

SISTEM PENDUKUNG KEPUTUSAN UNTUK MASALAH OPTIMISASI MULTIKRITERIA

Lucky E. Santoso

ABSTRACT

Multicriterion optimization problems arise in various areas of engineering, when engineers should make decisions in the situation where several, often conflicting, objectives must be satisfied optimally, while considering the scarcity of resources. A support system is required for such decisions. A system of computer software is developed based on a numerical multicriterion optimization method; the effectiveness of the system as a tool for solving multicriterion optimization problems is demonstrated using a case study.

Keywords: Decision support systems, multicriterion optimization.

ABSTRAK

Masalah optimisasi multikriteria muncul di berbagai bidang rekayasa, yaitu saat rekayasawan harus mengambil keputusan pada situasi di mana beberapa tujuan yang sering kali saling bertentangan serentak harus dipenuhi secara optimal, sambil mempertimbangkan terbatasnya sumber daya yang ada. Karenanya diperlukan suatu sistem untuk mendukung pengambilan keputusan tersebut. Telah dibangun suatu sistem perangkat lunak komputer berdasarkan metode numeris untuk optimisasi multikriteria; keefektifan sistem tersebut sebagai alat bantu dalam menyelesaikan masalah optimisasi multikriteria diperlihatkan oleh suatu studi kasus.

Kata kunci: Sistem pendukung keputusan, optimisasi multikriteria.

PENDAHULUAN

Perumusan Masalah

Masalah optimisasi multikriteria (*multicriterion optimization problems*) muncul di berbagai bidang rekayasa, yaitu saat rekayasawan harus mengambil keputusan pada situasi di mana beberapa tujuan—yang sering kali saling bertentangan—serentak harus dipenuhi secara optimal, sambil mempertimbangkan terbatasnya sumber daya yang ada. Karenanya diperlukan suatu sistem untuk mendukung pengambilan keputusan tersebut.

Diasumsikan bahwa suatu sistem pendukung keputusan (*decision support system*) akan cukup tepat untuk keperluan di atas. Tidak ada konsensus tentang definisi dari sistem pendukung keputusan, namun beberapa karakteristik umumnya dapat disebutkan di sini, yang mengacu pada Turban (1995):

1. Penggunaan model. Komunikasi antara pengambil keputusan dan sistem terjalin melalui model-model matematis, jadi pengambil keputusan bertanggung jawab membangun model matematis berdasarkan permasalahan yang dihadapinya.
2. Berbasis komputer. Sistem ini mempertemukan penilaian manusia (pengambil keputusan) dengan informasi komputer. Informasi komputer ini dapat berasal dari perangkat lunak komputer yang merupakan implementasi dari metode numeris untuk permasalahan matematis yang bersangkutan.
3. Fleksibel. Sistem harus dapat beradaptasi terhadap timbulnya perubahan pada permasalahan yang ada. Jadi pengambil keputusan harus dibolehkan untuk melakukan perubahan pada model yang telah diberikannya kepada sistem, ataupun memberikan model yang baru.
4. Interaktif dan mudah digunakan. Pengambil keputusan bertanggung jawab untuk menentukan apakah jawaban yang diberikan oleh sistem memuaskan atau tidak. Bagaimanapun juga sistem bertugas mendukung, bukan menggantikan pengambil keputusan. Jadi sistem harus memiliki kemampuan interaktif: pengambil keputusan harus diijinkan untuk menjelajahi alternatif jawaban dengan cara memvariasi parameter-parameter yang ada pada sistem.
5. Efektif, bukan efisien. Misalnya, akurasi lebih diutamakan daripada waktu komputasi.

Tujuan

Proyek ini bertujuan membangun sistem perangkat lunak komputer sebagai implementasi dari suatu metode numeris untuk menyelesaikan masalah optimisasi multikriteria yang model matematisnya diberikan oleh pengguna. Sesuai dengan karakteristik sistem pendukung keputusan di atas, sistem ini haruslah mudah digunakan, yaitu pengguna dapat memasukkan dan menyunting model masalah beserta parameter-parameter yang dikehendakinya secara mudah dan setiap saat bisa meminta jawaban dari sistem; selain itu sistem ini haruslah efektif, yaitu memiliki akurasi yang dapat diterima. Sebagai tambahan, sistem ini juga harus beroperasi pada *platform* yang populer, yaitu PC dengan sistem operasi Windows.

