

PENERAPAN ALGORITMA GENETIKA PADA OPTIMASI MODEL PENUGASAN

Wayan Firdaus Mahmudy

Jurusan Matematika, FMIPA, Universitas Brawijaya

ABSTRAK

Model penugasan (*assignment model*) merupakan kasus khusus dari model pemrograman linier (*linear programming*). Karakteristik khusus yang ada pada model khusus ini diantaranya ialah cenderung membutuhkan pembatas (*constrains*) dan variabel yang sangat banyak sehingga penggunaan komputer dalam penyelesaiannya dengan model matematis (misalnya dengan metode *simplex*) akan sangat mahal atau proses perhitungannya sangat panjang dan tidak praktis. Masalah akan bertambah jika ada pembatas khusus yang sulit dibuat model matematikanya.

Pada penelitian ini digunakan algoritma genetika untuk menyelesaikan masalah penugasan. Algoritma genetika merupakan salah satu metode heuristik yang merupakan cabang dari *evolutionary algorithm*, yaitu suatu teknik untuk memecahkan masalah-masalah optimasi yang rumit dengan menirukan proses evolusi makhluk hidup.

Masalah penugasan yang diselesaikan dalam penelitian ini adalah masalah penugasan mengajar dosen. Setiap dosen diberi formulir untuk memilih matakuliah mulai dari yang paling disukai. Semakin tinggi tingkatan kesukaan dosen terhadap matakuliah tertentu berarti biayanya (dalam matriks biaya/*cost*) semakin rendah.

Implementasi algoritma genetika dengan program komputer menghasilkan keluaran pembagian mengajar dosen dengan total biaya terkecil. Solusi yang dihasilkan sama dengan solusi dari metode deterministik menggunakan metode *simplex*. Waktu penyelesaiannya juga relatif cepat. Pada matriks biaya berukuran 16 x 16 solusi optimal dihasilkan dalam waktu kurang dari 5 detik.

ABSTRACT

Assignment model is a special case a linear programming. Special characteristic on this model needs much constrains. Using a software to find solution with mathematic model will be very expensive and unpractical.

This research applies genetic algorithm to find the solution of assignment model. Assignment problem to be solved is lecturer assignment. Every lecturer chooses some favourite course. Higher favorite course degree has lower cost.

An implementation of genetic algorithm by using software produce solutions of lecturer assignment with smallest cost. This solutions are equivalent with deterministic method using *simplex* methods. On cost matrix has 16 x 16 size, optimal solutions produced less than 5 seconds.

1. Pendahuluan

Model penugasan (*assignment model*) merupakan kasus khusus dari model pemrograman linier (*linear programming*). Karakteristik khusus yang ada pada model khusus ini diantaranya ialah cenderung membutuhkan pembatas (*constrains*) dan variabel yang sangat banyak sehingga penggunaan komputer dalam penyelesaiannya dengan model matematis (misalnya dengan metode *simplex*) akan sangat mahal atau proses perhitungannya sangat panjang dan tidak praktis. Masalah akan bertambah jika ada pembatas khusus yang sulit dibuat model matematikanya (Taha, 1993).

Untuk mengatasi kasus khusus seperti di atas dapat digunakan metode *heuristik*, yaitu suatu metode pencarian yang didasarkan atas intuisi atau aturan-aturan empiris untuk memperoleh solusi yang lebih baik daripada solusi yang telah dicapai sebelumnya (Gen, 1997).

Algoritma genetika merupakan salah satu metode heuristik yang merupakan cabang dari *evolutionary algorithm*, yaitu suatu teknik untuk memecahkan masalah-masalah optimasi yang rumit dengan menirukan proses evolusi makhluk hidup. Algoritma genetika berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat. Algoritma ini banyak digunakan dalam bidang fisika, biologi, ekonomi, sosiologi dan lain-lain yang sering menghadapi masalah optimasi yang model matematikanya kompleks atau bahkan sulit dibangun (Rennard, 2000).

Model Penugasan

Pada model penugasan terdapat sejumlah m sumber ditugaskan kepada sejumlah n tujuan (satu sumber untuk satu tujuan) sedemikian sehingga didapat ongkos total minimum.

