Jurnal Penelitian Sains ; hal 69 - 78
No.13. April 2003 ISSN: 1410-7058

FUZZY NONLINEAR MIX-INTEGER GOAL PROGRAMMING WITH GENETIC ALGORITHMS

Samsuryadi
Jurusan Matematika FMIPA Universitas Sriwijaya

ABSTRACT

System's reliability optimization problems are modeled using fuzzy nonlinear mix-integer goal programming problems, involving imprecise nonlinear mix-integer information. Furthermore, fuzzy nonlinear mix-integer goal programming is transformed into nonlinear mix-integer programming problem and the problem is solved using genetic algorithms by means of Matlab 5.3 software. The results of genetic algorithms with operator arithmetic crossover are the large of initial population number does not give the better fitness and the more generation numbers will result more high fitness.

Key words: genetic algorithm, arithmetic crossover, goal programming.

1. PENDAHULUAN

Pada akhir-akhir ini, algoritma genetik telah menjadi pertimbangan sebagai teknik potensial atau pendekatan baru untuk menyelesaikan masalah optimasi. Optimasi berkaitan dengan masalah pencarian solusi atas suatu himpunan pilihan yang mungkin untuk mengoptimumkan kriteria tertentu. Jika ada satu kriteria yang dipertimbangkan menjadi masalah optimasi satu tujuan dan jika lebih menjadi masalah optimasi multi tujuan (Dev: 1995).

Algoritma genetik dapat memperlakukan setiap jenis fungsi tujuan dan kendala (linier dan nonlinier) terdefinisi diskrit, kontinu atau ruang pencarian campuran (Gen dan Cheng: 1997). Karena sifat alamiahnya itu, algoritma genetik dapat digunakan untuk mencari solusi tanpa memperhatikan pokok masalah secara khusus.

Artikel ini adalah hasil dari tesis atau penelitian yang dilakukan oleh Syamsuryadi (2002) yang membahas model program *goal* bilangan bulat campuran nonlinier *fuzzy* dengan pendekatan algoritma genetik.

2. METODE PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini adalah:

1. Merumuskan algoritma genetik.
2. Menggunakan langkah 1 untuk menyelesaikan salah satu masalah optimasi reliabilitas sistem dengan *software* Matlab 5.3.

3. Model Program Goal Bilangan Bulat Campuran Nonlinier Fuzzy

Bentuk formulasi dari program *goal* bilangan bulat campuran nonlinier *fuzzy* didefinisikan sebagai berikut (Gen dan Cheng: 2000):

$$f_k(x^i, x^r) \lesssim b_k \quad k = 1,2,\ldots,q_0 \qquad (3.1a)$$

$$f_k(x^i, x^r) \gtrsim b_k \quad k = q_0 + 1,\ldots,q_1 \qquad (3.1b)$$

$$f_k(x^i, x^r) \cong b_k \quad k = q_1 + 1,\ldots,q_2 \qquad (3.1c)$$

$$\text{kendala} \quad g_c(x^i, x^r) \le G_c, \quad c = 1,2,\ldots,m \qquad (3.1d)$$

dengan x^i adalah vektor bilangan bulat berdimensi n; x^r adalah vektor bilangan riil berdimensi n; simbol $\lesssim$ (*fuzzy-min*), menyatakan pendekatan kurang dari atau sama dengan *level* aspirasi b_k; simbol $\gtrsim$ (*fuzzy-max*), menyatakan pendekatan lebih besar atau sama dengan *level* aspirasi b_k; simbol $\cong$ (*fuzzy-equal*), artinya bahwa $f_k(x^i, x^r)$ sekitar aspirasi b_k; $f_k(x^i, x^r)$ adalah kendala *goal* nonlinier ke-k, $g_c(x^i, x^r)$ adalah kendala sistem nonlinier ke-c; b_k adalah berdasarkan nilai target terhadap *goal* k; G_c adalah sumber tersedia dari c kendala sistem; q_0 adalah jumlah kendala *goal fuzzy-min*; $q_1 - q_0$ adalah jumlah kendala *fuzzy-max*; $q_2 - q_1$ adalah jumlah kendala *fuzzy-equal*; dan m adalah jumlah kendala sistem.