OPTIMISASI MULTIKRITERIA

Optimisasi Satu Kriteria

Model matematis umum untuk masalah optimisasi satu kriteria sering juga disebut sebagai program nonlinear, yaitu:

Diberikan fungsi-fungsi linear dan/atau nonlinear

$$f: \mathcal{R}^n \rightarrow \mathcal{R}, \mathbf{g}: \mathcal{R}^n \rightarrow \mathcal{R}^m, \mathbf{h}: \mathcal{R}^n \rightarrow \mathcal{R}^l$$

Tetapkanlah \mathbf{x}^*

agar $f(\mathbf{x}^*)$ memiliki nilai optimum

seraya tunduk pada $m+l$ kekangan

$$\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$$

$$\mathbf{h}(\mathbf{x}^*) \geq \mathbf{0}.$$

Di sini $\mathbf{x}=(x_1, x_2, \dots, x_n)^T$ akan disebut sebagai variabel keputusan (*decision variables*), $f(\mathbf{x})$ disebut fungsi sasaran (*objective function*), $\mathbf{g}(\mathbf{x}^*)=\mathbf{0}$ disebut kekangan kesamaan (*equality constraints*), dan $\mathbf{h}(\mathbf{x}^*)\geq\mathbf{0}$ disebut kekangan ketaksamaan (*inequality constraints*).

Di sini akan diambil kesepakatan bahwa yang dimaksud dengan nilai optimum adalah nilai minimum dari fungsi yang bersangkutan. Masalah mencari \mathbf{x}^* agar $f(\mathbf{x}^*)$ maksimum dapat diungkapkan sebagai masalah mencari \mathbf{x}^* agar $-f(\mathbf{x}^*)$ minimum, sebab $\max f(x) = -\min -f(x)$.

Optimisasi Multikriteria

Jika pada masalah optimisasi satu kriteria hanya ada satu fungsi sasaran yang terlibat, maka masalah optimisasi multikriteria meliputi lebih dari satu fungsi sasaran.

Jadi perlu diadakan sedikit perubahan pada model matematis di atas:

Diberikan fungsi-fungsi linear dan/atau nonlinear

$$\mathbf{f}: \mathcal{R}^n \rightarrow \mathcal{R}^k, \mathbf{g}: \mathcal{R}^n \rightarrow \mathcal{R}^m, \mathbf{h}: \mathcal{R}^n \rightarrow \mathcal{R}^l$$

Tetapkanlah \mathbf{x}^*

agar setiap komponen $\mathbf{f}(\mathbf{x}^*)$ memiliki nilai optimum

seraya tunduk pada $m + l$ kekangan

$$\mathbf{g}(\mathbf{x}^*)=\mathbf{0}$$

$$\mathbf{h}(\mathbf{x}^*)\geq\mathbf{0}.$$

Di sini akan muncul konflik kepentingan antar fungsi sasaran; karena jika $f_1(\mathbf{x}^{*(1)})$ memiliki nilai minimum, belum tentu $f_2(\mathbf{x}^{*(1)})$, $f_3(\mathbf{x}^{*(1)})$, ... $f_k(\mathbf{x}^{*(1)})$ juga memiliki nilai minimum. Konflik kepentingan di atas harus dikompromikan: nilai optimum suatu fungsi sasaran haruslah berupa nilai yang dapat diterima oleh keseluruhan fungsi sasaran.

METODE

Pada bagian ini dibicarakan metode-metode yang digunakan dalam proyek ini untuk menyelesaikan masalah optimisasi multikriteria yang telah didefinisikan di atas. Pilihan atas metode-metode ini didasarkan pada kemudahan implementasi.