Biasanya yang dimaksud dengan sumber ialah pekerjaan (atau pekerja), sedangkan yang dimaksud dengan tujuan ialah mesin-mesin (dalam masalah pendidikan pekerja adalah dosen dan mesin adalah mata kuliah). Jadi terdapat m pekerjaan yang ditugaskan kepada n mesin yang apabila pekerjaan i ($i = 1, 2, 3, \dots, m$) ditugaskan kepada mesin j ($j = 1, 2, 3, \dots, n$) akan muncul ongkos penugasan c_{ij} . Karena satu pekerjaan ditugaskan hanya pada satu mesin, maka *supply* yang dapat digunakan pada setiap sumber adalah 1. (atau $a_i=1$, untuk seluruh i). Demikian pula halnya dengan mesin-mesin; karena satu mesin hanya dapat menerima satu pekerjaan, maka *demand* dari setiap tujuan adalah 1 (atau $b_j=1$, untuk seluruh j). Jika ada suatu pekerjaan yang tidak dapat ditugaskan pada mesin tertentu, maka c_{ij} yang berkorespondensi dengannya dinyatakan sebagai M yang merupakan ongkos yang sangat tinggi (Taha, 1993).

Penggambaran umum persoalan penugasan adalah sebagai berikut:

		Mesin			
		1	2	...	N
Pekerjaan	1	C_{11}	c_{12}	...	c_{1n}
	2	C_{21}	c_{22}	...	c_{2n}

	m	c_{m1}	c_{m2}	...	c_{mn}

Secara matematis, model penugasan dapat dinyatakan sebagai berikut:

$$x_{ij} = \begin{cases} 0, & \text{jika pekerjaan ke - i tidak ditugaskan pada mesin ke - j} \\ 1, & \text{jika pekerjaan ke - i ditugaskan pada mesin ke - j} \end{cases}$$

Dengan demikian, model persoalan penugasan ini adalah:

$$\text{Minimumkan: } z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

berdasarkan pembatas:

$$\sum_{j=1}^m x_{ij} = 1, \quad i=1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j=1, 2, \dots, m$$

$$x_{ij} = 0 \text{ atau } 1$$

Dalam dunia pendidikan terdapat masalah seperti:

- jika seorang dosen sudah ditugaskan di mata kuliah semester satu maka penugasan kedua adalah pada mata kuliah semester lima.
- ada penugasan paralel yang berkaitan, misalnya seorang dosen yang ditugaskan mengajar di program Strata Satu dan Strata Dua. Jika dosen A mengajar di Strata Satu lebih dari 5 sks maka penugasan di Strata Dua maksimal 3 sks
- jika dosen A mengajar mata kuliah X maka dia juga harus mengajar mata kuliah Y.

Algoritma Genetika

Algoritma genetika adalah salah satu cabang *evolutionary algorithms*, yaitu suatu teknik optimasi yang didasarkan pada genetika alami. Dalam algoritma genetika untuk menghasilkan suatu solusi optimal,

proses pencarian dilakukan di antara sejumlah alternatif titik optimal berdasarkan fungsi probabilistik.

Apabila dibandingkan dengan prosedur pencarian dan optimasi biasa, algoritma genetika berbeda dalam beberapa hal sebagai berikut (Michalewicz,1996):

- Manipulasi dilakukan terhadap kode dari himpunan parameter, tidak secara langsung terhadap parameternya sendiri.
- Proses pencarian dilakukan dari suatu titik populasi, tidak dari satu titik saja.
- Proses pencarian menggunakan informasi dari fungsi tujuan.
- Pencariannya menggunakan *stochastic operators* yang bersifat probabilistik, tidak menggunakan aturan deterministik.

