORITMA GENETIKA

Algoritma genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Algoritma genetika merupakan evolusi atau perkembangan dunia komputer yang muncul karena terinspirasi oleh ilmu dalam biologi.

Algoritma ini dapat dipakai untuk mendapatkan solusi optimal dari sebuah masalah. Sebelum algoritma ini dijalankan, masalah yang akan dioptimalkan itu harus dinyatakan dalam fungsi tujuan, yang dikenal dengan fungsi *fitness*. Kromosom dengan nilai *fitness* yang tinggi akan memberikan probabilitas yang tinggi untuk bereproduksi pada generasi selanjutnya, sehingga untuk setiap generasi pada proses evolusi, fungsi *fitness* mensimulasikan seleksi alam dan akan mengarah ke populasi dengan *fitness* yang membaik.

Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: *operator reproduksi*, *operator crossover* (persilangan) dan *operator mutasi*. Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal, populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.
2. Membentuk generasi baru menggunakan tiga operator genetika yang telah disebut di atas.
3. Evaluasi solusi, proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Beberapa kriteria berhenti yang sering digunakan antara lain:
 - Berhenti pada generasi tertentu.
 - Berhenti setelah dalam beberapa generasi berturut-turut nilai *fitness* tertinggi tidak berubah.

A. Crossover

Crossover atau perkawinan silang merupakan proses mengkombinasikan dua individu untuk memperoleh individu-individu baru yang diharapkan mempunyai *fitness* lebih baik. Tidak semua pasangan induk mengalami proses *crossover*, banyaknya pasangan induk yang mengalami *crossover* ditentukan dengan nilai probabilitas *crossover*.

Ada beberapa jenis *crossover*, yaitu *one-point crossover*, *n-point crossover*, *order crossover*, *partially mapped crossover*, dan *cut and splice crossover*. Untuk menyelesaikan masalah dalam paper ini dipakai *cut and splice crossover*. *Cut and splice crossover* mempunyai beberapa karakteristik :

- Pemilihan secara bebas titik *crossover* dari orang tua.



- *Crossover* akan menghasilkan anak dengan panjang kromosom yang berubah.



Gambar 1 Cut and Splice Crossover

B. Encoding

Teknik pengkodean atau encoding adalah bagaimana mengkodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

Beberapa teknik pengkodean yang sering dipakai adalah *binary encoding*, *permutation encoding*, *value encoding*, dan *tree encoding*. Dalam *value encoding* setiap kromosom dapat berupa suatu nilai. Nilai bisa apa saja yang dihubungkan

dengan masalah, bentuk angka, bilangan real atau karakter ke beberapa objek rumit. Dalam memakai *value encoding* ini diperlukan pengembangan beberapa *crossover* dan mutasi baru untuk masalah yang spesifik.

IV. ANALISIS

Berikut ini metode atau teknik algoritma genetika yang digunakan dalam pengembangan aplikasi optimasi jaringan ini:

1. Inisialisasi Populasi
2. Evaluasi individu
3. Cut and splice crossover

A. Inisialisasi Populasi

Koneksi jaringan dimodelkan menggunakan graph berarah berbobot. Bobot dari koneksi tersebut diperoleh dari nilai *packet loss* yang didapat dari perintah *ping*, seperti contoh berikut :

ping statistics for 64.233.181.99:

packets: sent = 23931, received = 23713, lost = 218

approximate round trip times in milli-seconds:

minimum = 74ms, maximum = 1180ms, average = 156ms

Dari data *packet* tersebut, bobot graph dihitung menggunakan rumus di bawah ini :

$$p = \frac{\text{packet loss}}{\text{packet loss} + \text{packet received}}$$

$$p = \frac{218}{218 + 23713}$$

$$p = 0.009105523 = 0.9\%$$

Tabel di bawah ini adalah contoh data koneksi antara 11 router. Tabel tersebut merupakan *adjacency matrix* dari graph jaringan. Jika antara dua router tidak ada koneksi, maka diberi nilai 1000 sebagai nilai terbesar.