Optimisasi Satu Kriteria Tanpa Kekangan: Metode Pencarian Langsung dan Acak

Salah satu metode numeris untuk mencari lokasi minimum dari fungsi real $f(\mathbf{x})$, $\mathbf{x} \in \mathcal{R}^n$, adalah metode pencarian langsung (*direct search*) yang diajukan oleh Hooke dan Jeeves. Metode ini memanfaatkan dua langkah, yaitu langkah penjelajahan (*exploratory moves*) untuk menentukan arah pencarian yang tepat, dan langkah pola (*pattern moves*) untuk mempercepat pencarian. Parameter-parameter yang digunakan dalam metode ini adalah:

$maxm$ \equiv jumlah maksimum langkah pola yang diijinkan.

fpk \equiv panjang langkah awal, dinyatakan sebagai fraksi dari jangkauan $\mathbf{xk} - \mathbf{xp}$, di mana $\mathbf{xk} \in \mathcal{R}^n$ adalah perkiraan batas atas dan $\mathbf{xp} \in \mathcal{R}^n$ adalah perkiraan batas bawah.

wk \equiv fraksi dari panjang langkah awal sebagai kriteria konvergensi.

Keterangan lebih lanjut mengenai metode Hooke dan Jeeves diberikan antara lain oleh Avriel (1976) dan Bronson (1982).

Pencarian langsung belum tentu menemukan minimum yang merupakan *global minimum*. Untuk memperbesar peluang itu, digunakan kombinasi berupa pencarian langsung dan acak (*direct and random search*), yang memanfaatkan parameter-parameter berikut:

$ntest$ \equiv jumlah lokasi acak dalam pencarian *shotgun*.

nh \equiv jumlah maksimum pencarian *shotgun* yang diijinkan.

$ntmc$ \equiv jumlah lokasi acak untuk memperoleh lokasi awal pencarian yang baru.

$nraz$ \equiv jumlah pencarian yang dilakukan menggunakan lokasi awal pencarian yang baru.

Metode pencarian langsung dan acak dapat dijelaskan melalui *pseudocode* berikut, diawali dengan mengambil suatu nilai $\mathbf{xac} \in \mathcal{R}^n$ sebagai lokasi awal pencarian:

```

kl=1;
do {
  kount=0;
  do{
    Lakukan pencarian langsung Hooke dan Jeeves;
    Hasilnya  $\mathbf{x}$ ;
    Lakukan pencarian shotgun: Dari sebanyak  $ntest$ 
      lokasi acak di sekitar  $\mathbf{x}$  cari yang paling optimal;
    Hasilnya  $\mathbf{xs}$ ;
    if ( $f(\mathbf{xs}) \geq f(\mathbf{x})$  ||  $kount \geq nh$ ) break;
    Gunakan  $\mathbf{xs}$  sebagai lokasi awal pencarian;
    kount++;
  } while(1);
  if ( $kl==1$  ||  $f(\mathbf{x}) < fex$ ){
    fex =  $f(\mathbf{x})$ ;
     $\mathbf{xex}=\mathbf{x}$ ;
  }
  kl++;
} while(1);

```

```

}
if (kl>nraz) break;
Dari sebanyak ntmc lokasi acak di ruang
 $K \equiv \{\mathbf{x} : \mathbf{x} \in \mathfrak{R}^n, \mathbf{x}_p < \mathbf{x} < \mathbf{x}_k\}$ , cari yang paling optimal.
Gunakan sebagai lokasi awal pencarian;
kl++;
} while(1);

```

Lokasi optimal akan diberikan oleh \mathbf{xex} .

Optimisasi Satu Kriteria Dengan Kekangan: Penggunaan Fungsi Hukuman

Meninjau masalah optimisasi satu kriteria dengan kekangan:

Tetapkanlah \mathbf{x}^*
 agar $f(\mathbf{x}^*)$ memiliki nilai optimum
 seraya tunduk pada kekangan
 $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$
 $\mathbf{h}(\mathbf{x}^*) \geq \mathbf{0}$