Masalah utama pada algoritma genetika adalah bagaimana memetakan satu masalah menjadi satu string kromosom. Siklus perkembangan algoritma genetik diawali dengan pembuatan himpunan solusi baru (*initialization*) yang terdiri atas sejumlah string kromosom dan ditempatkan pada penampungan populasi. Kemudian dilakukan proses reproduksi dengan memilih individu-individu yang akan dikembangkan. Penggunaan operator-operator genetik seperti pindah silang (*crossover*) dan mutasi (*mutation*) terhadap individu-individu yang terpilih dalam penampungan individu akan menghasilkan keturunan atau generasi baru. Setelah proses evaluasi untuk perbaikan populasi, maka generasi-generasi baru ini akan menggantikan himpunan populasi asal. Siklus ini akan berlangsung berulang kali sampai tidak dihasilkan perbaikan keturunan, atau sampai kriteria optimum ditemukan.

Apabila $P(t)$ dan $C(t)$ merupakan *parents* dan *children* pada generasi ke- t , maka struktur umum algoritma genetika dapat dideskripsikan sebagai berikut:

```

procedure AlgoritmaGenetika
begin
  t = 0
  inisialisasi P(t)
  while (bukan kondisi berhenti) do
    evaluasi P(t)
    seleksi P(t)
    reproduksi C(t) dari P(t)
    bentuk P(t+1) dari P(t) dan C(t) terbaik
    t = t + 1
  end while
end
  
```

Inisialisasi Populasi

Siklus perkembangan algoritma genetik diawali dengan pembuatan himpunan solusi baru (*initialization*) yang terdiri atas sejumlah string kromosom dan ditempatkan pada penampungan populasi. Populasi awal sebagai daerah pencarian solusi optimal dibangkitkan secara acak.

Representasi kromosom diperlukan untuk menjelaskan setiap individu dalam populasi. Setiap individu atau kromosom tersusun atas urutan gen dari suatu alphabet. Suatu alfabet dapat terdiri dari digit biner (0 dan 1), *floating point*, *integer*, symbol-simbol (seperti A, B, C), matriks, dan lain sebagainya (Houck , 1999).

Pada masalah penugasan mengajar dosen misalkan terdapat 10 mata kuliah dan 5 orang dosen maka bisa dibuat individu / kromosom pertama sebagai berikut:

1	1	2	2	3	3	4	4	5	5
---	---	---	---	---	---	---	---	---	---

Gambar 1. Individu Pertama

Panjang kromosom sesuai dengan banyaknya matakuliah. Angka dalam setiap sel (gen) merupakan nomer dosen. Dari Gambar 1 bisa dibaca dosen ke-1 memegang mata kuliah ke-1 dan ke-2. Dosen ke-2 memegang mata kuliah ke-3 dan ke-4, demikian seterusnya.

Pada Gambar 1 diasumsikan setiap dosen memegang dua mata kuliah. Jika diinginkan banyaknya matakuliah yang dipegang tidak sama maka bisa dibuat sebagai berikut:

1	1	1	2	3	3	3	3	5	5
---	---	---	---	---	---	---	---	---	---

Gambar 2. Individu Pertama dengan Distribusi Gen Tidak Merata

Evaluasi Individu

Hal penting dalam algoritma genetika adalah pemilihan individu / kromosom untuk menghasilkan keturunan berikutnya. Pemilihan dilakukan berdasarkan nilai kesesuaian (*fitness value*) setiap individu.

Pada masalah penugasan terdapat matriks biaya. Dari setiap individu bisa dihitung biaya total. Karena tujuan optimasi adalah meminimalkan total biaya maka *fitness value* bisa dihitung sebagai berikut:

$$fitness\ value = konstanta - total\ biaya$$

Konstanta dihitung dengan cara menjumlahkan nilai elemen matriks biaya terbesar pada tiap baris.

Seleksi Populasi

Seleksi merupakan proses untuk memilih individu yang akan menjadi induk pada proses reproduksi. Pemilihan dilakukan secara probabilistik. Individu yang mempunyai *fitness value* lebih besar mempunyai peluang lebih besar untuk menjadi induk dari individu berikutnya.

