

ALGORITMA GENETIK UNTUK MENYELESAIKAN MASALAH OPTIMASI FUNGSI BERKENDALA DENGAN PENGKODEAN BILANGAN BULAT

Oleh : Yuliani Indrianingsih

Jurusan Teknik Informatika
Sekolah Tinggi Teknologi Adisutjipto (STTA)
e-mail : yulistta@gmail.com

Abstract

Function optimization problems related to the search for a set of solutions through the analysis process with several obstacles and constraints appropriate combination of optimization purposes. In many cases not easy to solve optimization problems with non-linear objective function exactly. In search of a solution to the problem is sometimes required complex mathematical formulations utuk provide a definite solution. Optimum solution requires a long calculation process and not practical. To solve this case requires heuristic methods..

Genetic algorithms can be used to solve the optimal solution, with the process of finding the optimal number of points based on probabilistic function. If there is one criteria considered to be single-purpose optimization problem, and if more become a multi-objective optimization problem. Genetic algorithms does not require a lot of math concepts, and can treat all forms of objective function and constraints. Because of the nature of the natural, genetic algorithms can be used to find solutions without regard to the subject matter specifically. Constraints function optimization problems with integer coding, can be solved by genetic algorithms. So as to produce the value of reliability function, cost and weight of a system.

From the results of the study revealed that the application of Genetic algorithm to constraints function optimization problems with integer coding, has a weakness that is probabilistic, because it is always associated with random numbers. A large population size does not ensure better fitness value, and the greater the number of generations, would produce a high fitness value. The best solution point is reached at generation 200 and generation 500, then approached a constant.

Key words : Genetic Algorithms, function optimization, integer coding.

1. PENDAHULUAN

Optimasi fungsi berkaitan dengan masalah pencarian solusi atas suatu himpunan melalui proses analisis dengan beberapa kendala dan kombinasi kendala sesuai tujuan optimasi. Dalam banyak kasus tidak mudah menyelesaikan masalah optimasi dengan fungsi tujuan non linier secara eksak. Pada pencarian solusi suatu masalah kadang-kadang dibutuhkan formulasi matematika yang kompleks utuk memberikan solusi yang pasti. Solusi optimum membutuhkan proses perhitungan yang panjang dan tidak praktis. Untuk memecahkan kasus ini membutuhkan metode heuristik .

Salah satunya yaitu algoritma genetik untuk memecahkan solusi optimal, dengan proses pencarian sejumlah titik optimal berdasar fungsi probabilistik. Jika ada satu kriteria yang dipertimbangkan menjadi masalah optimasi tunggal tujuan, dan jika lebih menjadi masalah optimasi multi tujuan. Masalah multi tujuan muncul dalam desain, pemodelan

dan perencanaan dari beberapa sistem yang kompleks dalam bidang industri, produksi, transportasi, pengaturan modal, dan lain-lain. Hampir setiap masalah keputusan di dunia nyata berkaitan dengan multi tujuan dan konflik tujuan, memerlukan penanggulangan terhadap berbagai hambatan dengan mencari solusi yang memenuhi setiap kendala.

Algoritma genetik tidak banyak memerlukan konsep matematika, dan dapat memperlakukan semua bentuk fungsi tujuan dan kendala (Gen dan Cheng:2000). Karena sifat alamiahnya, algoritma genetik dapat digunakan untuk mencari solusi tanpa memperhatikan pokok masalah secara khusus. Masalah optimasi fungsi berkendala dengan pengkodean bilangan bulat, dapat diselesaikan dengan algoritma genetik. Sehingga dapat dihasilkan nilai fungsi reliabilitas, biaya dan bobot dari suatu sistem. Contoh persoalan riil adalah desain komunikasi jaringan untuk menentukan penempatan komponen terbaik, dengan biaya minimum yang memenuhi kriteria kinerja sistem, seperti keterlambatan transmisi dan reliabilitas (Gen dan Cheng:2000).

