LUCRAREA 4

PROGRAMAREA NELINIARĂ. METODE DE ORDINUL 0

4.1. Aspecte generale

Programarea neliniară are o foarte mare importanţă în rezolvarea problemelor de optimizări, în general, şi în energetică, în particular, deoarece, destul de frecvent, metodele corespunzătoare acesteia sunt singurele mijloace de soluţionare.

În această clasă intră acele probleme în care funcţia obiectiv şi (sau) restricţiile sunt expresii matematice neliniare. Această neliniaritate provine dintr-o considerare mai riguroasă a proceselor reale, dar, din punct de vedere tehnic, creează dificultăţi în găsirea soluţiei optimale. Pentru cazul particular, funcţia obiectiv neliniară şi restricţiile liniare, se poate aplica metoda gradienţilor proiectaţi.

Dificultăţile rezolvării problemelor de programare neliniară se datorează faptului că aici nu se mai întâlnesc facilităţile din programarea liniară (optimul este atins într-un vârf al poliedrului convex), iar, pe de altă parte, este dificil să se facă diferenţa dintre optimul local şi optimul global. Există o clasă de probleme pentru care optimul local este şi global. Aceasta se numeşte programarea convexă. Importanţa acestei categorii de probleme rezidă din faptul că dispune de metode eficiente de rezolvare şi, în plus, există numeroase exemple de aplicaţii care se încadrează în această clasă.

Problema generală de programare matematică constă în găsirea extremului unei funcţii de mai multe variabile:

$$\textbf{FO:} \qquad F(X) = F(X_1, X_2, ..., X_n) \tag{4.1}$$

$$\textbf{RE:}\quad \begin{array}{l} g_i(X_1, X_2, ..., X_n) \le b_i\,, \quad i = \overline{1, m} \\ X_j \ge 0, \quad j = \overline{1, n} \end{array} \tag{4.2}$$

Utilizînd notaţia:

$$X^k = [X_1^k, X_2^k, ..., X_n^k]_t\,, \tag{4.3}$$

funcţia $F(X)$ este convexă dacă verifică relaţia:

$$F\left(\lambda X^1 + (1-\lambda)X^2\right) \le \lambda F(X^1) + (1-\lambda)f(X^2)\;; \quad \lambda \in [0,1] \tag{4.4}$$

unde X^1 şi X^2 sunt două puncte oarecare din domeniul de definiţie al funcţiei, presupus convex.

Problema de programare matematică, pentru care domeniul de soluţii descris de restricţiile (4.2) este convex, iar funcţia obiectiv (4.1) pentru acelaşi domeniu este convexă, poartă denumirea de problemă de programare convexă.

În general, pentru rezolvarea unei probleme de programare neliniară se foloseşte relaţia iterativă:

$$X^{(k+1)} = X^{(k)} + \lambda^{(k)} d^{(k)}\,, \quad k = 1,\ 2,\ 3, ... \tag{4.5}$$

unde:

$d^{(k)} \in R^n$ - reprezintă o direcţie de deplasare în iteraţia (k) din punctul curent $X^{(k)}$;

$\lambda^{(k)}$ – scalar ce reprezintă lungimea pasului de deplasare în iteraţia curentă.

Metodele de optimizare diferă prin procedurile concrete de alegere a parametrilor $d^{(k)}$ şi $\lambda^{(k)}$. Având în vedere reducerea efortului de calcul privind implementarea procedurii iterative, în fiecare etapă, informaţia disponibilă pentru obţinerea direcţiei $d^{(k)}$ şi a pasului $\lambda^{(k)}$ este strict limitată la valorile funcţiei şi a primei ei derivate.

După tipul de informaţii folosite distingem următoarele grupuri de metode:

- **metode de ordinul 0** care folosesc valorile funcţiei în punctul curent şi vecinătăţi;
- **metode de ordinul 1** care folosesc şi derivata de ordinul 1 (gradientul);
- **metode de ordinul 2** care folosesc derivata de ordinul 1 şi derivata de ordinul 2 (hessianul).

4.2. Metode de ordinul 0

Faptul că nu sunt necesare expresiile analitice ale derivatei funcţiei obiectiv $F(X)$ constituie un avantaj major al acestor metode. Ca urmare nu există condiţii referitoare la continuitatea şi derivabilitatea funcţiei $F(X)$. Trebuie remarcat că există totuşi şi un dezavantaj al acestor metode legat de faptul că au o viteză de convergenţă scăzută.

Metodele de ordin 0 au la bază relaţia iterativă (4.5), cu menţiunea că stabilirea noii direcţii de explorare se face pe baza unei serii de evaluări a funcţiei obiectiv $F(X)$, într-o manieră specifică fiecărei metode, iar lungimea paşilor $\lambda^{(k)}$ este, deasemenea, proprie fiecărei metode.

Faptul că nu sunt utilizate valorile derivatelor prezintă avantajul suplimentar că, printr-o alegere corespunzătoare a criteriilor de stop, se poate evita terminarea iteraţiilor într-un punct de inflexiune. În literatură sunt prezentate o varietate de metode de optimizare fără evaluarea derivatelor, precum şi foarte multe variante ale acestora. În continuare se prezintă metoda optimizării ciclice de-a lungul axelor de coordonate, care este cea mai uşor de aplicat.

Optimizarea ciclică de-a lungul axelor de coordonate

Descrierea metodei este sugerată de însăşi denumirea sa, respectiv se face o explorare unidimensională pe direcţii care coincid cu axele de coordonate. Astfel, în relaţia (4.5) direcţiile $d^{(k)}$ sunt succesiv axele sistemului ortogonal de coordonate:

$$d^{(0)}=[1 \quad 0 \quad 0 \quad \ldots \quad 0]^{t}, d^{(1)}=[0 \quad 1 \quad 0 \quad \ldots \quad 0]^{t}, \ldots, d^{(n-1)}=[0 \quad 0 \quad 0 \quad \ldots \quad 1]^{t} \qquad (4.6)$$

Paşii $\lambda^{(k)}$ pe direcţiile respective pot fi determinaţi cu următoarea relație:

$$\lambda^{(k)}=-\frac{\left(g^{(k)}, d^{(k)}\right)}{\left(d^{(k)}, H^{(k)} d^{(k)}\right)}, \quad \left(d^{(k)}, H^{(k)} d^{(k)}\right)>0 \qquad (4.7)$$

unde:

$H^{(k)}$ – matricea hessiană într-un punct curent $X^{(k)} \in R^{n}$;

$g^{(k)}$ – gradientul funcţiei obiectiv într-un punct curent $X^{(k)} \in R^{n}$.

Procesul iterativ se va încheia când va fi îndeplinit criteriul de stop impus. Potrivit acestei metode optimul este atins în n iteraţii. Dacă suprafeţele respective prezintă o vale sau o creastă care nu este paralelă cu axele de coordonate, această metodă nu poate fi aplicată.

4.3. Exemplu numeric

Să se determine minimul funcţiei:

$$F(X) = \min(x_1^2 + 4 \cdot x_2^2 - 4)$$

folosind metoda de optimizare ciclică de-a lungul axelor de coordonate, indicându-se ca punct de pornire $X^{(0)} = [5 \quad 4]^t$.

Conform metodei valoarea optimă se va atinge în 2 iteraţii.

Iteraţia 1

Se face deplasarea pe direcţia $d^{(0)} = [1 \quad 0]^t$, astfel încât pentru determinarea unei noi aproximaţii se va utiliza relaţia iterativă:

$$X^{(1)} = X^{(0)} + \lambda^{(0)} d^{(0)} = \begin{bmatrix} 5 \\ 4 \end{bmatrix} + \lambda^{(0)} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 + \lambda^{(0)} \\ 4 \end{bmatrix}$$

Pentru determinarea lungimii pasului de deplasare $\lambda^{(0)}$, se introduc valorile $x_1^{(1)}$ şi $x_2^{(1)}$, în expresia funcţiei obiectiv F:

$$F(\lambda^{(0)}) = (5 + \lambda^{(0)})^2 + 4 \cdot 4^2 - 4 = \left(\lambda^{(0)}\right)^2 + 10 \cdot \lambda^{(0)} + 85$$

În continuare se face derivata expresiei $F(\lambda^{(0)})$ în raport cu $\lambda^{(0)}$ şi se anulează.

$$\frac{\partial F\left(\lambda^{(0)}\right)}{\delta \lambda^{(0)}} = 2 \cdot \lambda^{(0)} + 10 = 0 \quad \rightarrow \quad \lambda^{(0)} = -5$$

Valoarea pasului de deplasare se introduce în relaţia iterativă obţinându-se astfel valoarea noii aproximaţii $X^{(1)}$:

$$X^{(1)} = \left[x_1^{(1)} \quad x_2^{(1)}\right]_t = [0 \quad 4]^t .$$

Iteraţia 2

Se face deplasarea pe direcţia $d^{(1)} = [0 \quad 1]^t$, astfel încât pentru determinarea unei noi aproximaţii relaţia iterativă devine:

$$X^{(2)} = X^{(1)} + \lambda^{(1)} d^{(1)} = \begin{bmatrix} 0 \\ 4 \end{bmatrix} + \lambda^{(1)} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 + \lambda^{(1)} \end{bmatrix}$$

Pentru determinarea lungimii pasului de deplasare $\lambda^{(1)}$, se introduc valorile $x_1^{(2)}$ şi $x_2^{(2)}$, în expresia funcţiei obiectiv F:

$$F(\lambda^{(1)}) = 0 + 4 \cdot \left(4 + \lambda^{(1)}\right)^2 - 4 = \left(\lambda^{(1)}\right)^2 + 8 \cdot \lambda^{(1)} + 15$$

În continuare, se face derivata expresiei funcției $F(\lambda^{(1)})$ în raport cu $\lambda^{(1)}$ şi se anulează.

$$\frac{\partial F\left(\lambda^{(1)}\right)}{\delta \lambda^{(1)}} = 2 \cdot \lambda^{(1)} + 8 = 0 \quad \rightarrow \quad \lambda^{(0)} = -4$$

Valoarea pasului de deplasare se introduce în relaţia iterativă obţinându-se astfel valoarea aproximaţiei $X^{(2)}$:

$$X^{(2)} = \left[x_1^{(2)} \quad x_2^{(2)}\right]^t = \left[0 \quad 0\right]^t,$$

ceea ce era evident încă de la început.

4.4. Rezolvarea problemelor PN cu ajutorul funcţiilor MatLab

În general, o problemă de găsire a minimului unei funcţii neliniare în absenţa restricţiilor are următoarea formă:

$$\min_X F(X) \tag{4.4}$$

unde $F(X)$ este o funcţie neliniară ce returnează un scalar.

Sintaxa

X = **fminunc**(@*fun*, X0) (a)

X = **fminunc**(@*fun*, X0, *options*) (b)

[X, *val_funcţie*, *convergenta*, *informatii*] = **fminunc**(...) (c)

unde:

(a) procesul are ca punct de plecare punctul $X0$ şi găseşte minimul funcţiei descrisă în fişierul *fun*. $X0$ poate fi un vector, un scalar sau o matrice;

(b) minimizează funcţia descrisă în fişierul *fun*, cu parametrii de optimizare precizaţi în structura *options*;

(c) X – soluţia problemei;

val_functie – returnează valorile funcţiilor obiectiv corespunzătoare soluţiei găsite;

convergenta – dă informaţii cu privire la convergenţa procesului. Dacă *convergenta* = 1, funcţia converge la soluţia X, dacă *convergenta* = 0, numărul maxim de evaluări a funcţiei sau numărul maxim de iteraţii a fost depăşit, dacă *convergenta* = -1 procesul de optimizare este divergent;

informatii – furnizează informaţii despre procesul de optimizare (numărul de iteraţii, numărul de evaluări ale funcţiei, algoritmul folosit).

Parametrii de intrare şi ieşire

fun – fişier funcţie ce conţine expresiile funcţiilor obiectiv. Acest fişier are ca variabilă de intrare X, iar ca variabilă de ieşire un scalar F, a cărui valoare reprezintă funcţia obiectiv evaluată în punctul X. Fişierul *fun* poate fi apelat astfel:

X = **fminunc** (@*fun*, $X0$)
function F = **fun**(X)

Exemplu numeric

Să se minimizeze funcţia:

$$F(X) = 4 \cdot (x_1 - 2)^2 + (x_2 - 3)^2$$

cu ajutorul funcţiei Matlab **fminunc**, indicându-se ca punct de pornire $X^{(0)} = [\,2 \quad 3\,]^t$.

Se va construi fişierul **fun** în care se va introduce expresia funcţiei $F(X)$.

```
function F = fun(X)

F = 4*(X(1)-2)^2 + (X(2)-3)^2;
```

Cu secvenţa Matlab:

```
>> X0 = [2;3];
>> options = optimset('Display', 'iter');
>> [X,val_functie]=fminunc(@fun,X0,options)
```

se obţin următoarele rezultate:

```
 Iteration  Func-count     f(x)     Step-size     First-order
                                                   optimality
     0           3            64                        32
     1           6       39.5156     0.03125            24
     2           9       1.24443           1          2.23
     3          12      0.647399           1          1.65
     4          15   6.5037e-005           1        0.0321

Optimization terminated: relative infinity-norm of gradient
less than options.TolFun.
x =
    2.0000
    3.0000
fval =
    6.5037e-05
```

4.5. Desfăşurarea lucrării

1. Se studiază textul lucrării.
2. Să se determine minimul funcţiei:

$$F(X) = \min(x_1^2 + 2 \cdot x_2^2 - 1)$$

   folosind metoda de optimizare ciclică de-a lungul axelor de coordonate, indicându-se ca punct de pornire $X^{(0)} = [3 \quad 1]^t$. Rezultatele obținute vor fi comparate cu cele obținute prin folosirea funcției Matlab **fminunc**.
3. Să se determine minimul funcţiei:

$$F(X) = \min(x_1^2 + 3 \cdot x_2^2 - 2)$$

   folosind metoda de optimizare ciclică de-a lungul axelor de coordonate, indicându-se ca punct de pornire $X^{(0)} = [2 \quad 1]^t$. Rezultatele obținute vor fi comparate cu cele obținute prin folosirea funcției Matlab **fminunc**.