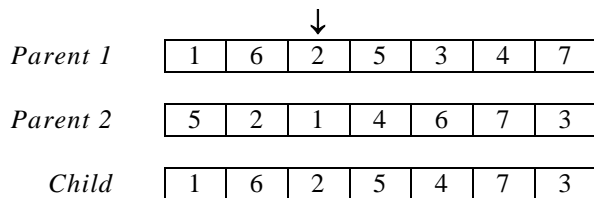
Untuk setiap individu *k* dengan nilai fungsi *fitness value* f_k maka peluang individu tersebut terpilih sebagai induk adalah:

$$p_k = \frac{f_k}{\sum_{j=1}^n f_j} \quad ; n : ukuran\ populasi$$

Reproduksi

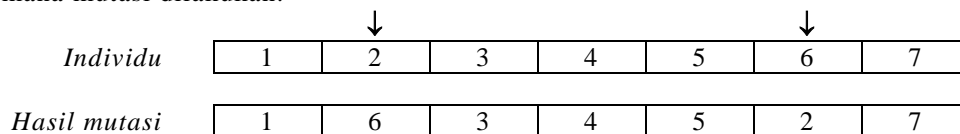
Individu baru dihasilkan dari dua induk dengan melakukan tukar silang (*crossover*). Tingkat tukar silang (*crossover rate*) yang tinggi memungkinkan eksplorasi daerah solusi dan mengurangi resiko pencarian terjebak dalam *local maksima*. Namun hal ini membuat daerah pencarian terlalu luas.

Tukar silang yang dipakai dalam penelitian ini adalah tukar silang 1 titik dengan memilih titik potong secara random dan mempertukarkan daerah sesudah titik tukar antar *parents*.



Gambar 3 Ilustrasi tukar silang 1 titik.

Individu yang baru terbentuk bisa mengalami proses mutasi. Gambar 4 di bawah ini menunjukkan bagaimana mutasi dilakukan.



Gambar 4 Ilustrasi mutasi dengan 2 titik tukar

2. Metode Penelitian

Data yang dipakai dalam penelitian ini adalah penugasan mengajar pada Program Studi Manajemen Informatika Jurusan Matematika, Fakultas MIPA Universitas Brawijaya. Setiap dosen diberi formulir untuk memilih matakuliah mulai dari yang paling disukai. Semakin tinggi tingkat kesukaan dosen terhadap matakuliah tertentu berarti biayanya (dalam matriks biaya/*cost*) semakin rendah. Data ini dikonversi menjadi matriks biaya. Jika dosen tidak memilih satu matakuliah maka pada matriks biaya akan diberi nilai 100. Matriks biaya ini menjadi masukan bagi algoritma genetika yang telah diimplementasikan dalam bentuk perangkat lunak.

3. Hasil Dan Pembahasan

Perangkat lunak yang dibuat pada penelitian ini dikhususkan pada masalah penugasan mengajar dosen. Perangkat lunak menghasilkan output pembagian mengajar dosen dengan total biaya terkecil. Uji coba program dilakukan menggunakan komputer dengan prosesor Pentium IV dan memori 256 kb.

Hasil ujicoba dibandingkan dengan solusi optimal yang dihasilkan dengan mencoba semua kemungkinan susunan pengajar.

Uji Coba dengan Pembagian Kelas Merata

Pada uji coba ini dimasukan data kesukaan dosen terhadap matakuliah sebagai berikut

DAFTAR DOSEN

Nama	Beban MK	Urutan MK	Kesukaan
1. Ach. Ridok	1	2 1 4	
2. Achmad Basuki	1	9 1 10 6	
3. Bondan	1	2 7 9 8	
4. Dian Eka R	1	4 5 2	
5. Edi Santoso	1	3 1 9 2 7 6 8	
6. Kasyful	1	4 3 1	
7. Marji	1	4 2 7	
8. Muh.Arif Rahman	1	3 4 6	
9. Nurul Hidayat	1	2 10 9	
10. Wayan FM	1	9 7 5 6	

DAFTAR MATAKULIAH

Matakuliah	Kelas
1. Algoritma	1
2. Analisis Sistem Informasi	1
3. Arsitektur Komputer	1
4. Bahasa Pemrograman 1 (Pascal)	1
5. Bahasa Pemrograman 2 (Delphi)	1
6. Bahasa Pemrograman 3 (C++)	1
7. Bahasa Pemrograman 4 (Visual Basic)	1
8. Bahasa Pemrograman 5 (Prog Internet)	1
9. Basis Data Lanjutan	1
10. Desain Multimedia	1

