

OPTIMASI MODEL-MODEL STATIK DENGAN MENGGUNAKAN SISTEM OPTIMASI VERSI-DOS

Soewarto Hardhienata *)
hardh@simulasi.lrb.lapan.go.id

INTISARI

Sistem Optimasi Versi-Dos adalah suatu paket program yang digunakan untuk optimasi model-model statik. Paket program ini ditulis dalam bahasa FORTRAN-77. Agar user (pengguna) dapat mengimplimentasikan model-model yang ia punyai dengan mudah dan baik, maka paket program ini disajikan dengan menggunakan teknik on-line-menu. Dengan penggunaan teknik ini dari user tidak dituntut pengetahuan yang mendalam tentang bahasa FORTRAN-77. Hanya dalam pengimplementasian model user harus memperhatikan syntax yang ada dalam bahasa tersebut. Pengimplementasian model dilakukan dengan cara melengkapi perintah-perintah yang diperlukan. Bentuk dari perintah-perintah tersebut diterangkan secara rinci dengan disertai contoh-contoh. User tinggal mengikuti contoh-contoh tersebut serta petunjuk yang ada pada tiap-tiap bagian dalam paket program. Untuk menyelesaikan problem-problem optimasi yang ada, disediakan empat strategi optimasi : FIBONACCI, COMPLEX, EVOLUSI dan COMBI. Cara pengoperasian program ditunjukkan dengan beberapa contoh implementasi model-model sederhana.

1. PENDAHULUAN

Model simulasi dapat dibedakan menjadi *model statik* dan *model dinamik*. Pada model statik hubungan antara variabel-variabel yang ada dalam model hanya terjadi pada satu level waktu (t) yang sama. Sedang pada model dinamik hubungan antara variabel-variabel model menyangkut level waktu yang berlainan. Disamping *besaran input x* dan *besaran output y*, pada model dinamik masih ada besaran lain *z* yang disebut dengan *besaran antara*.

Seperti yang telah disinggung pada ringkasan di depan, Sistem Optimasi Versi-DOS digunakan untuk optimasi model-model statik. Untuk model-model dinamik optimasinya digunakan Sistem Optimasi yang lain yaitu Sistem Optimasi SIMPLEX (Sistem Optimasi Model-Model Komplek) yang pembahasannya dapat ditemukan dalam makalah [HARDH94-4].

Pada Sistem Optimasi Versi-DOS ini disediakan empat Strategi Optimasi yang dapat digunakan untuk menyelesaikan problem-problem yang diberikan. Keempat strategi optimasi tersebut adalah :

- Strategi optimasi FIBONACCI dari Kiefer,
- Strategi optimasi COMPLEX dari M. J. Box,
- Strategi optimasi EVOLUSI dari H.-P. Schwefel dan
- Strategi optimasi COMBI dari S. Hardhienata.

Pembahasan secara rinci dari keempat strategi tersebut akan diberikan pada bab 2. Kemudian pada bab 3 akan

ditunjukkan cara instalasi paket program kedalam Personal Computer (PC). Pada bab 4 akan diuraikan cara implementasi model dengan menggunakan beberapa contoh. Makalah ini akan ditutup dengan saran-saran yang disajikan pada bab 5.

2. STRATEGI OPTIMASI DALAM SISTEM OPTIMASI VERSI-DOS

Berikut adalah uraian dari keempat strategi yang disediakan dalam Sistem Optimasi Versi-DOS. Agar pembaca dapat dengan mudah memahami bagaimana cara kerja strategi-strategi optimasi tersebut serta untuk memudahkan implementasi kedalam bahasa komputer, maka uraiannya sengaja dibuat mendekati bentuk bahasa komputer.

2.1. Metode Fibonacci

Metode Fibonacci diperkenalkan pertama kali oleh Kiefer pada tahun 1953 dan merupakan metode optimasi yang paling baik diantara metode-metode yang ada. Sayangnya metode ini hanya bisa digunakan untuk problem-problem yang hanya mempunyai satu variabel optimasi saja. Metode ini bekerja berdasarkan barisan bilangan dari Leonardo von Pisa yang lebih dikenal dengan nama Fibonacci. Barisan Fibonacci dihitung berdasarkan rumus seperti berikut:

$$F_0 = F_1 = 1$$
$$F_N = F_{N-1} + F_{N-2}$$

*) Anggota Kelompok Peneliti Simulasi dan Optimasi Sistem-Sistem Antariksa
Bidang Teknologi Transmisi Komunikasi Dirgantara - LAPAN

Metode Fibonacci memiliki algoritma sebagai berikut:

Phase Persiapan (diberikan oleh user)

- Fungsi sasaran : Q
- Batas Interval : G dan H dengan $G < X < H$
- Accuracy (Ketelitian): BETA

Phase Pencarian (dijalankan oleh komputer)

- 1) Penentuan panjang interval pencarian (L), dimana $L = H - G$
- 2) Penghitungan jumlah iterasi (L/BETA) yang diperlukan untuk mencapai ketelitian yang diinginkan dengan

$$F_N > L / BETA \geq F_{N-1};$$

F_N adalah bilangan Fibonacci ke N

- 3) Penghitungan harga fungsi sasaran dari $X^{(1)}$ yaitu $Q(X^{(1)})$ dengan

$$X^{(1)} = G + F_{N-2} * L / F_N$$

- 4) Penghitungan harga fungsi sasaran dari $X^{(2)}$ yaitu $Q(X^{(2)})$ dengan

$$X^{(2)} = H - F_{N-2} * L / F_N$$

- 5) Penentuan interval yang memuat optimum (disini untuk problem maksimum, untuk problem minimum tanda ketidaksamaan tinggal dibalik saja)

a. jika berlaku $Q(X^{(1)}) \leq Q(X^{(2)})$ maka optimum terletak pada interval $(G, H) = (X^{(1)}, H)$, pergi ke langkah 6.

b. jika berlaku $Q(X^{(1)}) > Q(X^{(2)})$ maka optimum terletak pada interval $(G, H) = (G, X^{(2)})$, pergi ke langkah 6.

- 6) Untuk $K = N - 1$ Down to 2 lakukan langkah-langkah seperti berikut:

- a. hitung harga fungsi sasaran dari $X^{(1)}$ yaitu $Q(X^{(1)})$ dengan

$$X^{(1)} = G + F_{K-2} * L / F_K$$

- b. hitung harga fungsi sasaran dari $X^{(2)}$ yaitu $Q(X^{(2)})$ dengan

$$X^{(2)} = H - F_{K-2} * L / F_K$$

- c. jika berlaku $Q(X^{(1)}) \leq Q(X^{(2)})$ maka interval yang dicari adalah $(G, H) = (X^{(1)}, H)$

d) jika berlaku $Q(X^{(1)}) > Q(X^{(2)})$ maka interval yang dicari adalah $(G, H) = (G, X^{(2)})$

Seperti yang telah disinggung di depan, metode Fibonacci untuk optimasi problem-problem dengan hanya satu variabel merupakan metode yang paling efisien. Akan tetapi metode ini hanya bisa digunakan untuk problem-problem yang mempunyai restriksi saja. Oleh karena itu, agar metode Fibonacci ini juga bisa digunakan untuk menyelesaikan problem-problem yang tidak mempunyai restriksi, maka interval yang memuat optimum harus dicari terlebih dahulu. Batas interval ini kemudian akan digunakan sebagai parameter awal oleh metode Fibonacci. Untuk mencari interval yang dimaksud, digunakan Metode Pencarian Langsung dengan Percepatan.

2.2 Metode Pencarian Langsung dengan Percepatan

Tujuan dari penggunaan metode ini adalah untuk menentukan interval yang memuat optimum yang akan dicari. Prinsip yang digunakan disini adalah dengan cara mengeliminir interval yang dapat dipastikan bahwa optimum tidak mungkin terletak dalam interval tersebut. Metode pencarian langsung dengan percepatan mempunyai algoritma seperti berikut :

Phase Persiapan (diberikan oleh user)

- Fungsi Sasaran : Q
- Titik Start : $X^{(0)}$
- Jangkauan awal : s

Phase Pencarian (dilakukan oleh komputer)

- 1) Penghitungan harga fungsi sasaran dari $X^{(0)}$ yaitu $Q(X^{(0)})$

- 2) Penghitungan harga fungsi sasaran dari $X^{(1)}$ yaitu $Q(X^{(1)})$ dengan $X^{(1)} = X^{(0)} + s$

- 3) Penentuan interval yang memuat optimum seperti berikut (disini untuk problem maksimum, untuk problem minimum tinggal tanda ketidaksamaan dibalik saja)

a. jika $Q(X^{(0)}) = Q(X^{(1)})$ maka interval yang dicari adalah $(X^{(0)}, X^{(1)})$, lanjutkan ke langkah 4.

b. jika $Q(X^{(0)}) < Q(X^{(1)})$ maka optimum tidak terletak pada daerah $X < X^{(0)}$

- c. Cari sepanjang sumbu X dengan arah harga X yang membesar berdasarkan rumus rekursi $X^{(k)} = X^{(k-1)} - 2^{(k-1)} * s$