1.1. Rumusan Masalah

Berdasarkan latar belakang di atas, maka akan digunakan pendekatan algoritma genetik untuk menyelesaikan masalah optimasi fungsi berkendala dengan pengkodean bilangan bulat.

1.2. Batasan Masalah

Masalah yang akan diselesaikan dalam penelitian ini dibatasi pada Algoritma genetik menggunakan pengkodean bilangan bulat, operator persilangan Aritmatika, mutasi seragam, jumlah generasi maksimum 1000 dan ukuran populasi 20.

1.3. Tujuan dan Manfaat Penelitian

Penelitian ini bertujuan untuk menyelesaikan masalah optimasi fungsi berkendala dengan pendekatan algoritma genetik, di mana selama ini penyelesaiannya masih manual dan cukup rumit. Persoalan riil yang dapat diselesaikan adalah desain komunikasi jaringan untuk menentukan penempatan komponen terbaik dengan biaya minimum, kelambatan transmisi, distribusi listrik dan pemipaan.

2. LANDASAN TEORI

2.1. Konsep Algoritma Genetik

Algoritma genetik (GA) merupakan suatu wujud pencarian random yang menirukan prinsip proses evolusi biologi alami guna mencari solusi optimal. Untuk suatu permasalahan kompleks, algoritma ini dimulai dengan suatu kumpulan parameter yang disebut kromosom atau string, kemudian masing-masing dievaluasi tingkat ketangguhannya oleh fungsi tujuan yang telah ditentukan.

Sifat AG adalah mencari kemungkinan-kemungkinan dari calon solusi untuk mendapatkan yang optimal bagi penyelesaian masalah. Ruang cakupan dari semua solusi yang layak (*feasible*) yaitu obyek-obyek diantara solusi yang sesuai dinamakan ruang pencarian (*search space*). Tiap titik dalam ruang pencarian mempresentasikan satu solusi yang layak. Tiap solusi yang layak ditandai dengan nilai *fitness*

Melalui proses seleksi alam atas operator genetik, gen-gen dari dua kromosom (*parent*) diharapkan akan menghasilkan kromosom baru dengan tingkat *fitness* yang lebih tinggi sebagai generasi baru atau keturunan (*offspring*) berikutnya. Kromosom-kromosom tersebut akan mengalami iterasi yang disebut generasi. Kromosom yang tangguh memiliki kecenderungan untuk menghasilkan keturunan yang berkualitas. Secara umum algoritma genetik terdiri dari 3 tahap, yaitu menentukan populasi awal secara acak, menghitung nilai

fitness masing-masing kromosom, menerapkan operasi genetik untuk mendapatkan alternatif solusi yang baru.

Goldberg mengemukakan bahwa AG mempunyai karakteristik-karakteristik yang perlu diketahui, sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

1. AG bekerja dengan pengkodean dari set solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
2. AG melakukan pencarian pada sebuah populasi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.
3. AG merupakan informasi fungsi obyektif (*fitness*), sebagai cara untuk mengevaluasi individu-individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
4. AG menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Variabel dan parameter yang digunakan pada AG adalah:

1. Fungsi *fitness* (fungsi tujuan) yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin dicapai.
2. Populasi jumlah individu yang dilibatkan pada setiap generasi.
3. Probabilitas terjadinya persilangan (*crossover*) pada suatu generasi.
4. Probabilitas terjadinya mutasi pada setiap individu.
5. Jumlah generasi yang akan dibentuk yang menentukan lama penerapan AG.

Langkah-langkah Algoritma Genetika, adalah:

1. Membangkitkan populasi awal

Populasi awal dibangkitkan secara random sehingga didapatkan solusi awal, terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Digunakan operator reproduksi /seleksi, *crossover* dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru, dimana generasi baru ini merupakan representasi dari solusi baru. Generasi baru dikenal dengan anak (*offspring*)

3. Evaluasi solusi

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasi sampai terpenuhi kriteria berhenti.