Sebelum diproses dengan algoritma genetika, program menghitung matriks biaya sebagai berikut:

2	2	100	100	2	3	100	100	100	100
1	100	1	3	4	100	2	100	1	100

ORIGINAL ARTICLE

100	100	100	100	1	2	100	1	100	100
3	100	100	1	100	1	1	2	100	100
100	100	100	2	100	100	100	100	100	3
100	4	100	100	6	100	100	3	100	4
100	100	2	100	5	100	3	100	100	2
100	100	4	100	7	100	100	100	100	100
100	1	3	100	3	100	100	100	3	1
100	3	100	100	100	100	100	100	2	100

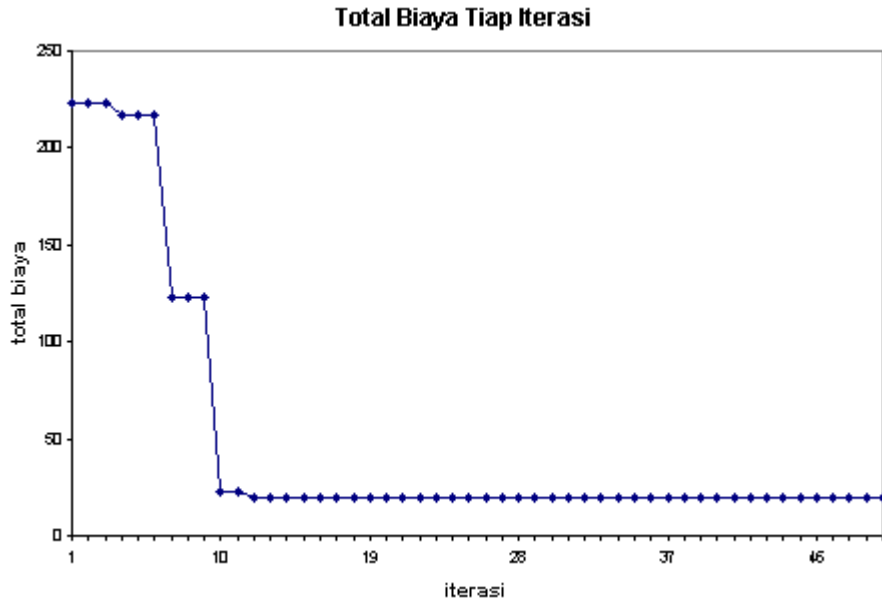
Banyaknya baris matriks sesuai dengan banyaknya kelas matakuliah. Banyaknya kolom sesuai dengan banyaknya dosen. Nilai yang ada pada matriks di atas merupakan urutan kesukaan dosen terhadap matakuliah. Nilai 100 menunjukkan dosen tidak memilih matakuliah tersebut.

Individu pertama (ukuran populasi=50) dibangkitkan secara acak. Pada akhir iterasi (iterasi ke-60, kondisi konvergen, waktu proses 3 detik) dihasilkan kromosom sebagai berikut