- d. Hitung harga fungsi sasaran dari $X^{(k)}$ yaitu $Q(X^{(k)})$

- e. Jika berlaku $Q(X^{(k-1)}) \geq Q(X^{(k)})$ maka hentikan pencarian
- f. Interval yang dicari adalah $(X^{(k-2)}, X^{(k)})$
- g. Lanjutkan ke langkah 4.
- h. jika $Q(X^{(0)}) > Q(X^{(1)})$ maka optimum tidak terletak pada daerah $X > X^{(1)}$
- i. Cari sepanjang sumbu X dengan arah harga X yang mengecil berdasarkan rumus rekursi
 $X^{(k)} = X^{(k-1)} - 2^{(k-1)} \cdot s$
- j. Hitung harga fungsi sasaran dari $X^{(k)}$ yaitu $Q(X^{(k)})$
- k. Jika berlaku $Q(X^{(k-1)}) \geq Q(X^{(k)})$ maka hentikan pencarian
- l. Interval yang dicari adalah $(X^{(k)}, X^{(k-2)})$
- m. Lanjutkan ke langkah 4.
- 4) Lakukan eksperimen selanjutnya seperti berikut :
- 5) Hitung harga fungsi sasaran dari $X^{(k+1)}$ yaitu $Q(X^{(k+1)})$ dengan $X^{(k+1)}$ adalah titik tengah antara $X^{(k)}$ dan $X^{(k-1)}$
- a. Tentukan interval yang dicari dari titik-titik $X^{(k-2)}, X^{(k-1)}, X^{(k)}$ dan $X^{(k+1)}$ seperti berikut :
- b. jika $Q(X^{(k+1)}) > Q(X^{(k-1)})$ maka interval yang dicari adalah $(X^{(k-1)}, X^{(k)})$
- c. selesai
- d. jika $Q(X^{(k+1)}) < Q(X^{(k-1)})$ maka interval yang dicari adalah $(X^{(k-2)}, X^{(k+1)})$
- e. selesai

Interval yang ditemukan ini akan diberikan kepada Metode Fibonacci. Metode Fibonacci kemudian akan mencari harga optimum sampai ketelitian yang diinginkan dapat dipenuhi.

2.3 Strategi Optimasi COMPLEX

Strategi optimasi COMPLEX pertama kali dipublikasikan oleh Box dalam suatu "Computer Journal" tahun 1965. Nama COMPLEX berasal dari singkatan CONstrained SIMPLEX yang merupakan hasil modifikasi yang dilakukan oleh Box pada strategi optimasi SIMPLEX (bukan strategi optimasi dari Danzig untuk program linier !) yang berasal dari Spendly, Hext dan Himsworth [SPEND62].

Dalam mencari posisi optimum, strategi Complex bekerja dengan cara memperbesar, memutar dan memperkecil suatu area yang disebut Simplex. Simplex adalah suatu figur topologi (polyeder) yang membentang dalam suatu ruang berdimensi n atau R^n . Strategi optimasi Complex mempunyai algoritma seperti berikut:

Phase Persiapan (*diberikan oleh user*)

- Fungsi Sasaran : Q
- Titik Start yang memenuhi restriksi : $X^{(0)}$
- Ketelitian yang diinginkan : BETA
- Jumlah sudut Simplex yang diinginkan : k dengan $n+1 \leq k \leq 2n$
n adalah jumlah parameter optimasi (titik simplex X_j mempunyai bentuk $X_j = (x_1, x_2, \dots, x_n)$)
- Optional :
 - a. Restriksi eksplisit dalam bentuk $G_j \leq X_j \leq H_j$ untuk satu atau lebih $X_j ; j = 1, 2, 3, \dots, n$
 G_j dan H_j adalah bilangan riil dan berada dalam daerah definisi dari X_j
 G_j adalah batas bawah dari restriksi eksplisit dari parameter X_j
 H_j adalah batas atas dari restriksi eksplisit dari parameter X_j
 - b. Restriksi implisit dalam bentuk
 $F_q(X_1, X_2, \dots, X_n) \leq P_q ; q = 1, 2, \dots, m$
 F_q fungsi sembarang

Phase Pencarian (*dijalankan oleh komputer*)

M adalah himpunan titik-titik yang memenuhi daerah definisi

1) Bentuk suatu Simplex permulaan dengan k segi

Berawal dari $X^{(0)}$ (titik start), munculkan kekurangan (k-1) segi $X_p ; p = 1, 2, 3, \dots, (k-1)$ dengan suatu proses acak seperti berikut :

$$X_{pj} = G_j + R_{pj} (H_j - G_j) ; j = 1, 2, 3, \dots, n$$

X_{pj} adalah koordinat ke j dari titik P
 R_{pj} adalah bilangan acak dalam interval (0,1)

Periksa apakah X_p melanggar restriksi implisit, jika demikian halnya, maka geser dengan cara seperti berikut:

$$X_p = 0.5 (X_p + X)$$

sampai X_p terletak pada daerah definisi

X adalah titik berat dari simplex tanpa titik terjelek.

2) Hitung harga fungsi sasaran dari tiap-tiap titik pojok simplex.

3) Tentukan titik pojok X_s yang merupakan titik pojok dengan harga fungsi sasaran yang terjelek.

4) Perbaiki X_s dengan refleksi seperti berikut :

$$X_s = X_s + \text{ALPHA} * (X - X_s) ;$$

dimana $\text{ALPHA} > 1$

Hitung harga fungsi sasaran dari X_s

Jika setelah refleksi X_s masih merupakan titik yang terjelek, maka lakukan kontraksi seperti berikut :

$$X_s = 0.5 (X_s + X) \text{ kearah pusat,}$$

sampai usaha ini berhasil.

Periksa, apakah X_s melanggar restriksi eksplisit. Jika demikian halnya, maka geser X_s seperti berikut :

$$X_{sj} = G_j + \text{DELTA} \text{ atau } X_{sj} = G_j - \text{DELTA}$$

Jika X_s melanggar restriksi implisit, maka kurangi jarak X_s dari sentra X setengahnya seperti berikut :

$$X_s = 0.5 (X_s + X) ,$$

sampai semua persyaratan dipenuhi.

5) Periksa apakah kriteria berhenti :

$$Q(X_b) - Q(X_s) \leq \text{BETA}$$

dipenuhi berturut-turut sebanyak GAMMA iterasi, dimana x_b adalah titik terbaik dari simplex aktual, x_s adalah titik terjelek dari simplex aktual dan BETA adalah ketelitian untuk kriteria berhenti. Jika demikian halnya, maka pencarian optimum dihentikan. Sebagai hasilnya adalah X_b dengan harga fungsi sasaran $Q(X_b)$. Jika tidak demikian halnya, maka lanjutkan iterasi berikutnya (kembali ke langkah 4.)

Box menyarankan pemilihan harga seperti berikut :

- $\text{ALPHA} = 1.3$
- $k = 2n$
- $\text{DELTA} = 0.00001$

Pemberitahuan :

Strategi Complex dalam Sistem Optimasi Versi-DOS ini hanya digunakan untuk menyelesaikan problem-problem optimasi dengan dua variabel atau lebih. Untuk problem-problem dengan hanya satu variabel optimasinya dilakukan dengan menggunakan metode Fibonacci.

2.4 Strategi Optimasi EVOLUSI

Dalam alam ini, makhluk hidup yang dapat lebih menyesuaikan diri dengan lingkungannya akan lebih

berhasil berkembang dari pada makhluk hidup yang kurang bisa menyesuaikan diri dengan lingkungannya. Proses ini pertama kali dikemukakan oleh C. Darwin (1859) dan dikenal dengan istilah Evolusi.

Strategi optimasi evolusi merupakan strategi optimasi yang bekerja berdasarkan mekanisme dari evolusi biologis yang pada tahun 1973 dipublikasikan pertama kali oleh Rechenberg [RECH73]. Rechenberg mencoba mengaplikasikan proses evolusi ini pada bidang numerik untuk menyelesaikan problem-problem optimasi. Strategi ini, yang juga dikenal sebagai Skema Evolusi dari dua anggota, mempunyai algoritma seperti berikut :

Berawal dari satu Induk $X_E^{(g)}$ dari generasi ke g dimunculkan satu keturunan $X_N^{(g)}$ melalui suatu proses acak. Individu yang memiliki sifat-sifat yang lebih baik bisa bertahan hidup dan bisa menghasilkan keturunan, sedangkan individu yang lainnya musnah.

Dibanding dengan proses evolusi yang sesungguhnya, di sini terlihat adanya penyederhanaan yang sangat tajam. Sebagai contoh bahwa jumlah populasi dari tiap-tiap generasi selalu konstan (dalam hal ini setiap generasi hanya terdiri dari dua individu) dan sebuah individu secara prinsip dapat bertahan hidup secara tak terbatas dan dapat terus menghasilkan keturunan.

Bagi strategi optimasi ini berarti bahwa Induk pertama $x_E^{(0)}$ akan berperan sebagai titik start. Titik start ini didefinisikan melalui harga dari tiap-tiap parameter optimasi $X_{E,i}^{(0)}$, $i = 1,2,3,\dots,n$. Disamping itu untuk setiap $X_{E,i}^{(g)}$ harus dilengkapi dengan Standard deviasi SD_i , $i = 1,2,2,\dots,n$. Standard deviasi ini akan digunakan untuk menentukan harga koordinat dari titik yang baru (koordinat dari keturunan) dan merupakan ukuran jarak antara induk dengan keturunannya. Standard deviasi ini (yang juga disebut sebagai jangkauan) selama proses berjalan, besarnya akan dibuat bervariasi dan disesuaikan. Pada sejumlah generasi tertentu akan dihitung besarnya perbandingan antara keturunan yang berhasil dan yang tidak berhasil (jumlah mutasi). Semua harga standard deviasi SD_i akan diperkecil jika harga perbandingan ini dibawah batas tertentu yang diberikan, sebaliknya harga standard deviasi akan diperbesar.