TABEL 1 : ADJACENCY MATRIX UNTUK JARINGAN KOMPUTER

		To										
Packet Loss		0	1	2	3	4	5	6	7	8	9	10
from	0	1000	0.3	0.2	1	1000	1000	1000	1000	1000	1000	1000
	1	1000	1000	0.1	1000	0.15	1000	1000	1000	1000	1000	1000
	2	1000	1000	1000	1000	0.1	0.6	1000	1000	1000	1000	1000
	3	1000	1000	1000	1000	0.5	1000	1000	1000	1000	1000	1000
	4	1000	0.1	1000	1000	1000	0.7	0.9	0.1	1000	1000	1000
	5	1000	1000	1000	1000	1000	1000	1000	1000	1000	1.1	1000
	6	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
	7	1000	1000	1000	1000	1000	0.1	1000	1000	0.3	1000	1000
	8	1000	1000	1000	1000	1000	1000	1000	1000	1000	1.1	1000
	9	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1
	10	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Inisialisasi populasi akan menghasilkan individu-individu secara acak. Setiap individu mewakili sebuah route dari titik pengirim ke titik penerima. Karena panjang kromosom dari setiap individu tidak selalu sama, maka setiap individu direpresentasikan menggunakan list berkait. Setiap simpul mempunyai tiga field, yaitu : *origin* (titik asal), *destination* (titik tujuan), dan nilai *packet loss* (p). Dengan data masukan berupa titik pengirim dan titik penerima, algoritma inisialisasi populasi yang diperlukan adalah sebagai berikut :

- 1 : Buat simpul baru dan isi field *origin* dengan pengirim

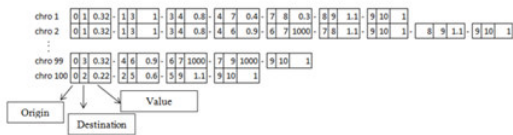
2 : Cari titik lain yang belum ada dalam list berkait tersebut secara acak dan isikan ke field *destination*.

3 : Isi field *value* dengan nilai yang sesuai dari *adjacency matrix*.

4 : Jika router yang diperoleh dari langkah 2 adalah titik penerima maka berhenti.

5 : Buat simpul baru dan isi field *origin* dengan router terakhir lalu ulangi langkah 2.

Algoritma di atas akan menghasilkan satu individu, dan proses tersebut diulang sebanyak maksimum individu yang diinginkan.



Gambar 2 Inisialisasi Populasi

B. Evaluasi Individu

Setiap individu dievaluasi dengan menghitung nilai fitnessnya yaitu dengan menjumlahkan field *value* dari setiap simpul dari sebuah list berkait.



Gambar 3 Nilai Fitness

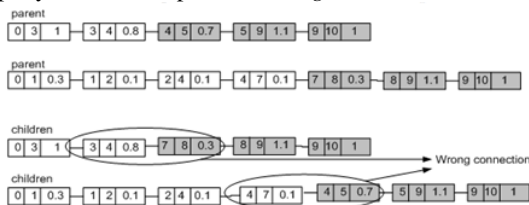
Dalam masalah ini, diperlukan pencarian nilai minimum fitness. Sebuah individu yang mempunyai nilai lebih dari 1000 bukanlah sebuah solusi karena mengandung simpul yang tidak terhubung dalam jaringan.

C. Cut and Splice Crossover

Bagaimana *cut and splice* crossover diterapkan dalam masalah routing jaringan? Beberapa asumsi yang diperlukan :

1. Persilangan antara dua individu yang bagus juga akan menghasilkan individu yang bagus, oleh sebab itu akan dipilih sejumlah individu dengan nilai bagus untuk dicrossover.

2. Pemilihan titik crossover merupakan hal penting karena tidak boleh dipilih titik crossover dari dua individu yang tidak mempunyai kesamaan pada field *origin*.



Gambar 4 Crossover yang Salah

Dengan dua asumsi di atas, maka algoritma crossover adalah sebagai berikut :

- 1 : Pilih sejumlah populasi dengan nilai terbaik.
- 2 : Pilih sebuah individu secara acak, dan pilih titik crossover secara acak dan menjadi kandidat 1.
- 3 : Pilih individu lain secara acak dan menjadi kandidat 2.

4 : Jika kandidat 2 tidak mempunyai titik crossover yang sama dengan kandidat 1, ulangi langkah 3.

5 : Buat anak baru dari kandidat 1 dan kandidat 2, dan keluarkan kandidat 1, kandidat 2, dan dua hasil persilangannya dari daftar yang akan dicrossover.

6 : Ulangi langkah 2 sampai tidak ada individu yang bisa dicrossover.

D. Algoritma Global

Algoritma global untuk masalah ini dibangun dari algoritma yang telah dibahas di atas.

- 1 : Inisialisasi beberapa parameter yang diperlukan :
 - a. Banyaknya populasi maksimum
 - b. Banyaknya kandidat untuk crossover
 - c. Banyaknya generasi yang dieksekusi
 - d. Banyaknya generasi berurutan dengan nilai fitness terbaik sama dan memberhentikan iterasi
- 2 : Masukkan data jaringan ke *adjacency matrix*
- 3 : Inisialisasi populasi dan catat nilai terbaik
- 4 : Proses *cut and splice crossover*
- 5 : Dari hasil crossover, pilih populasi terbaik sebanyak populasi maksimum
- 6 : Jika parameter c atau d belum terpenuhi, ulangi langkah 4.

V. HASIL PERCOBAAN

Bagian ini akan menunjukkan hasil dari beberapa percobaan menggunakan nilai parameter-parameter berikut :

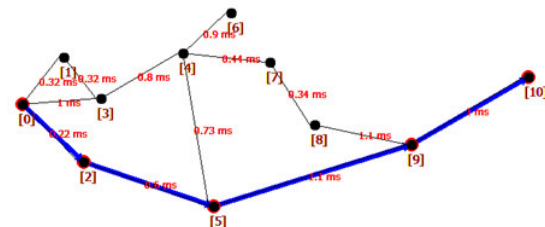
- a. Banyaknya populasi maksimum = 100
- b. Banyaknya kandidat untuk crossover = 50%
- c. Banyaknya generasi yang dieksekusi = 50
- d. Banyaknya generasi berurutan dengan nilai fitness terbaik sama dan memberhentikan iterasi = 5

Percobaan 1 :

Data input :

- Banyaknya node = 11
- Pengirim = 0
- Penerima = 11.

Hasil :



Fitness value : 2.92

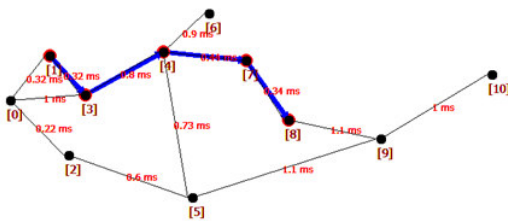
Route sequence: 0 → 2 → 5 → 9 → 10

Percobaan 2 :

Data input :

- Banyaknya node = 11
- Pengirim = 1
- Penerima = 8

Hasil :



Fitness value : 1.9

Route sequence: 1 → 3 → 4 → 7 → 8

Percobaan 3 :

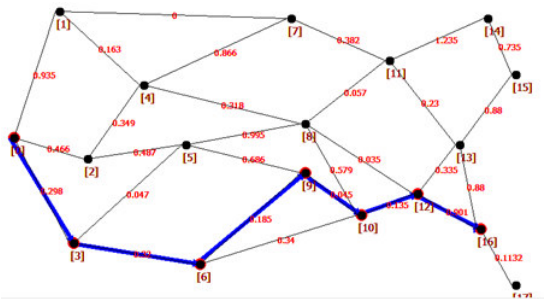
Data input :

Banyaknya node = 18

Pengirim = 0

Penerima = 16

Hasil :



Fitness value : 1.884

Route sequence: 0→3→6→9→10→12→16

VI. KESIMPULAN

Kesimpulan yang dapat diambil dari percobaan ini adalah :

1. Algoritma genetika dapat menghasilkan routing jaringan optimum
2. Aplikasi memberikan visualisasi hasil yang dapat membantu pemakai memahami proses routing

PENELITIAN LEBIH LANJUT

Performance dari algoritma genetika sangat dipengaruhi oleh banyaknya iterasi yang dieksekusi terutama dihubungkan dengan kepadatan graph. Banyaknya iterasi optimum perlu diteliti lebih lanjut. Hal lain yang sangat penting adalah proses penentuan titik crossover karena dalam routing jaringan titik crossover harus dipilih dengan teliti.

REFERENSI

- [1] Forouzan, Behrouz A., DeAnza College (2007). Data Communication and Networking, Fourth Edition. McGraw-Hill Representative.
- [2] [2] Leon-Garcia, Alberto & Indra Widjaja (2001). Communication Networks: Fundamental Concepts and Key Architectures. McGraw-Hill Representative.
- [3] [3] Obitko, Marek (1998). Introduction to Genetic Algorithms [online]. http://www.obitko.com/tutorials/genetic_algorithms/index.php.
- [4] [4] Pengantar Jaringan Komputer – LAN (2008). [online]. <http://www.sejutablog.com/pengantar-jaringan-komputer-lan/>
- [5] [5] Rennards (2002). Introduction to Genetic Algorithms. [online]. <http://www.rennard.org/alife/english/gavintrgb.html>
- [6] [6] Stallings, William (1996). Data and Computer Communication, Fifth Edition. Prentice Hall.