Dengan mendefinisikan suatu daerah layak (*feasible region*)

$$X = \{\mathbf{x} : \mathbf{x} \in \mathfrak{R}^n, \mathbf{g}(\mathbf{x}) = \mathbf{0}, \mathbf{h}(\mathbf{x}) \geq \mathbf{0}\}$$

dan mendefinisikan suatu fungsi hukuman (*penalty function*)

$$P(x) = \begin{cases} 0, & x \in X \\ +\infty, & x \notin X, \end{cases}$$

maka masalah optimisasi dengan kekangan di atas dapat ditransformasikan menjadi masalah optimisasi tanpa kekangan sebagai berikut:

Tetapkanlah \mathbf{x}^*
 agar $F(\mathbf{x}^*) = f(\mathbf{x}^*) + P(\mathbf{x}^*)$ memiliki nilai optimum

Optimisasi Multikriteria: Metode Pembobotan Ternormalisasi

Masalah optimisasi multikriteria bisa dibawa ke bentuk masalah optimisasi satu kriteria dengan cara skalarisasi: fungsi vektor $\mathbf{f}(\mathbf{x})$ yang mencakup fungsi-fungsi sasaran $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$ ditransformasikan ke fungsi skalar $f(\mathbf{x})$. Osyczka (1984) menunjuk bahwa salah satu metode transformasi yang sangat sederhana adalah metode pembobotan ternormalisasi (*normalized weighting method*).

Prinsip metode ini adalah menjumlahkan kesemua fungsi sasaran dengan memberikan koefisien pembobot untuk tiap fungsi sasaran. Koefisien pembobot ini menunjukkan kadar pentingnya suatu fungsi sasaran relatif terhadap fungsi-fungsi yang lain. Hasil transformasi adalah

$$f(\mathbf{x}) = \sum_{i=1}^k w_i f_i(\mathbf{x}) c_i$$

dengan $w_i \geq 0$ adalah koefisien pembobot di mana $\sum_{i=1}^k w_i = 1$. Di sini $c_i \equiv 1 / f_i(\mathbf{x}^{*(i)})$

adalah faktor normalisasi dengan $f_i(\mathbf{x}^{*(i)})$ adalah harga minimum dari $f_i(\mathbf{x})$.

PENGEMBANGAN

Sistem perangkat lunak yang dibangun terdiri dari tiga subsistem pokok:

1. Program Utama. Metode numeris untuk menyelesaikan masalah optimisasi diimplementasikan di sini.
2. Parser. Penggunaan parser memungkinkan dimasukkannya fungsi-fungsi matematis ke dalam sistem pada saat *run-time*. Ini memberikan fleksibilitas walaupun mengurangi efisiensi.
3. Antarmuka (*user interface*).

Subsistem 1 dan 2 dikembangkan menggunakan bahasa pemrograman C++. Secara tradisional perangkat lunak optimisasi biasa dikembangkan dengan menggunakan FORTRAN (bandingkan Kuester dan Mize, 1982; Osyczka, 1984), namun di sini digunakan C++ karena popularitasnya di *platform* PC-Windows. Digunakan ketelitian ganda (*double precision*) agar diperoleh akurasi yang memadai. Subsistem 3 dikembangkan menggunakan bahasa pemrograman Pascal. Pascal dipilih karena relatif mudah digunakan.

Secara garis besar langkah-langkah yang terdapat dalam Program Utama adalah:

1. Mentransformasikan *masalah optimisasi multikriteria* ke *masalah optimisasi satu kriteria dengan kekangan* menggunakan metode pembobotan ternormalisasi.
2. Mentransformasikan *masalah optimisasi satu kriteria dengan kekangan* ke *masalah optimisasi satu kriteria tanpa kekangan* dengan memanfaatkan fungsi hukuman.
3. Menyelesaikan *masalah optimisasi satu kriteria tanpa kekangan* dengan metode pencarian langsung dan acak.