Berdasarkan suatu hasil penyelidikan H.-P. Schwefel menyarankan suatu cara untuk mendapatkan pengontrolan standard deviasi yang optimal yang disebut dengan Rumus 1/5 Keberhasilan. Rumus tersebut bekerja seperti berikut [SCHW77] :

"Periksa setiap n mutasi, berapa besar angka keberhasilan yang tercatat selama 10n mutasi. Jika angka keberhasilan tersebut lebih kecil dari pada 2n, maka kalikan standard deviasi dengan faktor 0.85, jika angka keberhasilan lebih besar dari pada 2n maka bagi standard deviasi dengan faktor 0.85."

Pencarian optimum akan dihentikan jika :

$$w(g-dg) - w(g) \leq E_c \text{ atau} \\ 1/E_d(w(g-dg) - w(g)) \leq /w(g)/$$

dimana $w(g)$: harga fungsi sasaran pada generasi ke g
 dg : jarak generasi ke generasi dimana kriteria berhenti diperiksa ($dg \geq 20$)
 E_c, E_d : batas-batas yang diberikan oleh user ($E_c > 0$ dan $1 + E_d > 1$)

Schwefel menyarankan agar pemeriksaan kriteria berhenti dilakukan minimal pada tiap-tiap jarak 20n generasi. Interval tersebut menurut rumus $1/5$ diatas cukup untuk merubah faktor $(0.85)^{+20} = 25^{-1}$ dan untuk menghindari penyetonan pencarian optimum yang disebabkan oleh penyesuaian standard deviasi yang kurang baik.

Pada strategi optimasi Evolusi, restriksi harus dipasang dalam bentuk ketidaksamaan $G_j(X) \geq 0$ untuk $j = 1, 2, 3, \dots, m$. Jika sebuah titik melanggar restriksi, maka titik tersebut akan dianggap sebagai kegagalan dan akan dibuang. Dalam hal ini standard deviasi akan diperkecil dan dimunculkan titik baru sampai semua restriksi dipenuhi. Ini berarti bahwa jika sering terjadi pelanggaran restriksi maka standard deviasi akan sering pula diperkecil. Hal ini dapat mengakibatkan kecepatan konvergensi akan sangat berkurang.

Algoritma strategi Evolusi dapat diformulasikan seperti berikut :

- 1) Pilih sebuah titik start $X_E(g)$ pada daerah yang memenuhi restriksi. pasang penghitung mutasi $g = 0$.
- 2) Munculkan satu titik keturunan (anak) $X_N(g)$ seperti berikut

$$X_N(g) = X_E(g) + Z(g)$$

dengan komponen-komponen :

$$X_{N,i}(g) = X_{E,i}(g) + Z_i(g) ; i = 1, 2, 3, \dots, n \\ Z_i(g) \text{ adalah } (0, SD_i) \text{-Distribusi Normal}$$

Periksa, apakah $X_N(g)$ melanggar restriksi. Jika demikian halnya, maka lakukan langkah-langkah seperti berikut : a. buang titik tersebut,
 b perkecil standard deviasi & munculkan titik baru sampai semua restriksi dipenuhi.

- 3) Tentukan seperti berikut (untuk problem minimum):

$$X_E(g+1) = X_N(g) \text{ kalau } Q(X_N(g)) < Q(X_E(g)) \\ X_E(g+1) = X_E(g) \text{ kalau } Q(X_N(g)) > Q(X_E(g))$$

Pasang $g := g + 1$ dan pergi ke langkah 2.

2.5 Strategi EVOLUSI Beranggota Banyak

H.-P. Schwefel berpendapat bahwa konsep evolusi yang dipakai oleh Rechenberg terlalu sederhana. Oleh karena itu dia mencoba mengembangkan strategi dasar dari Rechenberg dengan cara pemunculan keturunan tidak lagi hanya melalui satu individu, tetapi melalui suatu kumpulan gen dengan banyak variasi. Dengan cara ini kemungkinan suatu keturunan akan mempunyai sifat-sifat yang lebih baik menjadi lebih besar. Dari keturunan-keturunan yang dihasilkan akan dipilih sebanyak u keturunan yang terbaik, kemudian dijadikan sebagai induk untuk generasi berikutnya. Induk ini dibiarkan menghasilkan sebanyak v keturunan ($v \gg u$) dan dari v keturunan ini dipilih lagi sebanyak u keturunan yang terbaik dan dijadikan sebagai induk untuk generasi berikutnya. Proses ini akan berjalan secara siklis sampai dengan kriteria berhenti yang diberikan dapat dipenuhi.

Peralihan dari satu generasi ke generasi berikutnya serta pemunculan dari tiap-tiap individu berjalan seperti berikut :

Sama seperti pada strategi evolusi dengan dua anggota, pada strategi dengan banyak anggota ini pada permulaan pencarian optimum juga harus diberikan titik start terlebih dahulu dan tiap-tiap parameter optimasi juga harus dilengkapi dengan standard deviasi. Dengan data ini kemudian akan dimunculkan generasi yang pertama. Karena kita membutuhkan sebanyak u induk, maka pada phase ini kita masih mempunyai kekurangan sebanyak $u-1$ induk. Pemunculan u -induk ini prosesnya sama dengan proses yang dipakai dalam pemunculan keturunan pada strategi dengan dua anggota. Baru pada pemunculan keturunan berikutnya prosesnya menjadi berbeda, karena disini digunakan mekanisme yang lain, yaitu setiap kali sebelum sebuah keturunan di munculkan, terlebih dahulu akan dipilih satu induk secara acak. Induk yang terpilih ini kemudian akan digunakan untuk memunculkan keturunan. Secara keseluruhan terjadinya keturunan dapat diuraikan seperti berikut :

Dari sebanyak u induk, dipilih secara acak satu titik (E) yang kemudian digunakan sebagai induk untuk memunculkan sebuah keturunan.

- Tentukan faktor SA sebagai adaptasi (penyesuaian) dari standard deviasi.
- Untuk $N = 1, 2, 3, \dots, n$ tentukan standard deviasi untuk keturunan dengan cara: $SD_{N,i} = SD_{Ei} \times SA$
- Untuk $N = 1, 2, 3, \dots, n$ tentukan vektor parameter optimasi dengan cara: $X_{N,i} = X_{Ei} \times Z_i$ dimana $Z_i: (0, SD_{N,i})$ - Distribusi Normal

Faktor SA untuk adaptasi dari standard deviasi pada umumnya mempunyai harga 1. Hanya jika pada pemunculan keturunan terjadi pelanggaran restriksi maka faktor SA ini akan dikurangi setengahnya.

Proses yang diuraikan diatas akan dijalankan sampai diperoleh sebanyak v keturunan. Dari v keturunan ini kemudian akan dipilih sebanyak u keturunan yang terbaik. Sebagai ukurannya disini adalah harga dari fungsi sasaran. u keturunan yang terpilih ini kemudian akan di jadikan induk untuk memunculkan generasi berikutnya. Siklus ini akan dijalankan terus sampai kriteria berhenti yang diberikan dapat dipenuhi.

Kriteria berhenti dipenuhi jika :

$$w_s(g) - w_b(g) \leq E_c \text{ atau} \\ \frac{u/E_d (w_s(g-dg) - w_b(g)) \leq}{Q(X_1(g)) + Q(X_2(g)) + \dots + Q(X_u(g))}$$

- dimana $w_s(g)$: harga fungsi sasaran terjelek pada generasi ke g
 $w_b(g)$: harga fungsi sasaran terbaik pada generasi ke g
 u : jumlah induk
 dg : jarak generasi ke generasi dimana kriteria berhenti diperiksa ($dg \geq 20$)
 E_c, E_d : batas-batas yang diberikan oleh user ($E_c > 0$ dan $1 + E_d > 1$)

2.6 Sifat-sifat dari Strategi Complex dan Evolusi

Strategi Complex dan Evolusi mempunyai sifat-sifat positif seperti berikut :

- tidak diperlukan diferensial parsial,
- fungsi sasaran tidak harus dipasang secara eksplisit,
- restriksi konvex/tidak konvex tidak menjadi masalah,
- handal terhadap deviasi harga dari fungsi sasaran dan kesalahan pembulatan,
- problem dengan/tanpa restriksi tambahan tetap bisa diselesaikan.

Disamping itu masing-masing strategi masih memiliki kelebihan dan kelemahan seperti berikut :

Strategi Complex :

- Kelebihan** : + kecepatan konvergensi relatif tinggi
 + sedikit kontrol parameter
 + pemilihan harga dari kontrol parameter tidak begitu mempengaruhi kemampuan dari strategi Complex

Kelemahan : - probabilitas untuk bisa menemukan optimum kecil pada problem-problem yang memiliki banyak optima.

Strategi Evolusi :

Kelebihan : + probabilitas untuk bisa menemukan optimum besar pada problem-problem yang memiliki banyak optima.

Kelemahan : - kecepatan konvergensi rendah
 - banyak kontrol parameter
 - pemilihan harga dari kontrol parameter besar pengaruhnya terhadap kemampuan dari strategi Evolusi.

