Capitolul 1

Funcţii ale pachetului de optimizare MATLAB

1.1 Preliminarii

Explozia informaţională din ultimele patru decenii a condus la algoritmi numerici de optimizare indispensabili pentru abordarea diferitelor probleme din domeniile ingineriei, fizicii, analizei numerice, teoriei sistemelor şi controlului, prelucrării semnalelor, învăţării automate etc. Precizând că această scurtă enumerare reprezintă doar cele mai evidente domenii de aplicaţie ale algoritmilor de optimizare, constatăm că teoria optimizării este considerată un instrument valoros în majoritatea domeniilor ştiinţifice.

În prezent există algoritmi eficienţi (rapizi) destinaţi rezolvării problemelor de optimizare convexe/neconvexe cu dimensiuni mici şi medii într-un interval de timp relativ scurt. Însă multe dificultăţi şi întrebări fără răspuns apar în cazul problemelor cu dimensiuni mari (e.g. $10^7 - 10^9$ variabile). Dimensiunile problemelor întâlnite în domeniile enumerate anterior se află în creştere continuă, astfel încât se formulează relativ des probleme cu dimensiuni de zeci de milioane de variabile. Ca urmare, algoritmii de optimizare rămân un subiect de actualitate în cercetarea modernă din matematica aplicată.

În continuare vom prezenta şi formula principalele clase de probleme de optimizare şi în plus, funcţiile Matlab uzuale pentru rezolvarea numerică a acestor tipuri de probleme.

1.1.1 Mediul Matlab

Matlab (**Mat**rix **lab**oratory) este un mediu de calcul numeric şi un limbaj de programare de generaţia a patra, dezvoltat de compania *Mathworks*. Mediul Matlab oferă o interfaţă uşor de utilizat pentru operaţii cu matrice, implementare de algoritmi, grafice de funcţii etc.