Sebelum AG dilakukan, ada dua hal penting yang harus dilakukan yaitu pendefinisian kromosom, merupakan solusi berbentuk simbol, dan fungsi *fitness* atau fungsi obyektif.

PENGGODEAN

Adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi untuk masalah dalam kromosom sebagai kunci pokok persoalan AG. Ada beberapa pengkodean yaitu pengkodean biner, bilangan riil, bilangan bulat dan struktur data.

- Pengkodean biner, memberikan banyak kemungkinan untuk kromosom walaupun dengan jumlah nilai-nilai yang mungkin terjadi pada suatu gen.
- Pengkodean bilangan riil, suatu pengkodean bilangan dalam bentuk riil. Masalah optimasi lebih tepat dengan pengkodean ini.
- Pengkodean bilangan bulat, sesuai untuk masalah optimasi kombinasi
- Pengkodean struktur data, menggunakan model struktur data. Biasanya untuk perancangan robot.

OPERATOR GENETIKA

GA merupakan proses pencarian yang heuristik dan acak, sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan GA dalam menemukan solusi optimum suatu masalah yang diberikan. Operator seleksi, *crossover* dan mutasi.

SELEKSI

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Langkah pertama dalam seleksi adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyek itu sendiri. Nilai *fitness* akan digunakan pada tahap berikutnya.

CROSSOVER

Crossover atau perkawinan silang bertujuan menambah keaneka ragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya.

- a. *Crossover* 1 titik
- b. *Crossover* 2 titik
- c. *Crossover* seragam

MUTASI

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* bertujuan memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik.

- a. Mutasi dalam pengkodean biner
- b. Mutasi dalam pengkodean permutasi
- c. Mutasi dalam pengkodean nilai
- d. Mutasi dalam pengkodean pohon

Kontrol Parameter GA yaitu sebagai berikut:

- a. Probabilitas *Crossover*

Menyatakan seberapa sering proses *crossover* akan terjadi antara dua kromosom orang tua.

- b. Probabilitas Mutasi

Menyatakan seberapa sering bagian-bagian kromosom akan dimutasikan .

2.3. Optimasi Pada Pengkodean Bilangan Bulat

Pengkodean bilangan bulat adalah metode mengkodekan bilangan dalam bentuk bilangan bulat, pengkodean ini cocok untuk masalah optimasi. Bentuk formulasi dari optimasi fungsi berkendala dengan pengkodean bilangan bulat sebagai berikut:

$$\text{Maks} \quad \sum_{k=1}^{q_2} w_k \lambda_k \quad (2.1)$$

$$\text{Kendala} \quad \lambda_k = \mu_k(f_k(x^i, x^f)), \quad k = 1, 2, \dots, q_2 \quad (2.2)$$

$$g_c(x^i, x^f) \leq G_c, \quad c = 1, 2, \dots, m \quad (2.3)$$

$$0 \leq \lambda_k \leq 1, \quad k = 1, 2, \dots, q_2 \quad (2.4)$$

3. RANCANGAN KOMPUTASI

3.1. Preparasi Algoritma Genetik

3.1.1. Representasi dan Inisialisasi Populasi Awal

Sebelum mengoperasikan algoritma genetik pertama-tama harus menyediakan populasi awal dengan proses acak. Yaitu menggunakan distribusi seragam acak untuk memperoleh bilangan riil $[0,1]$. Nilai acak dikodekan sesuai domain peubahnya yaitu bilangan bulat. Misalkan populasi dinyatakan dengan kromosom-kromosom dalam matriks sebagai berikut:

$$\begin{pmatrix} (x_{11}^i, x_{11}^r) & (x_{12}^i, x_{12}^r) & (x_{13}^i, x_{13}^r) & (x_{14}^i, x_{14}^r) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ (x_{p1}^i, x_{p1}^r) & (x_{p2}^i, x_{p2}^r) & (x_{p3}^i, x_{p3}^r) & (x_{p4}^i, x_{p4}^r) \end{pmatrix}$$