Kalau kelebihan dan kelemahan dari kedua strategi diatas kita perhatikan secara cermat, maka kita temukan keadaan yang sangat menarik. Hal yang menjadi keuntungan dari satu strategi akan menjadi kelemahan dari strategi yang lainnya dan sebaliknya. Dalam bab berikut akan diuraikan satu strategi yang mencoba menggabungkan kelebihan-kelebihan yang dimiliki oleh strategi Complex dan Evolusi untuk kemudian dimanfaatkan secara optimal.

2.7 Strategi Optimasi COMBI

Strategi Combi adalah strategi optimasi yang dikembangkan berdasarkan strategi Complex dan Evolusi. Dalam strategi Combi ini sifat-sifat baik yang dimiliki oleh strategi Complex dan Evolusi dicoba disatukan dan dimanfaatkan secara optimal. Berdasarkan penyelidikan yang dilakukan dengan menggunakan test-test fungsi standar dalam [HARD93-1], dapat ditunjukkan bahwa strategi Combi memiliki sifat-sifat seperti berikut :

- Untuk problem-problem dengan banyak optima, probabilitas untuk dapat menemukan optimum global minimal sama dengan yang dimiliki oleh strategi Evolusi.
- Kecepatan konvergensi dari strategi Combi jauh lebih tinggi dari pada kecepatan konvergensi dari strategi Evolusi.

Disini kecepatan konvergensi sesuai dengan jumlah penghitungan fungsi sasaran yang dibutuhkan untuk mencari posisi optimum. Jumlah ini menjadi penting karena penghitungan fungsi sasaran biasanya berhubungan dengan waktu simulasi yang cukup lama.

Strategi Combi memiliki algoritma seperti berikut (pembahasan secara lengkap dapat ditemukan pada [HARDH93-1] :

Phase Persiapan (diberikan oleh user)

- Fungsi Sasaran (Q)
- Titik Start yang memenuhi restriksi ($X^{(0)}$)
- Ketelitian yang diinginkan (BETA)
- Jumlah sudut Simplex yang diinginkan (k) dengan $n+1 \leq k \leq 2n$
n adalah jumlah parameter optimasi, titik simplex X_j memp. bentuk $X_j = (x_1, x_2, \dots, x_n)$
- Jumlah yang diinginkan untuk pengetesan dari optimum yang ditemukan (MAXTES)
- Jumlah maksimal dari generasi untuk phase evolusi (MAXGEN)
- Optional :

- Restriksi eksplisit dalam bentuk $G_j \leq X_j \leq H_j$ untuk satu atau lebih $X_j ; j = 1, 2, 3, \dots, n$
 G_j dan H_j adalah bilangan riil dan berada dalam daerah definisi dari X_j
 G_j adalah batas bawah dari restriksi eksplisit dari parameter X_j
 H_j adalah batas atas dari restriksi eksplisit dari parameter X_j

- Restriksi implisit dalam bentuk $F_q(X_1, X_2, \dots, X_n) \leq P_q ; q = 1, 2, \dots, m$
 F_q fungsi sembarang

Phase Pencarian (dijalankan oleh komputer)

- 1) Berawal dari titik start $X^{(0)}$ yang diberikan, dicari sebuah optimum X^{opt} dengan menggunakan strategi Complex.
- 2) Periksa apakah jumlah maksimal yang diinginkan untuk pengetesan optimum sudah tercapai atau belum. Jika sudah tercapai maka hentikan proses optimasi dan berikan X^{opt} sebagai hasil optimasi. Selesai.
- 3) Periksa X^{opt} dengan menggunakan mekanisme evolusi seperti berikut :
 - a. Berawal dari X^{opt} munculkan sebanyak u-1 titik induk.
 - b. Tentukan titik terbaik X^{best} dan bandingkan apakah X^{best} lebih baik dari pada X^{opt} . jika demikian halnya, maka pasang X^{best} sebagai titik start kemudian cari titik optimum berikutnya dengan menggunakan strategi Complex (pergi ke langkah 1 dengan $X^{(0)} = X^{best}$).
 - c. Munculkan sejumlah v titik keturunan.
 - d. Tentukan titik terbaik X^{best} dan bandingkan apakah titik tersebut lebih baik dari pada X^{opt} .

Jika demikian halnya, maka pasang X^{best} sebagai titik strat yang baru dan kemudian cari titik optimum berikutnya dengan menggunakan strategi Complex (pergi ke langkah 1 dengan $X^{(0)} = X^{best}$)

- e. Periksa apakah jumlah maksimal dari generasi untuk phase evolusi MAXGEN sudah tercapai atau belum. Bila sudah tercapai hentikan proses optimasi dan berikan X^{opt} sebagai hasil optimasi. Selesai.
- f. Dari v titik keturunan, pilih sebanyak u titik terbaik dan pasang sebagai titik induk. Pergi ke langkah 3.c.

Pemberitahuan :

- Pada strategi Combi dipasang u = 10 dan v = 100. Jika diinginkan, user dapat memodifikasi harga u dan v ini.
- Pada pencarian optimum berikutnya (jika pada pengetesan optimum dapat ditemukan titik lain yang lebih baik dari pada titik optimum) titik optimum baru yang akan ditemukan selalu akan lebih baik dari pada titik optimum sebelumnya. Hal ini dijamin oleh algoritma yang digunakan untuk mencari optimum disini, karena harga optimum yang ditemukan akan selalu lebih baik dari pada harga titik start yang diberikan. Oleh karena itu kemungkinan pengulangan pencarian optimum yang berkepanjangan tidak akan terjadi.

3 INSTALASI PAKET PROGRAM KEDALAM KOMPUTER

Paket program Sistem Optimasi Versi-DOS disediakan untuk diaplikasikan dengan menggunakan Personal Computer (PC). Spesifikasi hardware dari PC disyaratkan minimal seperti berikut :

- RAM 640 Kbyte
- Floppy Disk
- Hard Disk
- Printer (80 Character)

Spesifikasi software :

- Operating System DOS
- Compiler PRO-FORTRAN 77

Sistem Optimasi Versi-DOS terdiri dari 5 (lima) Disket dengan isi seperti berikut :

Disket 1 :

- Program-program untuk tampilan Menu
- Program Editor

Disket 2 :

- System Subprogram Optimasi

Disket 3, 4 dan 5 :

- FORTRAN Compiler (Pro Fortran 77)

Untuk menginstalasikan program keseluruhan harus dilakukan langkah-langkah seperti berikut :

- a. Membuat Directory Optimasi pada Hard Disk dengan perintah :

```
MKDIR OPTSYS (Enter)
```

- b. Masuk kedalam Directory OPTSYS dengan perintah :

```
CD OPTSYS (Enter)
```

- c. Mengcopy isi Disket kedalam Hard Disk :

Masukkan Disket 1 ke dalam Floppy A, kemudian tulis perintah

```
COPY A:*.* C:\OPTSYS (Enter)
```

Langkah ini diulang untuk Disket 2.

- d. Mengcopy FORTRAN Compiler ke dalam Hard Disk:

- Buat directory PROFOR dengan perintah :
MKDIR PROFOR (Enter)
- Masuk ke directory PROFOR dengan perintah :
CD PROFOR (Enter)
- Masukkan Disket 3 ke dalam Floppy A dan tulis perintah :
COPY A:*.* C:\PROFOR (Enter)
Langkah ini diulang untuk Disket 4 dan 5.

Agar komunikasi langsung antara directory OPTSYS dan PROFOR dapat dilakukan, perlu dipasang PATH dalam File AUTOEXEC.BAT seperti berikut :

```
PATH=C:\OPTSYS;C:\PROFOR;
```

Setelah langkah-langkah tersebut diatas selesai dilakukan, maka komputer kita reboot agar path yang kita pasang bisa berfungsi.

Untuk memanggil Sistem Optimasi Versi-DOS yang telah kita instalasikan ini cukup kita tuliskan :

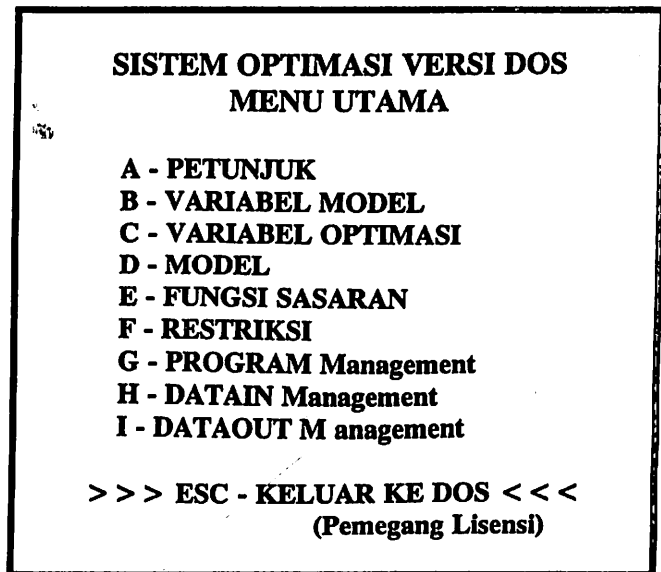
```
OPTSTART (Enter)
```

4 PENGOPERASIAN PROGRAM

Jika program sudah diinstalasikan kedalam hard disk seperti yang telah diuraikan pada bab 3 di depan, maka untuk mengaktifkannya tinggal dipanggil dengan perintah :

```
OPTSTART (Enter).
```