Fungsi keanggotaan dari goal *fuzzy* dinyatakan sebagai berikut:

Untuk kasus tanda lebih kecil *fuzzy*:

$$\mu_k(f_k(x^i,x^r)) = \begin{cases} 1 & ; f_k(x^i,x^r) < b_k \\ 1 - \dfrac{f_k(x^i,x^r) - b_k}{t_k^R} & ; b_k \le f_k(x^i,x^r) \le b_k + t_k^R \\ 0 & ; f_k(x^i,x^r) > b_k + t_k^R \end{cases} \qquad (3.2a)$$

Untuk kasus tanda sama dengan *fuzzy*:

$$\mu_k(f_k(x^i,x^r)\ ) = \begin{cases} 0, & ;\ f_k(x^i,x^r) < b_k - t_k^L \\ \dfrac{f_k(x^i,x^r)-(b_k-t_k^L)}{t_k^L} & ;\ b_k - t_k^L\, t \le f_k(x^i,x^r) \le b_k \\ 1 & ;\ f_k(x^i,x^r) = b_k \\ 1 - \dfrac{f_k(x^i,x^r)-b_k}{t_k^R} & ;\ b_k \le f_k(x^i,x^r) \le b_k + t_k^R \\ 0 & ;\ f_k(x^i,x^r) > b_k + t_k^R \end{cases} \qquad (3.2b)$$

Untuk kasus tanda lebih besar *fuzzy*:

$$\mu_k(f_k(x^i,x^r)) = \begin{cases} 0 & ;\ f_k(x^i,x^r) < b_k - t_k^L \\ \dfrac{f_k(x^i,x^r)-b_k}{t_k^R} & ;\ b_k - t_k^L \le f_k(x^i,x^r) \le b_k \\ 1 & ;\ f_k(x^i,x^r) > b_k \end{cases} \qquad (3.2c)$$

Selanjutnya (3.1) ditransformasi ke dalam masalah program bilangan bulat campuran nonlinier:

$$\text{Maks} \quad \sum_{k=1}^{q_2} w_k \lambda_k \qquad (3.3a)$$

$$\text{Kendala} \quad \lambda_k = \mu_k(f_k(x^i,x^r)),\ k = 1,2,\ldots,q_2 \qquad (3.3b)$$

$$g_c(x^i, x^r) \le G_c,\quad c = 1,2,\ldots,m \qquad (3.3c)$$

$$0 \le \lambda_k \le 1,\quad k = 1,2,\ldots,q_2 \qquad (3.3d)$$

dimana w_k adalah faktor bobot yang diberikan oleh pembuat keputusan.

4. HASIL DAN PEMBAHASAN

4.1 Pendekatan Algoritma Genetik

Gen, dkk (1998) dan Syamsuryadi (2002) mengusulkan suatu algoritma genetik untuk menyelesaikan masalah optimasi reliabilitas sistem dinyatakan dengan model program *goal* bilangan bulat campuran nonlinier *fuzzy* (f-nMIGP).

Representasi dan Inisialisasi. Misalkan V menyatakan kromosom-kromosom dalam suatu populasi. Kromosom-kromosom dinyatakan sebagai berikut:

$$V = [(x_1^i, x_1^r)\ (x_2^i, x_2^r)\ \cdots\ (x_n^i, x_n^r)] \quad (4.1)$$

Representasi kromosom ini cocok digunakan untuk f-nMIGP dengan vektor keputusan bilangan bulat campuran berdimensi n. Definisikan ukuran_p sebagai jumlah kromosom. Kromosom-kromosom awal dihasilkan secara acak dalam domainnya.

Evaluasi. Fungsi evaluasi didefinisikan dengan mempertimbangkan faktor bobot sebagai berikut.

$$eval(V_p) = \sum_{k=1}^{q_2} w_k \lambda_k, \quad p = 1,2, \ldots, \text{ukuran\_p} \quad (4.2)$$

Dengan λ diperoleh dari fungsi keanggotaan *fuzzy* dari (3.2) dan w adalah faktor bobot berdasarkan pada *goal* (atau tujuan) yang dianggap penting dan ditentukan oleh pembuat keputusan. Kromosom terbaik V* diseleksi dengan persamaan di bawah ini.