Di mana p adalah ukuran populasi, x_{pj}^i adalah jumlah komponen redundansi dan x_{pj}^r level reliabilitas komponen. Secara vektor dinyatakan:

$$V_p = [(x_{11}^i, x_{11}^r), (x_{12}^i, x_{12}^r), (x_{13}^i, x_{13}^r), (x_{14}^i, x_{14}^r)], p=1, \dots, \quad (3.1)$$

3.1.2. Menghitung Fungsi *Fitness*

Fungsi ini bertujuan dalam menentukan kelayakan kromosom untuk dipelihara atau tidak. Nilai *fitness* yang tinggi sebagai tolok ukur optimalitas solusi terhadap fungsi tujuan yang akan ditentukan. Evaluasi *fitness* dirumuskan sebagai:

$$\text{Eval}(V_p) = w_1 \lambda_1 + w_2 \lambda_2 + w_3 \lambda_3, p=1, 2, \dots, \text{ukuran}_p \quad (3.2)$$

Kromosom terbaik V^* diseleksi dengan persamaan sebagai berikut:

$$V^* = [\text{eval}(V_p), p=1, 2, \dots, \text{ukuran}_p] \quad (3.3)$$

3.1.3. Seleksi

Seleksi bertujuan memilah kromosom dalam populasi, sehingga dipilih kromosom yang memiliki peluang lebih besar untuk bertahan hidup dan berkembang biak. Seleksi menggunakan *roulette wheel*.

3.1.4. Persilangan

Fungsi ini bertujuan untuk mempersilangkan dua buah kromosom, sehingga menghasilkan kromosom baru yang membawa sifat berbeda. Proses diulang beberapa kali dalam populasi. Kromosom yang akan disilangkan ditentukan secara acak, digunakan persilangan Aritmatika. Nilai parameter crossover yaitu antara 0.6 - 1.0.

Operator aritmatika, konsep dasar operator jenis ini diambil dari convex set theory. Operator kawin silang (*cross-over*) aritmatika didefinisikan sebagai kombinasi dari dua kromosom yang menggunakan persamaan di bawah ini.

$$x_1' = \lambda_1 \cdot x_1 + \lambda_2 \cdot x_2 \quad (3.4)$$

$$x_2' = \lambda_1 \cdot x_2 + \lambda_2 \cdot x_1 \quad (3.5)$$

$$\lambda_1 + \lambda_2 = 1 \quad (3.6)$$

$$\lambda_1 \geq 0 \text{ dan } \lambda_2 \geq 0 \quad (3.7)$$

3.1.5. Mutasi

Mutasi berfungsi untuk mengubah secara acak variabel-variabel yang ada dalam kromosom, sehingga memunculkan variasi kandidat solusi. Mutasi yang digunakan adalah mutasi seragam. Nilai parameter mutasi yaitu kurang dari 0.1.

3.2. Pelaksanaan Penelitian

- Menentukan parameter yang digunakan dalam program genetik, ukuran populasi, generasi maksimum, peluang *crossover*, peluang mutasi dan faktor bobot.
- Menjalankan program dengan variasi ukuran populasi, generasi maksimum, persilangan dan mutasi, menggunakan software Matlab versi 5.3.
- Analisis pengaruh langkah 1 dan 2 terhadap nilai *fitness* dan nilai fungsi tujuan.

1. Hasil Dan Pembahasan

Dalam penelitian ini program dijalankan sebanyak sepuluh kali untuk memperoleh solusi optimal dari setiap model algoritma genetik. Solusi terbaik diperoleh berdasarkan nilai *fitness* tertinggi yang berguna untuk menentukan fungsi tujuan:

4.1. Penerapan Algoritma Genetik

Contoh kasus, dari Gen, Ida dan Kim (1998) mengusulkan masalah optimasi fungsi berkendala, sebagai berikut:

$$\text{Maks } w_1\lambda_1 + w_2\lambda_2 + w_3\lambda_3 \tag{4.1}$$

$$\text{Kendala } \lambda_1 = \mu_1 (f_1(x^i, x^r)) \tag{4.2}$$

$$\lambda_2 = \mu_2 (f_2(x^i, x^r)) \tag{4.3}$$

$$\lambda_3 = \mu_3 (f_3(x^i)) \tag{4.4}$$

$$g_1(x^i) = \sum_{j=1}^4 v_j (x^i_j) \leq V \tag{4.5}$$

$$0 \leq \lambda_k \leq 1, k = 1, 2, 3 \tag{4.6}$$

di mana w_k , adalah faktor bobot, x^i , x^r vektor bilangan bulat dan riil berdimensi, v_j adalah hasil bobot dan volume per elemen.

Perhitungan atau penyelesaian algoritma genetik dengan pengkodean bilangan bulat, variasi terhadap operator persilangan aritmatika, mutasi seragam dan penskalaan normal.

Dengan cara menentukan nilai-nilai parameter yaitu ukuran populasi= 20, $p_c=0.4$, $p_m=0.1$, maksimum generasi=1000, bobot tujuan $w_1=0.5$, $w_2=0.25$, $w_3=0.25$.

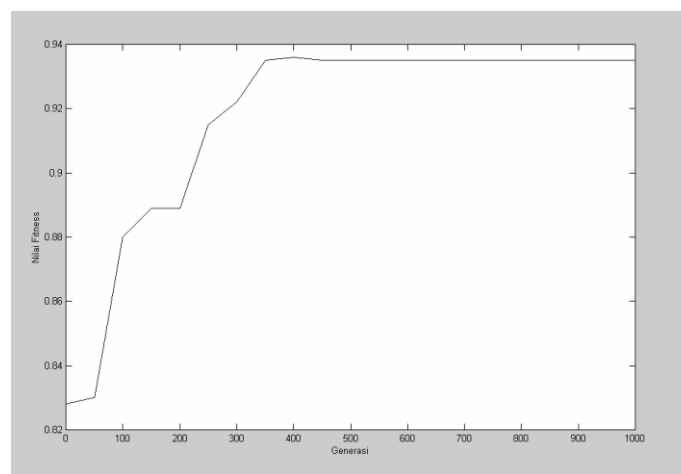
Program dijalankan sebanyak sepuluh kali, diperoleh hasil pada tabel 5.1, tabel 5.2, tabel 5.3 dan grafik 5.1, sebagai berikut:

Tabel 5.1. Proses Perhitungan Algoritma Genetik

Perlakuan	<i>Fitness</i>	Titik Solusi	f_1	f_2	f_3
1	0.936326	[(3, 0.834426) (3, 0.791546) (2, 0.921685) (3, 0.784708)]	0.970611	99.979442	147.048541
2	0.936999	[(3, 0.834733) (3, 0.797024) (2, 0.924963) (3, 0.794874)]	0.973131	105.241099	147.048541

3	0.936559	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.971364	101.528840	147.048541
4	0.936908	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.973818	106.999560	147.048541
5	0.936919	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.972932	104.858878	147.048541
6	0.936907	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.973329	105.826911	147.048541
7	0.936706	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.972081	103.073953	147.048541
8	0.935623	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.9725161	105.416934	147.048541
9	0.936108	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.970501	99.805513	147.048541
10	0.936983	[(3, 0.831190) (3, 0.797984 (2, 0.926130 (3, 0.797498)]	0.973237	105.514656	147.048541

Dari tabel 5.1 terlihat bahwa nilai f_3 tetap tidak mengalami perubahan, karena nilai variabel x^i adalah tetap. Perbandingan nilai *fitness* terhadap beberapa ukuran populasi dengan jumlah generasi 1000, memperlihatkan bahwa semakin besar ukuran populasi ternyata nilai *fitness* semakin menurun, akibatnya titik solusi dan nilai fungsi tujuan semakin mengecil. Berikut disajikan grafik hubungan antara generasi dengan *fitness* dari proses genetik, pada gambar 5.1 sebagai berikut:



Gambar.5.1. Hubungan Generasi dengan *Fitness*

Dari grafik di atas memperlihatkan bahwa semakin besar jumlah generasi, ternyata nilai *fitness* semakin meningkat, tetapi pada jumlah generasi lebih dari 200 nilai *fitness* mendekati konstan atau stabil.