Dengan pemanggilan tersebut MENU UTAMA akan muncul di layar monitor seperti yang terlihat pada gambar 4-1 berikut :

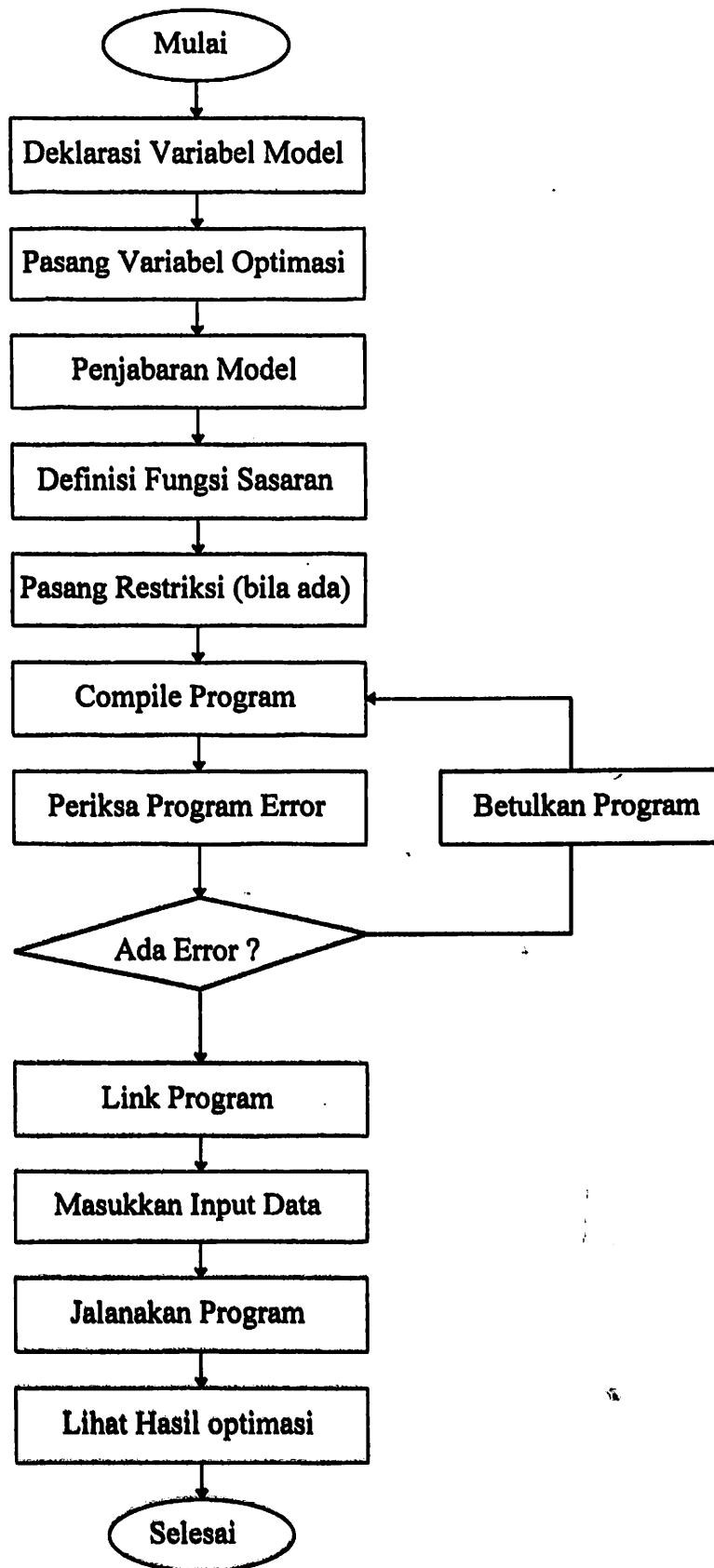


Gambar 4-1 : Tampilan Monitor Menu Utama

Jika user menekan salah satu huruf (A - I), maka ia akan berada pada posisi Editing atau pada salah satu SUB MENU. Jika berada pada posisi Editing, maka ia akan diminta untuk melengkapi bagian program yang telah disediakan sesuai dengan contoh dan keterangan yang ada.

Jika semua bagian program sudah dilengkapi, user dapat mulai mengkompail (menerjemahkan) program tersebut dan kemudian menyambungkannya dengan routine optimasi dan fortran-library yang ada. Dan setelah input data dipersiapkan, user bisa memerintahkan agar program dijalankan untuk mencari harga optimum dari persoalan yang diberikan. Untuk keperluan ini semua, user tinggal menekan tombol-tombol pada keyboard yang sesuai dan telah disediakan.

Secara skematis langkah-langkah yang harus dilakukan oleh user untuk optimasi dengan Sistem Optimasi Versi-DOS ini dapat dilihat pada flowchart pada halaman berikut ini. Contoh-contoh implementasi yang diberikan pada bab berikutnya diharapkan dapat memperjelas cara pengoperasian paket program.



Gambar 4-2. LANGKAH-LANGKAH DALAM IMPLEMENTASI MODEL DAN PENGOPERASIAN PROGRAM

Contoh Implementasi

Berikut ini akan ditunjukkan bagaimana User mengimplementasikan problem (Model) kedalam paket program.

Contoh-1 (Problem dengan Restriksi eksplisit)

Sebagai contoh persamaan dibawah ini kita anggap sebagai model yang akan kita optimasikan

$$Y_1 = 4 X_1^2 - 2.1 X_1^4$$

$$Y_2 = 0.33 X_1^6 - X_1 X_2$$

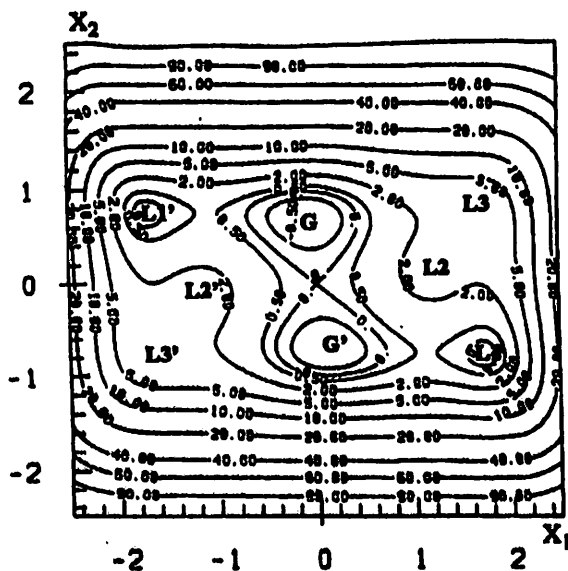
$$Y_3 = 4 X_2^2 - 4 X_2^4$$

Dengan Fungsi Sasaran $Q = Y_1 + Y_2 - Y_3$ minimal !

Restriksi yang diberikan misalnya : $-2.5 < X_1 < +2.5$
 $-2.5 < X_2 < +2.5$

Informasi :

Fungsi Sasaran $Q = Y_1 + Y_2 - Y_3$ ini mempunyai 2 (dua) Optima global (G) pada posisi (0.89,-0.712) dan (-0.89,0.712) dengan harga fungsi sasaran -1.0316 dan beberapa Optima lokal (L) seperti yang terlihat pada Gambar 4-3. Fungsi ini dikenal dengan nama Fungsi-Test dari Branin.



Gambar 4-3 : Garis-garis Ketinggian dari Fungsi Q

Implementasi

Untuk mengimplementasikan problem tersebut diatas, User harus melaksanakan langkah-langkah seperti berikut :

4.1 Deklarasi Variabel Model

Dengan menekan huruf B (MENU UTAMA : B - VARIABEL MODEL) User akan berada pada posisi editing untuk Deklarasi Variabel Model.

Deklarasi Variabel Model tersebut harus dimasukkan di tempat yang sudah disediakan yaitu antara AWAL DEKLARASI VARIABEL MODEL dan AKHIR DEKLARASI VARIABEL MODEL dengan mengikuti contoh dan petunjuk yang ada.

Deklarasi Variabel Model formatnya seperti berikut :

ONAME(I) = 'NAMA VARIABEL'

dimana I = 1,2,3,...,n dengan n = Jumlah Variabel

Karena Model diatas hanya mempunyai 2(dua) variabel, maka bentuk deklarasinya seperti berikut :

C AWAL DEKLARASI VARIABEL MODEL
 C =====

ONAME(1) = 'X1'
 ONAME(2) = 'X2'

C =====
 C AKHIR DEKLARASI VARIABEL MODEL

Untuk men-SAVE deklarasi variabel model tersebut serta untuk kembali ke MENU UTAMA harus ditekan berturut-turut ESC, F, S dan ESC, Q, Q.

4.2 Pemasangan Variabel Optimasi

Variabel Model yang telah dideklarasikan dan sekaligus berfungsi sebagai Variabel Optimasi harus dipasang pada posisi yang disediakan (MENU UTAMA : C - VARIABEL OPTIMASI).