Iterations	60	Chromosom	Fitness	Total Cost
1.	1	7 5 6 4 8 10 3 2 9	-> 980	20
2.	5	1 6 7 4 8 10 3 2 9	-> 980	20
3.	6	1 5 7 4 8 10 3 2 9	-> 980	20
4.	5	1 6 4 10 8 7 3 2 9	-> 978	22
5.	6	1 5 4 10 8 7 3 2 9	-> 978	22
6.	1	5 6 7 4 8 10 3 2 9	-> 977	23
7.	1	7 5 6 4 8 10 3 9 2	-> 977	23
8.	5	1 6 7 4 8 10 3 9 2	-> 977	23
9.	5	1 6 8 4 10 7 3 2 9	-> 977	23
10.	5	1 6 8 4 2 7 3 10 9	-> 977	23
11.	6	1 5 7 4 8 10 3 9 2	-> 977	23
12.	6	1 5 8 4 2 7 3 10 9	-> 977	23
13.	1	5 6 4 10 8 7 3 2 9	-> 975	25
14.	5	1 6 4 10 8 7 3 9 2	-> 975	25
15.	1	5 6 7 4 8 10 3 9 2	-> 974	26
16.	1	5 6 8 4 10 7 3 2 9	-> 974	26
17.	1	5 6 8 4 2 7 3 10 9	-> 974	26
18.	1	7 6 8 4 10 5 3 2 9	-> 974	26
19.	5	1 6 8 4 10 7 3 9 2	-> 974	26
20.	6	5 8 1 4 2 7 3 10 9	-> 973	27
21.	1	5 6 4 10 8 7 3 9 2	-> 972	28
22.	1	5 6 8 4 10 7 3 9 2	-> 971	29
23.	1	7 5 6 4 8 3 10 2 9	-> 884	116
24.	1	7 5 6 4 8 3 2 10 9	-> 884	116
25.	5	1 6 7 4 8 3 10 2 9	-> 884	116
26.	5	1 6 7 4 8 3 2 10 9	-> 884	116
27.	5	1 8 7 4 6 10 3 2 9	-> 884	116
28.	6	1 5 7 4 8 3 10 2 9	-> 884	116
29.	6	1 5 7 4 8 3 2 10 9	-> 884	116
30.	1	7 5 6 10 8 3 4 2 9	-> 883	117
31.	5	1 6 7 10 8 3 4 2 9	-> 883	117
32.	6	1 5 7 10 8 3 4 2 9	-> 883	117
33.	1	7 5 4 6 8 10 3 2 9	-> 882	118
34.	1	7 5 6 4 8 10 2 3 9	-> 882	118
35.	1	7 5 6 4 8 2 3 10 9	-> 882	118
36.	1	7 5 8 4 6 10 3 2 9	-> 882	118
37.	5	1 6 4 7 8 10 3 2 9	-> 882	118
38.	5	1 6 7 4 8 10 2 3 9	-> 882	118
39.	5	1 6 7 4 8 2 3 10 9	-> 882	118
40.	5	1 6 8 4 7 10 3 2 9	-> 882	118
41.	5	1 7 6 4 8 10 3 2 9	-> 882	118

42.	6 1 5 4 7 8 10 3 2 9 ->	882	118
43.	6 1 5 7 4 8 10 2 3 9 ->	882	118
44.	6 1 5 7 4 8 2 3 10 9 ->	882	118
45.	6 1 5 8 4 7 10 3 2 9 ->	882	118
46.	1 4 5 6 7 8 10 3 2 9 ->	881	119
47.	1 4 5 7 6 8 10 3 2 9 ->	881	119
48.	1 5 6 7 4 8 3 10 2 9 ->	881	119
49.	1 5 6 7 4 8 3 2 10 9 ->	881	119
50.	1 5 8 6 4 7 10 3 2 9 ->	881	119

Solusi terbaik (total biaya = 20) merupakan solusi optimal. Dengan mencoba semua kemungkinan diperlukan iterasi sebanyak 10! (sesuai banyaknya kelas) dengan waktu perhitungan 25 detik. Gambar 5 menunjukkan penurunan total biaya pada tiap iterasi.



Gambar 5. Grafik total biaya tiap iterasi

Dari hasil di atas program menghasilkan solusi penugasan sebagai berikut:

DAFTAR DOSEN MATAKULIAH

Matakuliah	Dosen
1. Algoritma	Ach. Ridok
2. Analisis Sistem Informasi	Marji
3. Arsitektur Komputer	Edi Santoso
4. Bahasa Pemrograman 1 (Pascal)	Kasyful
5. Bahasa Pemrograman 2 (Delphi)	Dian Eka R
6. Bahasa Pemrograman 3 (C++)	Muh.Arif Rahman
7. Bahasa Pemrograman 4 (Visual Basic)	Wayan FM
8. Bahasa Pemrograman 5 (Prog Internet)	Bondan
9. Basis Data Lanjutan	Achmad Basuki
10. Desain Multimedia	Nurul Hidayat

Hasil percobaan selengkapnya dengan pembagian kelas merata adalah sebagai berikut

Banyak Kelas	Semua Kemungkinan		Algoritma Genetika	
	Solusi Optimum	Waktu	Solusi Terbaik	Waktu
10	20	25	20	3
11	21	100	21	5

12	22	500	22	7
13	23	1000	23	10
14	24	5000	25	12
15	26	9500	27	15

Dari tabel di atas bisa dilihat jika banyak kelas semakin besar maka algoritma genetika hanya menghasilkan solusi yang mendekati optimal tetapi waktu perhitungannya relatif kecil.