$V^* = \{\text{eval}(V_p), p = 1, 2, \ldots, \text{ukuran\_p}\}$ (4.3)

Seleksi. Metode seleksi menggunakan gabungan seleksi roda rolet dan pendekatan *elitist*. Seleksi roda rolet adalah suatu metode untuk menghasilkan suatu generasi baru secara proporsional terhadap *fitness* (tujuan) dari setiap individu, dan seleksi *elitist* digunakan agar kromosom terbaik tidak hilang pada saat seleksi dan mengatasi kesalahan sampling stokastik. Prosedur seleksi digambarkan sebagai berikut.

Langkah 1. Hitung peluang komulatif Q_p dari setiap kromosom V_p (p = 1,2, ..., ukuran_p).

Langkah 2. Bangkitkan bilangan riil acak r dalam [0,1].

Langkah 3. Jika $r \leq Q_1$, pilih kromosom pertama, V_1: selainnya, jika $Q_{p-1} < r \leq Q_p$, maka pilih kromosom ke-p, V_p ($2 \leq p \leq$ ukuran_p).

Langkah 4. Ulangi langkah 2 dan 3 sampai diperoleh sebanyak ukuran_p kromosom.

Langkah 5. Jika kromosom terbaik tidak terpilih dalam generasi berikutnya, ganti satu kromosom secara acak dari populasi baru dengan satu kromosom terbaik generasi sebelumnya.

Persilangan. Persilangan aritmetika digunakan, yang didefinisikan sebagai suatu kombinasi konveks dari dua kromosom (Syamsuryadi: 2002). Jika himpunan kendala konveks, operasi persilangan aritmetika menjamin bahwa jika kedua orang tua fisibel, maka kedua anaknya fisibel juga. Bentuk persilangan aritmetika sebagai berikut.

$V_1^t = \lambda \cdot V_1 + (1-\lambda) \cdot V_2$ dan $V_2^t = (1-\lambda) \cdot V_1 + \lambda \cdot V_2$

dimana V menyatakan orang tua, V^t menyatakan anak, dan λ suatu bilangan acak dalam [0,1].

Mutasi. Parameter peluang mutasi seragam p_m mendefinisikan bahwa jumlah nilai harapan p_m . ukuran_p kromosom akan mengalami operasi mutasi. Bangkitkan bilangan acak r dalam [0,1], pilih kromosom untuk mutasi jika $r < p_m$. Untuk setiap orang tua V jika elemen x_j secara acak dipilih untuk mutasi, hasil anak adalah $V^t = [x_1 \ldots x_j^t \ldots x_n]$ dengan x_j^t adalah suatu nilai acak (distribusi peluang seragam) dari $[x_j^L, x_j^U]$. Jika x_j^t adalah bilangan bulat, maka mutasi mengembalikan ke bilangan acak bulat.

Secara umum rancangan algoritma genetik untuk masalah program *goal* bilangan bulat campuran nonlinier *fuzzy* adalah (Syamsuryadi: 2002):

Langkah 1. Mengeset parameter-parameter yang digunakan.

Langkah 2. Inisialisasi. Bangkitkan secara acak populasi awal.

Langkah 3. Membentuk persilangan. Bangkitkan bilangan acak r dalam [0,1]. Jika $r < p_c$ maka pilih kromosom untuk persilangan. Ulangi operasi ini sebanyak ukuran populasi. Untuk setiap pasang orang tua (V_1, V_2), dipersilangankan akan menghasilkan dua anak (V_1^t, V_2^t):

$V_1^t = \alpha \cdot V_1 + (1-\alpha) \cdot V_2$ dan $V_2^t = (1-\alpha) \cdot V_1 + \alpha \cdot V_2$

dimana α adalah suatu bilangan acak dalam [0,1].

Langkah 4. Membentuk Mutasi seragam. Bangkitkan bilangan acak r dalam [0,1]. Jika $r < p_m$, maka pilih kromosom V. Ulangi operasi ini sebanyak ukuran populasi. Secara acak pilih elemen x_j dari orang tua yang terpilih.

Langkah 5. Hitung nilai *fitness*. Hitung eval(V_p), p = 1,2, ..., ukuran_p untuk setiap kromosom (vektor peubah) dan tentukan V* terbaik.

Langkah 6. Memilih kromosom ukuran_p untuk generasi berikutnya berdasarkan metode seleksi roda rolet dan *elitist*.

Langkah 7. Lakukan pengecekan iterasi. Jika t < maks_gen, ke langkah 3 untuk generasi selanjutnya. Jika t = maks_gen, hentikan interasi.