Pemasangan variabel optimasi mempunyai Format seperti berikut :

NAMA VARIABEL = COMVEC(I)

dimana I = 1, 2, 3, ..., n dengan n = Jumlah Variabel Optimasi

Karena kedua variabel X1 dan X2 dalam contoh diatas juga menjadi Variabel Optimasi, maka pemasangannya mempunyai bentuk seperti berikut :

C AWAL PEMASANGAN VAR. OPTIMASI
 C =====

X1 = COMVEC(1)
 X2 = COMVEC(2)

C =====
 C AKHIR PEMASANGAN VAR. OPTIMASI

Tekan berturut-turut ESC, F, S dan ESC, Q, Q untuk men-SAVE pemasangan variabel optimasi dan kembali ke MENU UTAMA.

4.3 Memasukkan Model

Model harus dibuat dalam bentuk Persamaan Matematika. Pada contoh ini penjabaran model mempunyai bentuk seperti berikut :

C **AWAL PENJABARAN MODEL**
C =====

$$Y1 = 4.0 * X1**2 - 2.1 * X1**4$$

$$Y2 = 0.33 * X1**6 + X1 * X2$$

$$Y3 = 4.0 * X2**2 - 4.0 * X2**2$$

C =====
C **AKHIR PENJABARAN MODEL**

Tekan berturut-turut ESC, F, S dan ESC, Q, Q untuk men-SAVE Model dan kembali ke MENU UTAMA.

4.4 Mendefinisikan Fungsi Sasaran

Karena setiap Optimasi mutlak memerlukan adanya Fungsi Sasaran maka fungsi tersebut harus kita definisikan (MENU UTAMA : E - FUNGSI SASARAN).

Definisi dari Fungsi Sasaran mempunyai Format seperti berikut :

$$COMVEC(0) = FUNGSI SASARAN$$

Karena itu definisi dari Fungsi Sasaran bentuknya seperti berikut :

C **AWAL DEFINISI FUNGSI SASARAN**
C =====

$$COMVEC(0) = Y1 + Y2 - Y3$$

C =====
C **AKHIR DEFINISI FUNGSI SASARAN**

Tekan berturut-turut ESC, F, S dan ESC, Q, Q untuk men-SAVE Fungsi Sasaran dan kembali ke MENU UTAMA.

4.5 Pemasangan Restriksi

Problem diatas mempunyai dua Restriksi eksplisit. Restriksi tersebut harus dipasang dalam Input-File "DATAIN" pada kumpulan data OPAR (lihat kolom kedua terakhir dan kolom terakhir dari kumpulan data OPAR pada Sub Bagian g. Memasukkan Input-Data).

4.6 Mengkompail dan men-link Program

Setelah langkah-langkah a s/d e selesai dikerjakan, User harus melanjutkan langkah-langkah berikut sesuai dengan urutan yang diberikan :

- Compile Program
- Check (error) Program
- Menyambung Program dengan Routine Optimasi

Dalam SUB MENU PROGRAM MANAGEMENT telah tersedia semua perintah yang dibutuhkan. User hanya perlu menekan tombol yang sesuai.

Petunjuk :

Jika dalam kompail program muncul kesalahan-kesalahan (hal ini bisa dilihat dalam SUB MENU PROGRAM MANAGEMENT : B - Check PROGRAM), maka User harus terlebih dahulu membetulkan kesalahan tersebut dan kemudian program di kompail kembali. Sesudah itu baru program disambung dengan Routine - Optimasi (SUB MENU PROGRAM MANAGEMENT : C - Link PROGRAM).

4.7 Memasukkan Input-File "DATAIN"

Untuk menjalankan Optimasi program memerlukan Input-Data dan Control-Parameter. Keduanya harus dimasukkan kedalam Input-File "DATAIN" (SUB MENU : DATAIN MANAGEMENT - A - DATAIN). Untuk problem yang kita punyai ini Input-File DATAIN mempunyai bentuk seperti berikut :

```
TEXT; CONTOH OPTIMASI 1/
STRA; COMP/
COMP; MIN; 0; 500; 1.E-5; 3; 30; 1; 0; 0/
OPAR; X1; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OPAR; X2; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OEND/
```

Input diatas memuat empat bagian yang tandai dengan Nama-Komando TEXT, STRA, COMP dan OPAR. Pembacaannya menurut format yang telah ditentukan, karena itu dalam memasukkan data User harus mengikuti Format-Standard dari bahasa Fortran77. Antara Data digunakan tanda pemisah ";". Symbol akhir dari suatu kelompok data adalah "/". Input-File DATAIN ini harus diakhiri dengan "OEND/".

TEXT

Tulisan yang berada dibelakang Nama-Komando TEXT dipakai sebagai Judul/Keterangan yang akan dicetak pada Output-File DATAOUT. Untuk User disediakan sebanyak tiga baris, dimana tiap baris maksimal boleh sampai 76 huruf. Pada contoh diatas dipilih sebagai judul "CONTOH OPTIMASI 1".

STRA

Nama-Komando ini digunakan untuk memilih Strategi Optimasi yang dipakai. Dalam paket program ini tersedia tiga Strategi Optimasi :

STRA; COMP/ => Optimasi akan dicari dengan menggunakan Complex-Strategi

STRA; EVOL/ => Optimasi akan dicari dengan menggunakan Evolution-Strategi

STRA; COMB/ => Optimasi akan dicari dengan menggunakan Combi-Strategi

COMP

Dibelakang Nama-Komando COMP dimasukkan Control-Parameter untuk Complex-Strategi.

COMP Nama-Komando

- MIN** Jenis problem minimal
(MAX untuk jenis problem maximal)
- 0** Jumlah Restriksi implisit (M)
Pada contoh kita diatas tidak terdapat restriksi implisit.
- 500** Jumlah maximal dari penghitungan Fungsi Sasaran yang diijinkan (SIMMAX > 0)
- 1.E-5** Ketelitian Optimasi (BETA>0.0)
BETA adalah parameter untuk Kriteria-Berhenti.
- 3** Jumlah Sudut Simplex yang dipilih (K)
K boleh mempunyai harga $N + 1 \leq K \leq 2N$, dimana N adalah jumlah parameter Optimasi.
- 30** Nomor dari Random-Generator yang dipilih (ORAND) ($1 \leq \text{ORAND} \leq 30$)
- 1** Kontrol-Parameter untuk strat-baru (NEUSTA)
NEUSTA = 0 : Tidak ada start baru
NEUSTA = 1 : Akan dijalankan strat baru sampai Kriteria Berhenti dipenuhi.
- 0** Parameter untuk mengontrol banyaknya Output dari Routine optimasi (OPTPRT)
OPTPRT = 0 : Tidak ada output kecuali keterangan error
OPTPRT = 1 : Semua langkah akan dibuat protokolnya dengan, nomor-nomor dari titik-titik yang dihasilkan akan dicatat.
- 0** Parameter untuk mengontrol banyaknya output dari protokol simulasi (SIMPRT)

Petunjuk :

Parameter SIMPRT tidak punya pengaruh apa-apa disini, karena parameter tsb hanya dipakai untuk Simulator GPSS-FORTRAN yang dipasang pada Computer besar.

OPAR

Harga awal dari setiap Parameter Optimasi diberikan oleh User melalui kumpulan data OPAR. Dalam contoh ini ada dua kumpulan data OPAR untuk X1 dan X2.

OPAR Nama-Komando

- X1** Nama Parameter Optimasi (NAME) (Boleh sampai 8 Character)
- +1** Harga untuk Start Optimasi (STARTWERT)
- 0.75** Standard Deviasi / Jangkauan awal (SD)
- 0.001** Batas absolut yang tidak boleh dilewati oleh Standard Deviasi (EA)
- 0.001** Batas relatif (EB)
- 0** Jenis dari daerah definisi untuk parameter yang bersangkutan
- 0 atau blank** : kontinu
- 1** : bilangan bulat
- 2** : fungsi dari bilangan bulat
- 3** : tabel harga melalui kumpulan data ODIS
- 2.5** Batas bawah untuk parameter optimasi X1 (G)
- +2.5** Batas atas untuk parameter optimasi X1 (H)

Petunjuk :

- (SD), (EA) dan (EB) tidak mempunyai arti untuk Complex-Strategi dan hanya akan digunakan untuk Evolution-Strategi dan Combi-Strategi.
- Untuk Optimasi tanpa Restriksi eksplisit kolom untuk (G) dan (H) harus dikosongkan.

4.8 Menjalankan Program

Dengan menekan tombol D dalam SUB MENU : PROGRAM MANAGEMENT, program akan dijalankan secara otomatis oleh komputer.