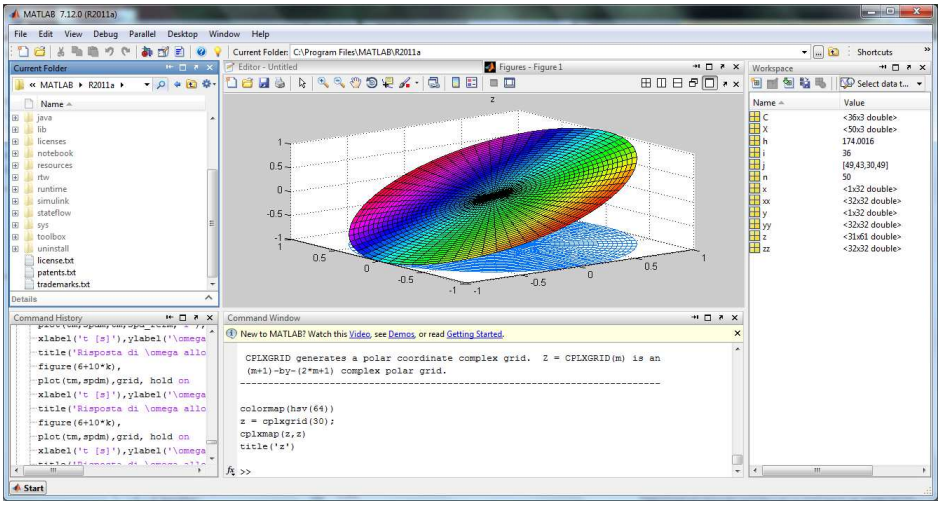

Figura 1.1: *Matlab.*

Încă de la sfârşitul anilor 1970, Matlab a câştigat o audienţă largă în cadrul academic şi în comunitatea matematicii aplicate. Începând cu dezvoltarea de pachete de programe destinate problemelor numerice de algebră liniară, au fost incluse şi dezvoltate ulterior funcţii (toolbox-uri) specializate în rezolvarea de probleme de optimizare, rezolvarea de sisteme de ecuaţii, modelare şi identificare de sisteme, control etc.
Pachetul *Optimization Toolbox*, din cadrul mediului Matlab, furnizează algoritmi utilizaţi la scară largă, destinaţi problemelor de optimizare standard şi de mari dimensiuni. De asemenea, pachetul include şi funcţii ce rezolvă probleme de optimizare structurate: programare liniară, programare pătratică, programare binară, probleme CMMP neliniare, rezolvare de sisteme de ecuaţii neliniare şi optimizare multicriterială. Dificultatea problemelor de optimizare ce apar de cele mai multe ori în practică limitează capacitatea algoritmilor existenţi la găsirea unei soluţii aproximative corespunzătoare problemei date. De exemplu,

problema de optimizare ce presupune găsirea punctului global de maxim corespunzător funcţiei $\text{sinc}(x) = \frac{\sin \pi x}{\pi x}$ este o problemă dificil de rezolvat. Pentru motivarea acestei afirmaţii se poate observa uşor în Fig. 1.2 că funcţia $\text{sinc}(x)$ prezintă o infinitate de maxime locale (denivelările de la baza dealului), despre care nu cunoaştem *a priori* dacă sunt sau nu valori maxime globale. Pentru a ilustra simplitatea cu care putem figura funcţiile în mediul Matlab, prezentăm în continuare secvenţa de cod care generează graficul din Fig. 1.2.

```
[X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);
f = sinc(sqrt((X/pi).^2+(Y/pi).^2));
surf(X,Y,f);
axis([-10 10 -10 10 -0.3 1])
xlabel('{\bfx}')
ylabel('{\bfy}')
zlabel('{\bfsinc} ({\bfR})')
```

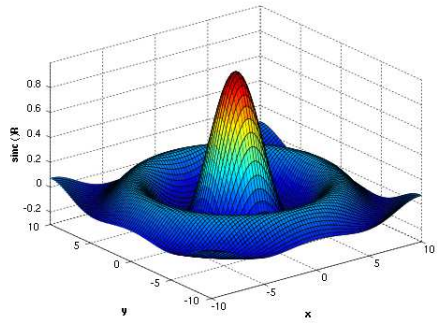

Figura 1.2: *Graficul funcţiei sinc*(x) .

1.2 Probleme rezolvate de laborator

1.2.1 Minimizare neconstrânsă

Problemele de optimizare neconstrânsă presupun minimizarea (sau maximizarea) unei anumite funcţii obiectiv (criteriu) prin intermediul unui set de variabile de decizie, fără ca acestea să respecte anumite restricţii (constrângeri). Se consideră funcţia (în general, neliniară)

$f : \mathbb{R}^n \to \mathbb{R}$ de două ori diferenţiabilă, căreia i se asociază problema de optimizare:

$$\min_{x \in \mathbb{R}^n} f(x),$$

unde x are rolul vectorului *variabilelor de decizie*. Problema de optimizare se consideră rezolvată dacă s-a obţinut un vector x^* pentru care valoarea funcţiei $f(\cdot)$ în x^* este minimă, adică $f(x^*) \leq f(x)$ pentru orice $x \in \mathbb{R}^n$. Mulţimea în care se efectuează căutarea se numeşte *mulţime fezabilă*, care în cazul de faţă este dată de întreg spaţiul Euclidian $\mathbb{R}^n$.