4.2 Penerapan Algoritma Genetik

Program AG dibuat dengan *software* Matlab 5.3, selanjutnya diterapkan untuk menyelesaikan masalah desain reliabilitas sistem dengan *goal fuzzy* (Gen, dkk: 1998; Syamsuryadi: 2002). Masalah ini merupakan suatu variasi dari masalah alokasi reliabilitas optimal oleh Dhingra (1992) yang dirumuskan sebagai berikut:

Maks

$$f_1(x^i, x^r) = \prod_{j=1}^{4} \{1 - \exp(x_j^i \ln(1 - x_j^r))\} \gtrsim b_1 \quad (4.1)$$

Min

$$f_2(x^i, x^r) = \sum_{j=1}^{4} C(x_j^r) \left[x_j^r + \exp\left(\frac{x_j^r}{4}\right)\right] \lesssim b_2 \quad (4.2)$$

$$\text{Min } f_3(x^i) = \sum_{j=1}^{4} d_j x_j^i \exp\left(\frac{x_j^i}{4}\right) \lesssim b_3 \quad (4.3)$$

$$\text{Kendala } g_1(x^i) = \sum_{j=1}^{4} v_j(x_j^i) \le V \quad (4.4)$$

$1 \le x_j^i \le 10$, bilangan bulat positif, j = 1, ..., 4 (4.5)

$0.5 \le x_j^r \le 1\text{-}10^{-6}$, bilangan riil, j = 1, ..., 4 (4.6)

dengan x_j^i adalah banyak komponen redundansi dari subsistem j, x_j^r adalah reliabilitas komponen dari subsistem j, v_j adalah hasil bobot dan volume per elemen dari subsistem j, d_j adalah bobot setiap komponen dari subsistem j, dan $C(x_j^r)$ adalah harga setiap komponen dengan reliabilitas x_j^r dari subsistem j sebagai berikut:

$$C(x_j^r) = \alpha_j \left(\frac{-t_o}{\ln(x_j^r)} \right)^{\beta_j}, \quad j = 1, \ldots, 4$$

Dengan α_j dan β_j adalah representasi konstanta karakteristik fisik dari subsistem j, t_o adalah waktu beroperasi selama komponen tidak gagal, $F_1 = b_1 - t_1^L$, $F_2 = b_2 + t_2^R$, dan $F_3 = b_3 + t_3^R$. Simbol $\tilde{>}$ (*fuzzy-maks*) menandakan bahwa pembuat keputusan terpenuhi dengan mantap jika hasil lebih kecil dari tingkat aspirasi b_k yang memadai ke suatu batas toleransi t_k^L dan simbol $\tilde{<}$ (*fuzzy-min*) menandakan bahwa pembuat keputusan terpenuhi dengan mantap jika hasil lebih besar dari tingkat aspirasi b_k yang memadai ke suatu batas toleransi t_k^R.

Permasalahan di atas dapat ditransformasikan dalam bentuk masalah program bilangan bulat campuran nonlinier berikut ini:

Maks $w_1\lambda_1 + w_2\lambda_2 + w_3\lambda_3$ (4.7)

Kendala $\lambda_k = \mu_k(f_k(x^i, x^r))$ (4.8)

$$g_1(x^i) = \sum_{j=1}^{4} v_j(x_j^i) \le V \qquad (4.9)$$

$0 \le \lambda_k \le 1, k = 1, 2, 3$ (4.10)

dimana w_k, adalah faktor bobot diberikan oleh pembuat keputusan. Fungsi keanggotaan dari *goal fuzzy*, λ_k didefinisikan menggunakan Persamaan (3.2).

Koefisien-koefisien konstanta untuk masalah ini dinyatakan dalam Tabel 1. (Dhingra: 1992).