STUDI KASUS

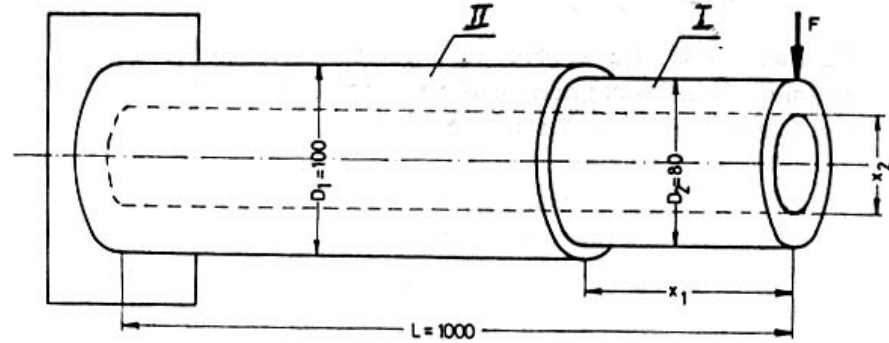
Sistem yang telah dibangun tersebut diuji dengan satu studi kasus yang akan dibicarakan di sini. Diajukan suatu permasalahan yang tergolong sebagai masalah optimisasi multikriteria berikut model matematisnya, di mana penyelesaian untuk masalah ini telah diketahui dan digunakan sebagai referensi. Selanjutnya model matematis di atas dimasukkan ke sistem, lalu keluaran sistem dibandingkan dengan referensi tersebut.

Kasus diambil dari buku teks karangan Osyczka (1984) berupa masalah perancangan balok (*beam*), yaitu penentuan ukuran-ukuran dari balok yang diperlihatkan pada Gambar 1. Ukuran-ukuran tersebut dipilih sebagai variabel keputusan:

x_1 = panjang bagian I dari balok

x_2 = diameter interior balok.

Dianggap bahwa balok mampu menanggung gaya (*force*) maksimum $F_{\max} = 12000$ N dan besarnya tegangan lentur (*bending stress*) yang dibolehkan untuk bahan balok adalah $\sigma_g = 180$ N/mm. Maka berlaku kekangan kekuatan lentur (*bending strength*)



Gambar 1. Skematik Balok

Sumber: Osyczka, A. 1984. *Multicriterion optimization in engineering*. Chichester, England: Ellis Horwood Ltd.

$$\frac{F_{\max} x_1}{(D_2^4 - x_2^4)/(32D_2)} \leq \sigma_g; \quad \frac{F_{\max} l}{(D_1^4 - x_2^4)/(32D_1)} \leq \sigma_g$$

yang melalui substitusi menghasilkan

$$\frac{9.78 \cdot 10^6 x_1}{4.096 \cdot 10^7 - x_2^4} \leq 180; \quad x_2 \leq 75.2.$$

Juga dianggap bahwa besar diameter interior balok sekurang-kurangnya 40 mm dan bahwa panjang bagian I dapat dipilih secara bebas. Maka berlaku kekangan geometrik berikut:

$$x_2 \geq 40; \quad x_1 \geq 0.$$

Hingga di sini telah diperoleh empat kekangan ketaksamaan.

$$h_1(\mathbf{x}) \equiv 180 - \frac{9.78 \cdot 10^6 x_1}{4.096 \cdot 10^7 - x_2^4} \geq 0$$

$$h_2(\mathbf{x}) \equiv 75.2 - x_2 \geq 0$$

$$h_3(\mathbf{x}) \equiv x_2 - 40 \geq 0$$

$$h_4(\mathbf{x}) \equiv x_1 \geq 0$$

Perancang balok ingin agar balok yang dirancangnya memenuhi dua sasaran:

1. Minimisasi isi balok.
2. Minimisasi kepatuhan statik (*static compliance*) balok.

Isi balok dinyatakan oleh

$$f_1(\mathbf{x}) = \frac{\pi}{4} \left[x_1 (D_2^2 - x_2^2) + (1 - x_1) (D_1^2 - x_2^2) \right] \text{mm}^3$$

dan kepatuhan statik balok dinyatakan oleh

$$f_2(\mathbf{x}) = \frac{64}{3\pi E} \left[\left(\frac{1}{D_2^4 - x_2^4} - \frac{1}{D_1^4 - x_2^4} \right) x_1^3 + \frac{l^3}{D_1^4 - x_2^4} \right] \text{mm/N}$$