4.9 Tampilan Hasil Optimasi

Setelah program selesai dijalankan, hasilnya bisa dilihat pada file DATAOUT dengan cara menekan tombol A pada SUB MENU : DATAOUT MANAGEMENT. Berikut ini adalah contoh print-out hasil optimasi dari Contoh-1:

OPTIMASI

=====

DATA INPUT :

TEXT; CONTOH OPTIMASI -1/
STRA; COMP/
COMP; MIN;0;500;1.E-5;3;30;1;0;0/
OPAR; X1; +1.;0.75;0.001;0.001;0;-2.5;2.5/
OPAR; X2; +1.;0.75;0.001;0.001;0;-2.5;2.5/
OEND/

* CONTOH OPTIMASI - 1 *

JENIS PROBLEM :

PROBL. MINIMUM
KONTINUE
DENGAN RESTRIKSI

PARAMETER

JUMLAH : 2

NAMA	TIPE	DAERAH	HRG.START	STD-DEVIASI	RESTRIKSI-EXPL. BTS BWH	BTS ATS
X1	0		1.00000	0.750000	-2.5000	+2.5000
X2	0		1.00000	0.750000	-2.5000	+2.5000

RESTRIKSI

JUML. EXPLISIT : 2
JUML. IMPLISIT : 0

KTRL. VARIABEL

STRATEGI YANG DIPILIH	STRAT	= COMP
JUMLAH TITIK SIMPLEX	K	= 3
JUMLAH SIMULASI MAXIMAL	SIMMAX	= 500
SIGNIFIKANZNIVEAU	BETA	= 0.10000E-04
NMR RANDOM GENERATOR UNT. OPTIMASI	ORAND	= 30
PERMIN. START BARU	NEUSTA	= 1
PRINT-CONTROL UNTUK OPTIMASI	OPTPRT	= 0
PROTIKOL-CONTROL UNTUK SIMULASI	SIMPRT	= 0
ALPHA		= 1.30
GAMMA		= 5
EPSILON		= 5
DELTA		= 0.10E-05
LAMDA		= 0.20

HASIL OPTIMASI

=====

JUMLAH SIMULASI = 86
ITERASI = 41
FS SAS. = 2.04439
X1 = -1.62895
X2 = -0.564763

BERHENTI KRN
KRIT. BERHENTI DIPENUHI

Contoh 2 (seperti contoh 1, tetapi optimasinya dilakukan dengan menggunakan strategi Evolusi)

Seperti yang terlihat pada hasil optimasi dari contoh 1 di depan, dengan titik start $(X1, X2) = (+1.0, +1.0)$ strategi Complex tidak bisa menemukan optimum global, dia hanya menemukan optimum lokal saja. Sebaliknya dengan contoh 2 yang akan kita coba berikut ini, strategi Evolusi dapat menemukan optimum global yang dicari, hanya saja penghitungan fungsi sasaran yang diperlukan jauh lebih banyak (874) dari pada yang diperlukan oleh strategi Complex (86)

Untuk optimasi problem pada contoh 1 dengan menggunakan strategi Evolusi, kita hanya perlu merubah input-file "DATAIN" (SUB MENU : DATAIN Management - A - DATAIN). Bentuk dari file DATAIN sekarang seperti berikut :

```
TEXT; CONTOH OPTIMASI 2/
STRA; EVOL/
EVOL; MIN;10;100;0.3;0.7;0.01;2.0;4.5;1.E-5;1.E-
5;2000; 30;0; 0/
OPAR; X1; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OPAR; X2; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OEND/
```

Data yang dicetak tebal adalah data baru untuk strategi Evolusi (bandingkan dengan input data untuk strategi Complex pada contoh 1). Data tersebut mempunyai arti seperti berikut :

- **EVOL**
 Sesudah nama-perintah EVOL diikuti kontrol parameter untuk strategi Evolusi.

EVOL	Nama perintah
MIN	Problem minimum (ART) (MAX untuk problem maximum)
10	Banyaknya titik yang dipilih dari keturunan dalam satu generasi dan kemudian dipasang sebagai induk untuk generasi berikutnya (ELTERN)
100	Banyaknya titik keturunan yang dihasilkan dari induk
0.3	Parameter untuk mengontrol kecepatan penyesuaian Standard deviasi (SN)
0.7	Parameter untuk mengatur penyesuaian dari SN (SNN)
0.001	Batas bawah interval untuk harga dari SN (SNMIN)
2.0	Batas atas interval untuk harga dari SN (SNMAX)
4.5	Standard-deviasi untuk Random-Generator.
1.E-5	Parameter untuk kriteria berhenti (E_c)
1.E-5	Parameter untuk kriteria berhenti (E_d)
2000	Banyaknya penghitungan fungsi sasaran yang diijinkan (SIMMAX)

30;0;0/ lihat ORAND, OPTPRT dan SIMPRT pada contoh 1.

Setelah kita selesai merubah input data "DATAIN" ini, maka kita sekarang dapat melakukan optimasi dengan menggunakan strategi Evolusi (SUB MENU : PROGRAM Management - D - Run PROGRAM). Dengan data input seperti yang kita tuliskan diatas strategi Evolusi memberikan hasil seperti berikut :

```
JUMLAH SIMULASI = 874
X1 = -0.897461E-01
X2 = 0.712407
Q(X1,X2) = -1.03613
```

Contoh 3 (seperti contoh 1, tetapi optimasi dilakukan dengan menggunakan strategi COMBI)

Untuk optimasi problem pada contoh 1 dengan menggunakan strategi optimasi Combi kita tinggal merubah sedikit saja input-file "DATAIN". Bentuknya sekarang menjadi seperti berikut :

```
TEXT; CONTOH OPTIMASI 3/
STRA; COMB/
COMB; MIN; 0; 500; 1.E-5; 3; 30; 1; 0; 0;1;5/
OPAR; X1; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OPAR; X2; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OEND/
```

Data yang dicetak tebal adalah data baru untuk strategi Combi (bandingkan dengan input data pada contoh 1). Input data yang baru tersebut mempunyai arti seperti berikut :

COMB
 Sesudah nama perintah COMB diikuti oleh kontrol parameter untuk strategi Combi.

COMB Nama perintah
 (MIN; 0;....;0;0) artinya sama dengan yang tertera pada contoh 1 untuk strategi Complex.

- 1** Jumlah pengetesan optimum yang diinginkan (MAXTES)
- 5** Jumlah maximal untuk generasi yang diperbolehkan dalam phase evolusi (MAXGEN)

setelah kita selesai merubah input-file ini, kita dapat menjalankan optimasi dengan menggunakan strategi Combi (SUB MENU : PROGRAM Management - D - Run PROGRAM). Strategi Combi memberikan hasil seperti berikut :

```
JUMLAH SIMULASI = 197
X1 = 0.898466E-01
X2 = -0.713016
Q(X1,X2) = -1.03613
```

Catatan :

Seperti pada optimasi dengan menggunakan strategi Evolusi, strategi Combi juga bisa menemukan optimum global yang dicari. Bedanya, strategi Combi hanya membutuhkan penghitungan fungsi sasaran (simulasi) yang jauh lebih sedikit (197) dari pada yang dibutuhkan oleh strategi Evolusi (874). Disamping itu, strategi Combi juga membutuhkan lebih sedikit parameter kontrol dari pada strategi Evolusi.

Contoh 4 (seperti contoh 1, tetapi disini masih ditambah dengan restriksi implisit)

Seperti contoh 1, tetapi ditambah restriksi implisit seperti berikut : $X1 - X2 \leq 0$ ($X1$ harus lebih kecil dari pada $X2$)

Pemberitahuan :

Seperti sudah disinggung pada contoh 1 di muka, problem optimasi ini mempunyai dua optima global :

- $Q(X1,X2) = Q(+0.089,-0.713) = -1.03163$
- $Q(X1,X2) = Q(-0.089,+0.713) = -1.03613$

Strategi Combi tanpa restriksi implisit menemukan titik $(X1,X2) = (+0.089,-0.713)$ dengan $Q(+0.089,-0.713) = -1.03613$ sebagai penyelesaian yang optimal ($X1 > X2$). Sekarang dengan ditambahkan restriksi implisit, maka strategi Combi harus bisa menemukan titik optimum yang lain.

Implementasi :

Dengan menekan tombol B pada SUB MENU : RESTRIKSI, kita berada pada posisi editing untuk restriksi implisit. Pemasangan restriksi implisit mempunyai bentuk seperti berikut :

IF(NEGASI DARI RESTRIKSI) FLAG = .FALSE.

Pada "Pemasangan Variabel Optimasi" (lihat langkah b pada contoh 1 di muka) kita pasang $X1 = COMVEC(1)$ dan $X2 = COMVEC(2)$. Sedangkan negasi dari restriksi $X1 - X2 \leq 0$ adalah $X1 - X2 > 0$. Oleh karena itu restriksi implisit yang kita punyai ini sekarang mempunyai bentuk seperti berikut :

```

C   PERMULAAN PEMASANGAN RESTRIKSI
C   IMPLISIT
C   =====
      IF((COMVEC(1) - COMVEC(2)) .GT. 0.0)
      FLAG = .FALSE.
C   =====
C   AKHIR DARI PEMASANGAN RESTRIKSI
C   IMPLISIT

```

Untuk menyimpan pemasangan restriksi ini dan kembali ke MENU UTAMA tekan berturut-turut ESC, F, S dan ESC, Q, Q..

Sekarang program harus kita terjemahkan (compail) dan kita link lagi (SUB PROGRAM : PROGRAM Management - A - Compile PROGRAM dan C- Link PROGRAM) (lihat langkah f. pada contoh 1). Sebelum kita menjalankan program, terlebih dahulu kita harus memberitahukan tentang keberadaan restriksi implisit pada input-file "DATAIN". Input-file tersebut sekarang mempunyai bentuk seperti berikut :

```

TEXT; CONTOH OPTIMASI 4/
STRA; COMB/
COMB; MIN; 1; 500; 1.E-5; 3; 30; 1; 0; 0;1;5/
OPAR; X1; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OPAR; X2; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OEND/

```

Dibanding dengan input-file yang kita pakai pada contoh 3 hanya terdapat perbedaan pada kelompok data COMB elemen ketiga dari 0 menjadi 1 (yang dicetak tebal). Setelah pemasangan restriksi implisit ini kita beritahukan, kita dapat menjalankan program dengan menekan tombol D - Run PROGRAM pada SUB PROGRAM : PROGRAM Management. Strategi Combi memberikan hasil seperti berikut :

JUMLAH SIMULASI	= 153
X1	= -1.194839
X2	= +0.780982
Q(X1,X2)	= 0.505845

Seperti yang telah disinggung di depan, problem optimasi ini juga mempunyai beberapa optima lokal. Pada contoh 4 diatas, strategi Combi hanya menemukan optimum lokal saja. Akan tetapi strategi Combi dapat kita perintahkan untuk melanjutkan pencarian optimum dengan cara menambah harga parameter MAXTES dari 1 menjadi 2. Dengan penambahan dari harga MAXTES tersebut maka kelompok data COMB sekarang mempunyai bentuk seperti berikut :

```

COMB; MIN;1;500;1.E-5;3;30;1;0;2;5/

```

Dengan input data ini strategi Combi memberikan hasil seperti berikut :

JUMLAH SIMULASI	= 292
X1	= -0.895259E-01
X2	= +0.712653
Q(X1,X2)	= -1.03613

Dengan perintah tambahan tadi strategi Combi bisa menemukan optimum global yang dicari. Disini juga terlihat bahwa optimum yang ditemukan oleh strategi Combi adalah optimum yang lain dari pada yang ditemukan pada contoh 4, yaitu optimum yang

memenuhi restriksi implisit $X1 - X2 \leq 0$. Pada prinsipnya kita bisa memerintahkan kepada strategi Combi untuk melanjutkan pencarian optimum seperti yang kita kehendaki, yaitu dengan cara menambah harga MAXTES.

Pemberitahuan :

Ada kemungkinan untuk menggabungkan input-file "DATAIN" yang terdapat pada contoh 1, 2 dan 3 kedalam satu input-file yang kompak seperti berikut :

```
TEXT; CONTOH OPTIMASI 1,2,3/
STRA; — /
COMP; MIN; 0; 500; 1.E-5; 3; 30; 1; 0; 0/
EVOL; MIN; 10; 100; 0.3; 0.7; 0.01; 2.0; 4.5; 1.E-5; 1.E-5; 2000; 30; 0; 0/
COMB; MIN; 1; 500; 1.E-5; 3; 30; 1; 0; 0; 1; 5/
OPAR; X1; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OPAR; X2; +1.; 0.75; 0.001; 0.001; 0; -2.5; +2.5/
OEND/
```

Untuk :

STRA; COMP/ => optimasi akan dilakukan dengan menggunakan strategi Complex. Input-data untuk strategi Evolusi dan Combi akan dieliminir secara otomatis.

STRA; EVOL/ => optimasi akan dilakukan dengan menggunakan strategi Evolusi. Input-data untuk strategi Complex dan Combi akan dieliminir secara otomatis.

STRA; COMB/ => optimasi akan dilakukan dengan menggunakan strategi Combi. Input-data untuk strategi Complex dan Evolusi akan dieliminir secara otomatis.

Dengan input-file dalam bentuk kompak seperti diatas memungkinkan bagi kita untuk men-switch strategi secara mudah.

Jika daerah definisi dari parameter optimasi merupakan harga-harga dari suatu tabel, maka penyajiannya harus dilakukan dengan menggunakan kelompok data ODIS yang mempunyai format seperti berikut :

ODIS; NAMA; HARGA/

ODIS	Nama perintah	* * *
NAMA	Sesuai dengan nama dari parameter optimasi yang dimaksud	
HARGA	Disusun dalam bentuk deret dengan harga yang meningkat	

5 SARAN-SARAN

Dewasa ini harga dari sebuah Personal Computer (PC) bisa dikatakan sudah relatif murah. Karena itu tidak mengherankan jika hampir setiap peneliti, baik dilingkungan Lembaga Penelitian maupun Perguruan Tinggi, telah memiliki komputer untuk menunjang kegiatan penelitian yang dilakukan. Atas pertimbangan tersebut, Kelompok Peneliti Simulasi dan Optimasi - Bidang Teknologi Transmisi Komunikasi Dirgantara - LAPAN mencoba mengembangkan sebuah Paket Program Optimasi dengan menggunakan teknik *on-line-menu* yang bisa dijalankan pada sebuah PC. Penggunaan teknik tersebut dimaksudkan agar rekan-rekan peneliti yang tidak menguasai bahasa pemrograman pun dapat mengoperasikan paket program yang ditawarkan. Seperti yang telah disebutkan di depan, paket program ini khusus digunakan untuk optimasi model-model statik. Sedang untuk optimasi model-model dinamik juga telah disediakan sistem yang lain, yaitu Sistem Optimasi Model-Model Kompleks (SIMPLEX), yang pembahasannya disajikan dalam makalah [HARD94-4].

Dengan adanya sistem-sistem optimasi yang disebutkan di atas, penulis menyarankan kepada rekan-rekan peneliti di LAPAN pada khususnya dan di lembaga lain pada umumnya untuk ikut memanfaatkan fasilitas dan kemampuan yang ada tersebut guna menunjang penelitian yang dilakukan. Kerjasama dapat dilakukan dalam bentuk analisa sistem, pembuatan model, simulasi dan optimasi secara bersama-sama maupun dalam bentuk instalasi software ditempat rekan-rekan bekerja. Karena instalasi dan pengoperasian program tidak begitu sukar, maka rekan-rekan peneliti juga bisa melakukannya sendiri. Untuk keperluan konsultasi ataupun untuk mendapatkan software beserta lisensinya bisa dihubungi :

**Kelompok Peneliti Simulasi dan Optimasi
Bidang Teknologi Transmisi Komunikasi
Dirgantara, LAPAN
PO BOX 13 Semplak - Bogor 16310**

**Telepon (0251) - 21667
Fax (0251) - 23010
E-Mail : hardh@simulasi.lrb.lapan.go.id**

LITERATUR :

- [AOKI71] : Aoki, M.
Introduction to optimization techniques
fundamentals and applications of nonlinear
programming ; Macmillan, New York, 1971
- [APPE88] : Appelbaum, K.
Numerische Optimierung von Simulations-
modellen mit Hilfe der Evolutionsstrategie
Skripsi dalam Bidang Informatik- IMMD-IV
Universitas Erlangen-Nuernberg, 1988
- [AZAD79] : Azadivar, F.
Optimization of stochastic systems through
simulation using stochastic approximation
method
Desertasi, Purdue University, 1979.
- [BIET78] : Biethahn, J.
Optimierung und Simulation
Betriebswirtschaftlicher Verlag,
Wiesbaden, 1978
- [BOX65] : A new method of constrained optimization
and comparison with other methods
Computer Journal 8, 1965
- [BRAN72] : Branin, F.H. Jr., Hoo, S.K.
A method for finding multiple extrema of a
function of n variables
Lootsma, 1972
- [BREM70] : Bremermann, H.J.
A method of unconstrained global
optimization
Math. Biosci. 9, 1970
- [BREN73] : Brent, P.R.
Algorithms for minimization without
derivatives
Prentice-Hall, Englewood Cliffs, 1973
- [DIXO72] : Dixon, L.C.W.
Nonlinear optimization
Crane, Russak, 1972
- [DOER85] : Numerische Optimierung mit kombinierten
Simulationsmodellen
Skripsi dalam bidang Informatik IMMD-IV
Universitas Erlangen - Nuernberg, 1985
- [HAMO90] : Hardhienata, S. ; Molnar, I.
Comparison of optimization strategies for
combined simulation
Proceedings of the 1990 European
Simulation Multiconference
Nuremberg, 1990
- [HAMO92] : Hardhienata, S. ; Molnar, I.
Optimierung mit Hilfe von Simulations-
modellen Simulation als betriebliche
Entscheidungshilfe - Band 3
Springer Verlag, 1992
- [HARD93] : Numerische Optimierungsstrategi fuer
Simulationsmodelle mit Anwendungen in
Informatik und Verfahrenstechnik
Disertasi, IMMD-IV Universitas Erlangen-
Nuernberg, 1993
- [HARD94-4] : Hardhienata, S.
Optimasi Model-Model Dinamik dengan
Sistem Optimasi SIMPLEX
Laporan Intern Bidang Transmisi, 1994
- [HIME72] : Himmelblau, D.M.
Applied Nonlinear Programming
Mc graw-Hill, 1972
- [HORS79] : Horst, R.
Nichtlineare Optimierung
Carl Hanser Verlag, 1979
- [KERC84] : Kerchel-Mohr, K.J.
Ein universelles Verfahren zur loesung von
Optimierungsproblemen
Studienarbeit, Universitas Darmstad, 1984
- [KERC85] : Kerchel-Mohr, K.J. ; Molnar, I.
Ein universelles Optimierungsmodul zur
Lösung von Entscheidungs-problemen in
der Simulation
Skripsi, Universitas Darmstad, 1985
- [PRON84] : Pronzato, L.
A general-purpose global optimizer :
implementation and applications
Mathematics and Computer in Simulation
XXIV, 1984
- [SCHW77] : Schwefel, H.-P.
Numerische Optimierung von Computer-
Modellen mittels der Evolutionsstrategie
Birkhaeuser verlag, Basel und Stuttgart,
1977
- [WERN84] : Werner, G.
Optimization Theory and Applications
Friedr. Vieweg & Sohn,
Braunsweig/Wiesbaden, 1984