Uji Coba dengan Pembagian Kelas Tidak Merata

Pada uji coba ini dimasukan data kesukaan dosen terhadap matakuliah sebagai berikut:

DAFTAR DOSEN

Nama	Beban MK	Urutan MK	Kesukaan
1. Ach. Ridok	1	12 2 1 4 3 7 5	
2. Achmad Basuki	1	9 1 10 6 2 12	
3. Bondan	2	10 7 12 9 8 3 2 5	
4. Dian Eka R	2	4 5 2 11 6 7	
5. Edi Santoso	2	3 1 9 2 7 6 8	
6. Kasyful	3	4 3 1 12 8 9	
7. Marji	1	4 2 7 11 3 6 5 8	
8. Muh.Arif Rahman	1	3 4 6 12 11 10	
9. Nurul Hidayat	1	2 10 9 12 5 4	
10. Wayan FM	2	9 7 5 11 6 3	

DAFTAR MATAKULIAH

Matakuliah	Kelas
1. Algoritma	1
2. Analisis Sistem Informasi	2
3. Arsitektur Komputer	3
4. Bahasa Pemrograman 1 (Pascal)	2
5. Bahasa Pemrograman 2 (Delphi)	1
6. Bahasa Pemrograman 3 (C++)	1
7. Bahasa Pemrograman 4 (Visual Basic)	1
8. Bahasa Pemrograman 5 (Prog Internet)	1
9. Basis Data Lanjutan	1
10. Desain Multimedia	1
11. Desain Web	1
12. Grafika Komputer	1

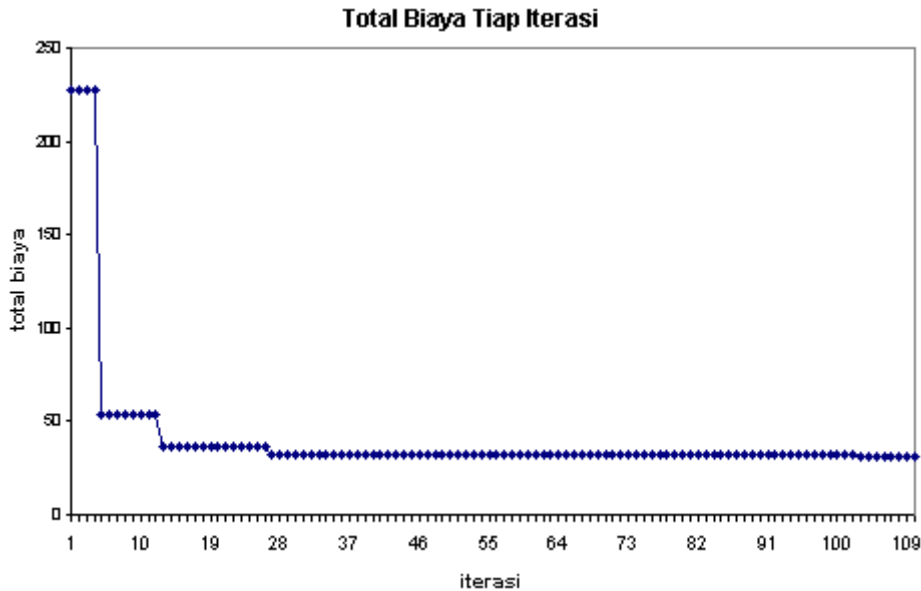
Pada data di atas beban mengajar dosen tidak sama. Matriks biaya yang dihasilkan sebagai berikut:

3	2	100	100	2	3	100	100	100	100
2	5	7	3	4	100	2	100	1	100
2	5	7	3	4	100	2	100	1	100
5	100	6	100	1	2	5	1	100	6
5	100	6	100	1	2	5	1	100	6
5	100	6	100	1	2	5	1	100	6
4	100	100	1	100	1	1	2	6	100
4	100	100	1	100	1	1	2	6	100
7	100	8	2	100	100	7	100	5	3
100	4	100	5	6	100	6	3	100	5
6	100	2	6	5	100	3	100	100	2
100	100	5	100	7	5	8	100	100	100

100 1 4 100 3 6 100 100 3 1
 100 3 1 100 100 100 100 6 2 100
 100 100 100 4 100 100 4 5 100 4
 1 6 3 100 100 4 100 4 4 100

Pada akhir iterasi (konvergen pada iterasi ke-110, waktu proses 4 detik) dihasilkan solusi terbaik dengan total biaya = 31 yang mendekati solusi optimal. Dengan mencoba semua kemungkinan dihasilkan solusi optimal dengan total biaya = 31, tetapi perhitungan ini memerlukan iterasi sebanyak 16! dengan waktu perhitungan perhitungan lebih dari 5 jam.

Gambar 6 menunjukkan penurunan total biaya pada tiap iterasi.



Gambar 8. Grafik total biaya tiap iterasi

Dari hasil di atas program menghasilkan tampilan solusi sebagai berikut:

DAFTAR DOSEN MATAKULIAH

Matakuliah	Dosen
1. Algoritma	Edi Santoso
2. Analisis Sistem Informasi	Marji
3. Analisis Sistem Informasi	Nurul Hidayat
4. Arsitektur Komputer	Edi Santoso
5. Arsitektur Komputer	Kasyful
6. Arsitektur Komputer	Kasyful
7. Bahasa Pemrograman 1 (Pascal)	Dian Eka R
8. Bahasa Pemrograman 1 (Pascal)	Kasyful
9. Bahasa Pemrograman 2 (Delphi)	Dian Eka R
10. Bahasa Pemrograman 3 (C++)	Muh.Arif Rahman
11. Bahasa Pemrograman 4 (Visual Basic)	Wayan FM
12. Bahasa Pemrograman 5 (Prog Internet)	Bondan
13. Basis Data Lanjutan	Achmad Basuki
14. Desain Multimedia	Bondan
15. Desain Web	Wayan FM
16. Grafika Komputer	Ach. Ridok

Pola hasil percobaan selengkapnya dengan pembagian kelas tidak merata hampir sama dengan pola

hasil percobaan dengan pembagian kelas merata. Pada banyak kelas sedikit, algoritma genetika memberikan hasil optimal, tapi jika banyak kelas semakin besar maka algoritma genetika hanya menghasilkan solusi yang mendekati optimal tetapi waktu perhitungannya relatif kecil.

Kesimpulan

- Algoritma genetika dapat menyelesaikan masalah penugasan dosen dalam waktu yang cepat. Solusi optimal diperoleh jika banyaknya kelas sedikit, jika banyak kelas semakin besar maka algoritma genetika hanya menghasilkan solusi yang mendekati optimal.

Saran

- Untuk mendapatkan hasil yang baik dan cepat, nilai tingkat tukar silang dan mutasi harus diatur sedemikian rupa. Perlu dilakukan penelitian untuk mendapatkan nilai ini secara otomatis bergantung masalah yang diselesaikan.

DAFTAR PUSTAKA

- Gen, Mitsuo dan Cheng, Runwei (1997). *Genetic Algorithms and Engineering Design*. John Wiley & Sons.
- Houck, Christopher R, Jeffrey K dan Michael G Kay. (1999). *A Genetic Algorithms for Function Optimization: A Matlab Implementation*. http://www.dai.ed.ac.uk/groups/evalg/eag_local_copies_op_papers_body.html.
- Michalewicz, Zbigniew. (1996). *Genetic Algorithms + Data Structures = Evaluation Programs*. Springer.
- Rennard, JP. (2000). *Genetic Algorithms Viewer*. www.rennard.org/alife.
- Taha, Hamdy A., (1993). *Operations Research: An Introduction*. Edisi ketiga. Macmillan Publishing Co, New York.