Tabel 5.2. Hubungan Ukuran Populasi dengan Nilai *Fitness*

Ukuran_p	<i>Fitness</i>	Titik Solusi	f_1	f_2	f_3	Waktu (detik)
10	0.937060	[(3, 0.831190) (3, 0.797984) (2, 0.926130) (3, 0.797498)]	0.973448	105.929401	147.048541	29.5560
20	0.936999	[(3, 0.834733) (3, 0.797024) (2, 0.924963) (3, 0.794874)]	0.973131	105.241099	147.048541	45.4900
30	0.936783	[(3, 0.837360) (3, 0.794613) (2, 0.923251) (3, 0.791456)]	0.973448	103.644134	147.048541	60.7700

Pada tabel 5.2, nilai *fitness* tertinggi tidak menjamin nilai solusi titik yang lebih baik. Untuk jumlah generasi 1000 dengan ukuran populasi 10,20, dan 30 ternyata hasilnya memberikan nilai *fitness* yang menurun akibatnya nilai fungsi tujuan menurun. Rata-rata waktu eksekusi CPU yang digunakan meningkat.

Tabel 5.3. Hubungan Jumlah Generasi dengan Nilai *Fitness*

Gen_maks	<i>Fitness</i>	Titik Solusi	f_1	f_2	f_3	Waktu (detik)
200	0.936490	[(3, 0.843569) (3, 0.803651) (2, 0.927953) (3, 0.804731)]	0.976177	113.162553	147.048541	10.1186
500	0.936676	[(3, 0.830182) (3, 0.788941) (2, 0.929701) (3, 0.792138)]	0.972066	103.073342	147.048541	22.6575
1000	0.936999	[(3, 0.834733) (3, 0.797024) (2, 0.924963) (3, 0.794874)]	0.973131	105.241099	147.048541	45.4900

Pada tabel 5.3, memperlihatkan bahwa semakin banyak jumlah generasi ternyata nilai *fitness* semakin meningkat, titik solusi terbaik terjadi pada jumlah generasi 200 dan generasi 500. Rata-rata waktu eksekusi CPU yang digunakan meningkat.

5. Kesimpulan

1. Banyaknya ukuran populasi tidak menjamin nilai *fitness* lebih baik.
2. Semakin besar jumlah generasi, akan menghasilkan nilai *fitness* yang tinggi.
3. Penerapan Algoritma Genetik untuk penyelesaian masalah optimasi, memiliki kelemahan yang bersifat probabilistik, karena selalu berhubungan dengan bilangan random.

Daftar Pustaka

- [Ami05] Aminuddin.(2005).*Prinsip-prinsip Riset Operasi*. Erlangga. Surabaya.
- [Dav96] Davis, Lawrence.(1996).*Handbook of Genetic Algorithm*, Van Nonstrad Reinhold, New York.
- [Gen97] Gen, Mitsuo, Runwei Cheng (1997).*Genetic Algorithm and Engineering Design*. John Wiley&Sons, Inc, New York.
- [Gol89] Gold Berg, David E.(1989).*Genetic Algorithm and Search, optimization, and Machine Learning*. Addison_Wesley Publishing Company, Inc, Massachusetts.
- [Mic96] Michalewics, Z.(1996).*Genetic Algorithm+Data Structures =Evolution Programs*.Springer-Verlag.Berlin. Heidelberg, New York.
- [Sri08] Sri Kusumadewi.(2008).*Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*. Graha Ilmu. Yogyakarta.