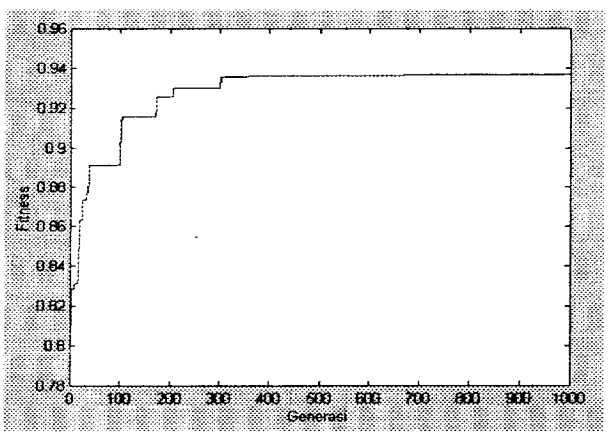
Tabel 1. Koefisien Konstanta untuk Desain Reliabilitas dengan *Goal Fuzzy*

Banyak Subsistem				4
Batas F_1				0.75
Batas F_2				400
Batas F_3				500
Batas V				250
Waktu operasi t_0				1000 jam
Subsistem	$10^5 . \alpha_j$	β_j	v_j	d_j
1	1.0	1.5	1	6
2	2.3	1.5	2	6
3	0.3	1.5	3	8
4	2.3	1.5	2	7

Langkah selanjutnya mengeset nilai-nilai parameter sebagai berikut: ukuran_p = 20, p_c = 0.4, p_m = 0.1, maks_gen = 1000, t_1^L = 0.25, t_2^R = 300, t_3^R = 360 dan bobot tujuan w_1 = 0.5,
w_2 = 0.25, dan w_3 = 0.25.

Setelah program dijalankan sebanyak sepuluh kali diperoleh nilai *fitness* tertinggi adalah 0.936999 dengan titik solusi [(3, 0.834733) (3, 0.797024) (2,0.924963) (3, 0.794874)] dan nilai fungsi tujuan $f_1(x^i, x^r)$ = 0.973131, $f_2(x^i, x^r)$ = 105.241099 dan $f_3(x^i)$ = 147.048541. Grafik hubungan generasi dengan *fitness* dari proses genetik ini disajikan pada Gambar 1. Rata-rata Waktu CPU dari keseluruhan proses adalah 45.4900 detik.

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-nilai fungsi tujuan semakin mengecil, tetapi rata-rata waktu CPU yang digunakan mengalami peningkatan (Tabel 2).



Gambar 1. Hubungan Generasi dengan *Fitness*

Sedangkan perbandingan nilai *fitness* terhadap beberapa jumlah generasi dengan ukuran populasi 20 dalam Tabel 3, memperlihatkan bahwa semakin banyak jumlah generasi ternyata nilai *fitness* semakin meningkat, rata-rata waktu CPU meningkat juga, tetapi titik solusi terbaik terjadi pada jumlah generasi 200 dan generasi 500 titik solusi terkecil.

5. KESIMPULAN

Banyaknya ukuran populasi awal tidak menjamin baiknya nilai *fitness*.

Jumlah generasi semakin besar akan menghasilkan nilai *fitness* yang lebih tinggi.

6. DAFTAR PUSTAKA

Davis, Lawrence. 1991. *Handbook of Genetic Algorithms*. Van Nonstrand Reinhold. New York.

Dev, K. 1995. *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall. New Delhi.

Dhingra, A.K. 1992. Optimal Apportionment of Reliability and Redundancy in a Series System under Multiple Objectives. *IEEE Transactions on Reliability*. Vol.41. No.4: 576-582.

Gen, Mitsuo and Runwei Cheng. 1997. *Genetic Algorithms and Engineering Design*. John Wiley & Sons, Inc. New York.

Gen, Mitsuo and Runwei Cheng. 2000. *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, Inc. New York.

Gen, M., K. Ida and J.R. Kim. 1998. System Reliability Optimization with Fuzzy Goals Using Genetic Algorithm. *Journal of the Japan Society of Fuzzy Theory and Systems*. Vol.10. No.2: 356-365.

Goldberg, David E. 1989. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley Publishing Company, Inc. Massachusetts.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Third, Revised and Extended Edition. Springer-Verlag Berlin Heidelberg. New York.

Oyama, Okira, Shigeru Obayashi, and Kazuhiro Nakahashi. 2000. *Real-Coded Adaptive Range and Its Application to Aerodynamic Design.*

home.earthlink.net/~akiraoyama/papers/jsme2000.pdf

Syamsuryadi. 2002. Penggunaan Algoritma Genetik pada Program Goal Bilangan Bulat Campuran Nonlinier Fuzzy untuk Menyelesaikan Masalah Optimasi Reliabilitas Sistem. *Tesis S2 Ilmu Komputer*, Universitas Gadjah mada.