di mana $E = 2.06 \cdot 10^5 \text{ N/mm}^2$ adalah modulus Young. Dengan memberikan $l = 1000 \text{ mm}$, $D_1 = 100 \text{ mm}$, $D_2 = 80 \text{ mm}$, diperoleh dua fungsi sasaran berikut:

$$f_1(\mathbf{x}) = 0.785 \left[x_1 (6400 - x_2^2) + (1000 - x_1)(10000 - x_2^2) \right]$$

$$f_2(\mathbf{x}) = 3.298 \cdot 10^{-5} \left[\left(\frac{1}{4.096 \cdot 10^7 - x_2^4} - \frac{1}{10^8 - x_2^4} \right) x_1^3 + \frac{10^9}{10^8 - x_2^4} \right]$$

Untuk optimisasi kedua fungsi sasaran secara serentak (multikriteria), perancang balok menggunakan metode pembobotan ternormalisasi dengan koefisien pembobot untuk fungsi sasaran pertama dan kedua dipilih masing-masing sebesar $w_1 = 0.2$ dan $w_2 = 0.8$. Ini ekuivalen dengan optimisasi fungsi skalar $f(\mathbf{x})$ berikut:

$$f(\mathbf{x}) = 0.2 \frac{f_1(\mathbf{x})}{2.961 \cdot 10^6} + 0.8 \frac{f_2(\mathbf{x})}{3.38 \cdot 10^4}.$$

Optimisasi multikriteria ini memberikan hasil:

$$\mathbf{x}^* = (207.0, 52.5)^T$$

$$\mathbf{f}(\mathbf{x}^*) = (5.101 \cdot 10^6, 3.62 \cdot 10^{-4})^T.$$

Model matematis di atas dimasukkan ke sistem perangkat lunak yang telah dibangun. Parameter-parameter yang digunakan oleh sistem diberi nilai sebagai berikut:

koefisien pembobot $w_1 = 0.2$ dan $w_2 = 0.8$;
 $\mathbf{xp} = (0, 40)$; $\mathbf{xk} = (1000, 76)^T$; $\mathbf{xac} = (250, 60)^T$;
 $maxm = 300$; $fpk = 0.01$; $wk = 0.0001$;
 $ntest = 300$; $nh = 3$; $ntmc = 500$; $nraz = 3$

Sistem dioperasikan pada PC dengan prosesor Pentium III 450 MHz yang bekerja pada sistem operasi Microsoft Windows 98. Ternyata penyelesaian yang diberikan oleh sistem cocok dengan hasil yang ada pada referens.

Proses masukan dan keluaran yang terjadi selama pengujian juga menunjukkan bahwa pengguna sistem ini dapat memasukkan dan menyunting model masalah beserta parameter-parameter yang dikehendaknya secara mudah dan setiap saat bisa meminta jawaban dari sistem.

KESIMPULAN

Telah dibangun sistem perangkat lunak komputer sebagai implementasi dari suatu metode numeris untuk menyelesaikan masalah optimisasi multikriteria. Hasil pengujian dengan suatu studi kasus menunjukkan bahwa sistem tersebut mudah digunakan, yaitu pengguna dapat memasukkan dan menyunting model masalah beserta parameter-parameter yang dikehendaknya secara mudah dan setiap saat bisa meminta jawaban dari sistem; selain itu sistem tersebut juga efektif, yaitu mampu memberikan penyelesaian yang akurat.

Jadi dapat disimpulkan bahwa tujuan proyek ini, yaitu mengembangkan suatu sistem pendukung keputusan untuk masalah optimisasi multikriteria, telah tercapai.

DAFTAR PUSTAKA

- Avriel, M. 1976. *Nonlinear programming*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Bronson, R. 1982. *Operations research*. New York: McGraw-Hill.
- Kuester, J.L. and Mize, J.H. 1982. *Optimization techniques with FORTRAN*. New York: McGraw-Hill.
- Osyczka, A. 1984. *Multicriterion optimization in engineering*. Chichester, England: Ellis Horwood Ltd.
- Turban, E. 1995. *Decision support and expert systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc.