LUCRAREA 1

OPTIMIZAREA CU MATLAB

1. 1. Aspecte generale

Conducerea, în general, şi a unor procese, în particular, este reprezentată printr-un şir succesiv de decizii, iar acestea reprezintă alegeri între mai multe alternative disponibile, cu scopul atingerii unuia sau mai multor obiective. Fundamentarea ştiinţifică a deciziei, indiferent de situaţia implicată, poate fi considerată ca un proces general, sistematic, constând din mai multe trepte: definirea problemei, evidenţierea variantelor disponibile, evaluarea variantelor posibile din punct de vedere al consecinţelor, fundamentarea şi selectarea variantei optimale.

Modelele de optimizare iau forma unor ecuaţii, care deşi din punct de vedere matematic pot fi foarte complicate, au o exprimare formală foarte simplă:

$$\max\ (\min) F(X), \quad X = [x_1, x_2, \ldots, x_n]_t \tag{1.1}$$

unde:

F – reprezintă valoarea criteriului (utilitatea) ce caracterizează funcţionarea sistemului;

X – vectorul variabilelor controlate sau de optimizare (ale căror valori pot fi măsurate, modificate sau fixate arbitrar).

In plus, la modelul (1.1) se mai adaugă, de cele mai multe ori, una sau mai multe ecuaţii sau inecuaţii, care indică faptul că variabilele de optimizare trebuie să se încadreze între anumite limite. Restricţiile reprezintă relaţii de constrângere care trebuie satisfăcute pentru ca soluţia să fie acceptabilă practic. O restricţie de egalitate, exprimată implicit sau explicit are forma:

$$h_i(X) = 0, \quad i = 1, \ldots, m \tag{1.2}$$

Teoretic, fiecare restricţie de egalitate poate fi folosită pentru eliminarea unei variabile, lucru care uneori este foarte complicat sau chiar imposibil din punct de vedere algebric. O restricţie de inegalitate are forma:

$$g_i(X) \leq (\geq) 0, \quad i = 1, \ldots, p \tag{1.3}$$

Se poate menţiona faptul că restricţiile de inegalitate au o importanţă majoră în căutarea unei soluţii mai bune pentru o problemă de optimizare.

Funcţia F, numită şi funcţie obiectiv sau funcţie scop, formează împreună cu restricţiile modelul sistemului care urmează a fi optimizat. Acesta reprezintă atât un model al sistemului cât şi un model de decizie. Modelul respectiv poate fi folosit pentru a găsi exact sau aproximativ valorile optime ale variabilelor controlabile, adică acele valori care asigură cea mai bună performanţă a sistemului.

Teoria optimizării s-a bazat, într-o primă etapă, pe dezvoltările matematice ale secolelor precedente, în care "lumea" era curată, alcătuită din funcţii pătratice, cu restricţii ideale şi derivabilitate omniprezentă. Dar "lumea" reală este plină de discontinuităţi, spaţii şi soluţii perturbate, care se pretează cu dificultate proceselor iterative de determinare a soluţiilor perturbate, care se pretează cu dificultate proceselor iterative de determinare a soluţiilor optimale. De aceea optimizarea bazată doar pe calcule matematice nu poate fi întotdeauna aplicată. Metodele principale folosite în cazul problemelor de optimizare se pot clasifica, pe baza tehnicilor de programare folosite, astfel:

- Metode bazate pe programarea liniară;
- Metode bazate pe programarea neliniară;
- Metode bazate pe programarea dinamică.

1.2. Optimizarea cu MatLab

Produsul MatLab este compus dintr-o serie de programe standard scrise pentru calcule matematice, modelare şi simulare numerică, prelucrări de date statistice, reprezentări grafice asistate de calculator.

Cea mai importantă caracteristică a MatLab-ului este uşurinţa cu care poate fi extins. Prin aceasta, orice utilizator poate adăuga, propriile programe scrise în MatLab, la fişierele originale, dezvoltand aplicaţii specifice domeniului în care lucrează.

De asemenea, MatLab-ul include aplicaţii specifice, numite Toolbox-uri. Acestea sunt colecţii extinse de funcţii MatLab (fişiere *.m) care dezvoltă mediul de programare de la o versiune la alta, pentru a rezolva probleme din domenii variate. Structural, MatLab-ul este organizat sub forma unui nucleu de bază, cu interpretor propriu, în jurul căruia sunt construite toolbox-urile.

Din cadrul acestui grup de programe face parte şi toolbox-ul Optim. Optim este o colecţie de funcţii utilizată pentru optimizarea liniară şi neliniară, ce grupează următoarele tipuri de probleme:

- programare liniară şi pătratică;
- determinarea minimului şi maximului;
- funcţii neliniare rezolvate în sensul celor mai mici pătrate;
- rezolvarea ecuaţiilor neliniare;
- rezolvarea problemelor de minimax şi semi-infinite;
- optimizarea multiobiectiv.

Fiecare componentă a setului de programe Optim este o funcţie MatLab (fişier *.m). Acestea sunt:

1. programe pentru minimizarea funcţiilor neliniare:

   - **fgoalattain** - probleme multi-obiectiv;
   - **fmincon** - minimizare cu restricţii;
   - **fminbnd** - minimizare fără restricţii, cazul unidimensional;
   - **fminunc** - minimizarea fără restricţii cu metode de tip gradient;
   - **fminsearch** - rezolvarea ecuaţiilor neliniare;

- **lsqnonlin** - aplicarea metodei celor mai mici pătrate;
- **fminimax** - căutarea soluţiei minimax;
- **fseminf** - minimizarea semi-inf.

2. programe pentru probleme de minim date matricial:

- **linprog** - programare liniară;
- **quadprog** - programare pătratică;
- **lsqnonneg** - aplicarea metodei celor mai mici pătrate în condiţii de nonnegativitate.

Setarea opţiunilor implicite de lucru pentru MatLab, precum şi stabilirea configuraţiei dorite de utilizator se realizează cu funcţia optimset. Funcţia optimset returnează vectorul options. Componentele acestuia sunt folosite de funcţiile de mai sus în totalitate sau parţial. In cele ce urmează se vor prezenta cele mai utilizate componente ale vectorului options.

- **Display** - dacă are valoarea ”iter” atunci se afişează valorile de ieşire la fiecare iteraţie; dacă are valore ”final” se vor afişa rezultatele procesului de optimizare din ultima iteraţie;
- **MaxFunEvals** - numărul de evaluări al funcţiei obiectiv;
- **MaxIter** - numărul maxim de iteraţii;
- **TolFun** - toleranţa de oprire pentru valoarea funcţiei obiectiv;
- **DiffMaxChange** - valoarea maximă de modificare pentru variabile în metoda diferenţelor finite;
- **DiffMinChange** - valoarea maximă de modificare pentru variabile în metoda diferenţelor finite;
- **GradObj** - gradientul funcţiei obiectiv definit de utilizator;
- **GradConstr** - gradienţii pentru restricţiile neliniare definiţi de utilizator;
- **Hessian** - Hessianul funcţiei obiectiv definit de utilizator;
- **LineSearchType** - indică folosirea algoritmului liniar de căutare;
- **TolCon** - toleranţa de oprire la încălcarea restricţiilor.

Exemplu: options = optimset('Display','iter','TolFun',1e-8)

Funcţia optimset returnează vectorul options în care parametrul Display este setat pe valoarea 'iter', iar parametrul TolFun este setat pe valoarea 1e-8.

1.3. Desfăşurarea lucrării

1. Se studiază textul lucrării.
2. Se studiază sintaxele funcțiilor Matlab corespunzătoare rezolvării problemelor de programare liniară. Se vor identifica variabilele de intrare si variabilele de iesire si structura acestora.
3. Se studiază funcțiile Matlab corespunzătoare rezolvării problemelor de programare neliniară. Se vor identifica variabilele de intrare și variabilele de iesire, respectiv structura acestora.
4. Se vor identifica opţiunile implicite de lucru pentru MatLab, precum şi stabilirea configuraţiei dorite de utilizator.