Funcţii MATLAB

Pentru cazul în care funcţia obiectiv nu are o structură prestabilită sau o formă particulară, sunt disponibile următoarele funcţii MATLAB pentru aproximarea unui punct de minim local în cazul unei probleme neconstrânse:

1. Funcţia `[...]=fminunc(...)` returnează un punct de minim local al funcţiei introduse ca argument, cu ajutorul informaţiei de ordinul I (gradient) şi a celei de ordin II (Hessiana). Sintaxa funcţiei are expresia:

```
x=fminunc(fun,x0)
x=fminunc(fun,x0,optiuni)
[x,val]=fminunc(...)
[x,fval,exitflag,output,grad,hessian]=fminunc(...)
```

Variabila `fun` reprezintă funcţia obiectiv diferenţiabilă ce trebuie furnizată ca o variabilă de tip *function handle*. O variabilă de tip *function handle* poate fi, e.g. un fişier `objfun.m` care primeşte la intrare variabila de decizie x şi returnează valoarea funcţiei în x. Dacă metoda de rezolvare a problemei necesită şi gradientul funcţiei, iar opţiunea `GradObj` este setată `'on'`, atunci argumentul `fun` furnizează, în plus, şi valoarea gradientului în punctul x. Dacă metoda de rezolvare a problemei necesită Hessiana, iar opţiunea `Hessian` este setată `'on'`, atunci argumentul `fun` furnizează, în plus, şi valoarea hessianei în punctul x.

Vectorul `x0` reprezintă punctul iniţial de unde porneşte procesul de căutare al minimului şi trebuie furnizat ca un vector din $\mathbb{R}^n$.

Variabila `optiuni` reprezintă setul de opţiuni specific fiecărei rutine Matlab. Comanda `optiuni=optimset('fminunc')` afişează setul de opţiuni implicit. Fiecare dintre acestea se poate modifica în funcţie de problema de minimizare. De exemplu, pentru setarea opţiunii `GradObj` pe `on`, comanda este `optiuni.GradObj='on'`.

Pentru o documentare amănunţită introduceţi următoarea comandă:
`help fminunc`

2. Funcţia `[...]=fminsearch(...)` returnează un punct de minim local al funcţiei introduse ca argument, utilizând însă o metodă diferită de cea folosită de `fminunc`, anume metoda Nedler-Mead, ce utilizează informaţie de ordin zero (evaluarea funcţiei). Sintaxa este similară celei prezentatě pentru funcţia `fminunc`.

Exemplul 1. Considerăm problema de optimizare

$$\min_{x\in\mathbb{R}^2} f(x) \quad \left(= e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)\right).$$

Să se rezolve problema (să se găsească un punct de minim local) utilizând rutina `fminunc`, cu punctul de iniţializare $x_0 = [1; -1]$.

Rezolvare. Se creează fişierul `objfun.m` ce va conţine următoarea secvenţă de cod:

```
function f=objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

Se apelează rutina `fminunc` în următorul mod:

```
x0=[-1,1]; optiuni=optimset(LargeScale,off);
[x,fval,exitflag,output]=fminunc(@objfun,x0,optiuni);
```

Exemplul 2. Considerăm problema de optimizare:

$$\min_{x\in\mathbb{R}} f(x) \quad (= \cos 3\pi x + \sin 2\pi x).$$

(i) Să se rezolve problema (să se găsească un punct de minim local) utilizând rutina `fminunc`, cu punctul de iniţializare $x_0 = 0$.

(ii) Să se rezolve problema cu ajutorul metodei secţiunii de aur.

Rezolvare. (i) Se creează fişierul `objfun.m` cu următoarea secvenţă de cod:

```
function f=objfun(x)
f = cos(3*pi*x) + sin(2*pi*x);
```

Se apelează rutina `fminunc`:

```
x0=0; optiuni=optimset(LargeScale,off);
[x,fval,exitflag,output]=fminunc(@objfun,x0,optiuni);
```

(ii) Deoarece funcţia este periodică, de perioadă 2π, putem aplica metoda secţiunii de aur, în care se alege lungimea intervalului iniţial egală cu 2π, a cărei implementare este expusă în cele ce urmează. Se creează fişierul `objfun.m` şi `goldsection.m`:

```
function f=objfun(x)
  f = cos(3*pi*x) + sin(2*pi*x);
end

function [xst,fst]= goldsection(x,eps)
  a=x-pi; b=x+pi; lambda=a+0.382*(b-a);
  miu=a+0.618*(b-a);
  while ((b-a)>=eps)
    if (objfun(lambda)<=objfun(miu))
         b=miu;
    else
         a=lambda;
    end
    lambda=a+0.382*(b-a); miu=a+0.618*(b-a);
  end

  if (objfun(lambda)<=objfun(miu))
    fst=objfun(lambda); xst=lambda;
  else
    fst=objfun(miu); xst=miu;
 end
```

Exemplul 3 (Interpolare polinomială)**.** O problemă des întâlnită în analiza numerică şi inginerie o reprezintă aproximarea optimă a unei funcţii neliniare cu un polinom de grad dat. Mai exact, fiind dată funcţia

neliniară $f : \mathbb{R} \to \mathbb{R}$ cu structură necunoscută, dorim aproximarea acesteia cu un polinom $g : \mathbb{R} \to \mathbb{R}$,

$$g(z) = a_n z^n + \cdots + a_1 z + a_0.$$

Datele cunoscute în legătură cu funcţia neliniară f reprezintă setul de perechi: $(y_1, f(y_1)), \ldots, (y_m, f(y_m))$, unde scalarii $y_i \in \mathbb{R}$ sunt daţi.
Notând vectorul coeficienţilor polinomului cu $x = [a_0 \ldots a_n]^T$ şi $b = [f(y_1) \ldots f(y_m)]^T$, problema se reduce la găsirea vectorului optim de coeficienţi ce satisface relaţia:

$$Ax = \begin{bmatrix} 1 & y_1 & \ldots & y_1^n \\ 1 & y_2 & \ldots & y_2^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_m & \ldots & y_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \approx \begin{bmatrix} f(y_1) \\ f(y_2) \\ \vdots \\ f(y_m) \end{bmatrix} = b.$$

Echivalent, obţinem următoarea problemă de optimizare fără constrângeri:

$$\min_{x \in \mathbb{R}^{n+1}} \; f(x) \quad \left(= \|Ax - b\|^2\right). \tag{1.1}$$

Să se rezolve problema (1.1), pe cazuri particulare pentru matricea A şi vectorul b de diferite dimensiuni, utilizând rutinele `fminunc` şi `lsqlin`. Să se compare rezultatele obţinute. Ce se observă?

1.2.2 Minimizare constrânsă

Fie funcţia (în general neliniară) $f : \mathbb{R}^n \to \mathbb{R}$. Forma generală a unei probleme de optimizare cu constrângeri (numite şi restricţii), asociate funcţiei f, se formulează în următorul mod:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{s.l.: } Cx \le d, \quad g(x) \le 0, \quad l \le x \le u, \\ &\qquad Ax = b, \quad h(x) = 0, \end{aligned} \tag{1.2}$$

unde $l, u \in \mathbb{R}^n, d \in \mathbb{R}^{m_1}, b \in \mathbb{R}^{p_1}, C \in \mathbb{R}^{m_1 \times n}, A \in \mathbb{R}^{p_1 \times n}$, iar $g(x) : \mathbb{R}^n \to \mathbb{R}^{m_2}, h(x) : \mathbb{R}^n \to \mathbb{R}^{p_2}$ sunt funcţii multidimensionale (vectori de funcţii), reprezentând constrângerile neliniare de inegalitate şi respectiv, de egalitate.

Exemplul 4. Considerăm următoarea problemă de optimizare:

$$\begin{aligned} &\min_{x\in\mathbb{R}^2} (x_1-3)^2+(x_2-2)^2 \\ &\text{s.l.:} \quad x_1^2-x_2-3\le 0, \;\; x_2-1\le 0, \;\; -x_1\le 0. \end{aligned}$$

Funcţia obiectiv şi cele trei constrângeri de inegalitate sunt definite de următoarele expresii:

$$f(x_1,x_2)=(x_1-3)^2+(x_2-2)^2,$$

$$g_1(x_1,x_2)=x_1^2-x_2-3, \qquad g_2(x_1,x_2)=x_2-1, \qquad g_3(x_1,x_2)=-x_1.$$

Fig 1.3 ilustrează mulţimea fezabilă. Problema se reduce la a găsi un punct în mulţimea fezabilă în care $(x_1-3)^2+(x_2-2)^2$ ia cea mai mică valoare. Observăm că punctele $[x_1\ x_2]^T$ cu $(x_1-3)^2+(x_2-2)^2=c$ sunt cercuri de rază c cu centrul în $[3\ 2]^T$. Aceste cercuri se numesc mulţimile nivel sau contururile funcţiei obiectiv având valoarea c. Pentru a minimiza c trebuie să găsim cercul cu cea mai mică rază care intersectează mulţimea fezabilă. După cum se observă din Fig. 1.3, cel mai mic cerc corespunde lui $c=2$ şi intersectează mulţimea fezabilă în punctul de optim $x^*=[2\ 1]^T$.

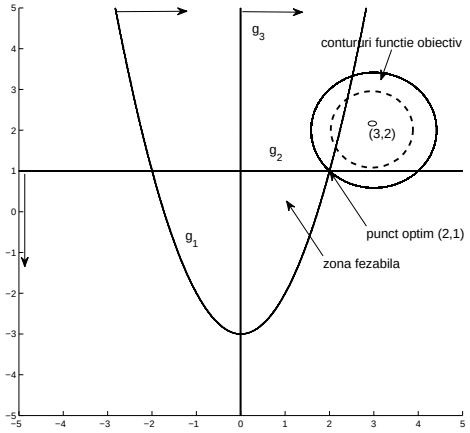

Figura 1.3: *Soluţia grafică a problemei de optimizare.*

Funcţii MATLAB

Pentru cazul în care funcţia de minimizat (maximizat) nu are o structură prestabilită sau o formă particulară, pentru rezolvarea problemei de optimizare (1.2) este disponibilă funcţia MATLAB `fmincon` ce prezintă următoarea sintaxă:

```
x = fmincon(fun,x0,C,d,A,b,l,u,nonlcon,optiuni)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(...).
```

variabila `fun` reprezintă funcţia obiectiv ce trebuie furnizată ca o variabilă de tip *function handle*. Dacă metoda de rezolvare a problemei cere şi gradientul funcţiei, iar opţiunea `GradObj` este setată `'on'`, atunci argumentul `fun` furnizează, în plus, şi valoarea gradientului în punctul x. Dacă metoda de rezolvare a problemei cere Hessiana, iar opţiunea `Hessian` este setată `'on'`, atunci argumentul `fun` furnizează, în plus, şi valoarea Hessianei în punctul x;

matricele şi vectorii C, d, A, b, l, u definesc constrângerile liniare cu structura prezentată în modelul (1.2);

variabila de tip handle `nonlcon` returnează constrângerile neliniare de egalitate şi inegalitate reprezentate în (1.2);

`x0` reprezintă punctul iniţial de unde porneşte procesul de căutare al minimului şi trebuie furnizat ca un vector din $\mathbb{R}^n$;

variabila `optiuni` reprezintă setul de opţiuni specific fiecărei rutine `MATLAB`. Comanda `optiuni=optimset('fmincon')` afişează setul de opţiuni implicit. Fiecare dintre acestea se poate modifica în funcţie de problema de minimizare. De exemplu, pentru setarea opţiunii `GradObj` pe `on`, comanda este `optiuni.GradObj='on'`.

Pentru o documentare amănunţită introduceţi următoarea comandă:
`help fmincon`

Exemplul 5. Considerăm problema de optimizare:

$$\min_{x\in\mathbb{R}^2} f(x) \qquad \left(= e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)\right).$$

$$\text{s.l.: } g(x) \leq 0 \quad \left(\begin{cases} x_1x_2 - x_1 - x_2 + 1.5 \leq 0 \\ -x_1x_2 - 10 \leq 0 \end{cases}\right).$$

Să se rezolve local problema utilizând rutina `fmincon`, cu punctul de iniţializare $x_0 = [1\ 1]^T$.

Rezolvare. Se creează fişierul `objfun.m` ce defineşte funcţia obiectiv şi `confun.m` în care se definesc constrângerile:

```
function f=objfun(x)
f=exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);

function[g,h]=confun(x)
g=[x(1)*x(2)-x(1)-x(2)+1.5 -x(1)*x(2)-10];
h=[];
```

Apoi se apelează rutina pentru minimizare cu restricţii:

```
x0=[-1,1];
optiuni=optimset(fmincon);
optiuni.LargeScale=off;
[x,fval]=fmincon(@objfun,x0,[],[],[],[],...
          [],[],@confun,optiuni)
```

Exemplul 6 (Analiză statistică)**.** Analiza datelor şi interpretarea acestora într-un sens cât mai corect este preocuparea principală din domeniul statisticii. Problema se formulează în următorul mod: pe baza unei colecţii de date cunoscute (reprezentate în Fig. 1.4 prin puncte), să se realizeze predicţia cu o eroare cât mai mică a unui alt set de date parţial cunoscut. În termeni matematici, această problemă presupune determinarea unei direcţii de-a lungul căreia elementele date (punctele) tind să se alinieze, astfel încât să se poată prezice zona de apariţie a punctelor viitoare. S-a constatat că direcţia de căutare este dată de vectorul singular al matricei formate din colecţia de puncte date, ce poate fi găsit prin intermediul următoarei probleme de optimizare:

$$\begin{aligned} &\max_{x\in\mathbb{R}^n} f(x) \qquad \left(= \frac{1}{2}x^T A^T A x\right) \\ &\text{s.l.: } g(x) \le 0 \quad (\|x\| \le 1)\,, \end{aligned} \tag{1.3}$$

unde $A \in \mathbb{R}^{m\times n}$ reprezintă matricea ale cărei coloane $[a_1 \dots a_n]$ sunt punctele cunoscute iniţial. Considerând cazul particular în care se dă matricea $A = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 2 & 1 \end{bmatrix}$, să se rezolve problema precedentă cu ajutorul funcţiei `fmincon`, alegând punctul iniţial $x_0 = [-1\ \ -2\ \ 0.5]^T$.

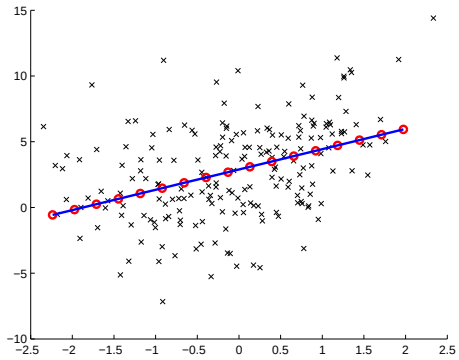

Figura 1.4: *Un exemplu de dispersie a datelor în spaţiul* $\mathbb{R}^2$*. Observăm că punctele se aliniază de-a lungul dreptei ce iese în evidenţă.*

Rezolvare. Se creează fişierele `objfun.m` şi `confun.m` pentru funcţia obiectiv şi respectiv, pentru constrângeri :

```
function [f]=objfun(x)
f=x(1)^2+4.25*x(2)^2+x(3)^2+0.5*x(1)*x(2)+2*x(2)*x(3);

function [g,h]=confun(x)
g=x(1)^2+x(2)^2 + x(3)^2 - 1;
h=[];
```

Se apelează rutina pentru minimizare cu restricţii:

```
x0=[-1,-2,0.5];
optiuni=optimset(fmincon);
optiuni.LargeScale=off;
[x,fval]=fmincon(@objfun,x0,[],[],[],...
            [],[],[],@confun,optiuni)
```

Exemplul 7 (Analiza stabilităţii robuste)**.** Stabilitatea robustă a sistemelor liniare afectate de perturbaţii reprezintă o temă de cercetare de actualitate în teoria controlului. Numeroase aplicaţii practice necesită analiza stabilităţii robuste (Fig. 1.5) pentru asigurarea funcţionării corespunzătoare a diferitelor sisteme afectate de zgomot şi perturbaţii (e.g. sistemul de viraj al autovehiculelor, sistemul de ghidare al unei macarale).

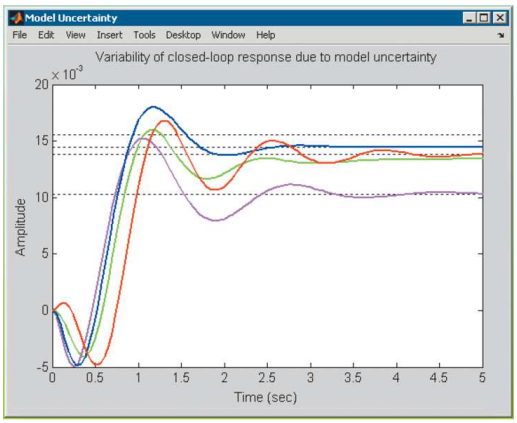

Figura 1.5: *Exemplu de reglare robustă în cazul unui sistem cu parametri afectaţi de incertitudini.*

În general, problema se reduce la studiul stabilităţii unui polinom cu coeficienţi afectaţi de incertitudini. Considerăm ca exemplu următorul polinom:

$$s^3 + (1 + a_1 + a_2)s^2 + (a_1 + a_2 + 3)s + (1 + 6a_1 + 6a_2 + 2a_1a_2), \quad (1.4)$$

unde parametrii a_1 şi a_2 satisfac condiţiile:

$$a_1 \in [1.4 - 1.1\delta,\ 1.4 + 1.1\delta] \quad \text{şi} \quad a_2 \in [0.85 - 0.85\delta,\ 0.85 + 0.85\delta],$$

unde $\delta > 0$. Parametrii a_1 şi a_2 pot reprezenta cantităţi fizice (e.g. lungimi, mase) ce sunt afectate de incertitudini în mod obişnuit, sau parametrii regulatorului ce pot varia datorită erorilor de implementare. Suntem interesaţi în determinarea celei mai mari valori a variabilei δ pentru care polinomul rămâne stabil. Această problemă se poate reformula într-una de optimizare folosind *criteriul Hurwitz*: un polinom de gradul 3 de forma

$$p(s) = s^3 + q_2s^2 + q_1s + q_0$$

este stabil dacă şi numai dacă $q_2q_1 - q_0 > 0$ şi $q_0 > 0$. De aceea, considerăm marginea maximă de stabilitate dată de

$$\delta^* = \min\{\delta_1, \delta_2\},$$

unde δ_1 şi δ_2 sunt soluţiile următoarelor probleme de optimizare:

$$\begin{aligned}\delta_1 = \min_{a,\delta} \quad & \delta \\ \text{s.l.: } & 2 - 2a_1 - 2a_2 + a_1^2 + a_2^2 \leq 0, \\ & 1.4 - 1.1\delta \leq a_1 \leq 1.4 + 1.1\delta \\ & 0.85 - 0.85\delta \leq a_2 \leq 0.85 + 0.85\delta,\end{aligned}$$

$$\begin{aligned}\delta_2 = \min_{a,\delta} \quad & \delta \\ \text{s.l.: } & 1 + 6a_1 + 6a_2 + 2a_1a_2 \leq 0, \\ & 1.4 - 1.1\delta \leq a_1 \leq 1.4 + 1.1\delta \\ & 0.85 - 0.85\delta \leq a_2 \leq 0.85 + 0.85\delta.\end{aligned}$$

Să se rezolve problemele de optimizare neliniară precedente utilizând funcţia `fmincon`.

Rezolvare. Pentru a simplifica rezolvarea furnizăm secvenţa de cod Matlab ce rezolvă prima problemă de optimizare şi determină δ_1, cea de-a doua rezolvându-se în mod analog. Considerăm următoarea secvenţă de cod aferentă funcţiei obiectiv şi constrângerilor problemei de optimizare ce determină δ_1:

```
function [f]=objfun(delta)
f=delta;
end

function [g]=confun(x)
g=[1+6*x(1)+6*x(2)+2*x(1)*x(2);x(1)-1.4-1.1*x(3);...
   -x(1)+1.4-1.1*x(3); x(2)-0.85-0.85*x(3);...
   -x(2)+0.85-0.85*x(3)];
end
```

Se apelează apoi rutina pentru minimizare cu restricţii:

```
x0=[1.4 0.85 1000];
optiuni=optimset(fmincon);
optiuni.LargeScale=off;
[x,fval]=fmincon(@objfun,x0,[],[],[],...
      [],[],[],@confun,optiuni)
```

1.2.3 Programare liniară

Pentru cazurile particulare de probleme de optimizare (e.g. funcţie obiectiv liniară sau pătratică), există funcţii Matlab specifice ce rezolvă problemele de optimizare aferente prin intermediul unor algoritmi dedicaţi respectivelor clase de probleme.
Funcţia `linprog` rezolvă probleme de *programare liniară*, ce vizează minimizarea unei funcţii obiectiv liniare, în prezenţa unui set de constrângeri liniare. O problemă de programare liniară se formulează în general după cum urmează:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \; & c^T x \\ \text{s.l.:} \; & Cx \leq d, \;\; l \leq x \leq u, \\ & Ax = b, \end{aligned}$$

unde $c, l, u \in \mathbb{R}^n, d \in \mathbb{R}^m, b \in \mathbb{R}^p, C \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{p\times n}$. Sintaxa funcţiei `linprog` cuprinde:

```
x = linprog(c,C,d,A,b,l,u,x0,optiuni)
[x,fval,exitflag,output,lambda] = linprog(. . . ).
```

Pentru mai multe detalii, se poate apela comanda: `help linprog`

Exemplul 8 (*Repartizarea optimală a resurselor umane*)**.** Fie un număr de n persoane disponibile pentru m sarcini. Valoarea atribuită unei zile de lucru a persoanei i la o sarcină j este definită de c_{ij}, pentru $i = 1, \ldots, n$ şi $j = 1, \ldots, m$. Problema se reduce la determinarea repartizării optime de sarcini ce maximizează valoarea lucrului total. Repartizarea sarcinilor constă în alegerea valorilor x_{ij}, ce reprezintă proporţia din timpul persoanei i consumată pe sarcina j pentru $i = 1, \ldots, n$ şi $j = 1, \ldots, m$. Vom avea astfel următoarele constrângeri:

$$\sum_{j=1}^{m} x_{ij} \leq 1, \quad \sum_{i=1}^{n} x_{ij} \leq 1, \quad x_{ij} \geq 0.$$

Prima relaţie denotă că o persoană nu poate depăşi 100% din timpul său lucrând. A două relaţie precizează că unei singure persoane îi este permis să lucreze la o sarcină la un moment dat. În final, a treia relaţie interzice

ca timpul petrecut pe o sarcină să fie negativ. În concluzie, problema enunţată anterior are următoarea formulare:

$$\min_{x\in\mathbb{R}^{nm}} f(x) \qquad \left(= \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}\right)$$

$$\text{s.l.: } Cx \le d \qquad \left(\begin{cases} \sum_{j=1}^{m} x_{ij} \le 1 \ \ \forall i = 1,\dots,n \\ \sum_{i=1}^{n} x_{ij} \le 1 \ \ \forall j = 1,\dots,m \\ x_{ij} \ge 0 \ \ \forall i,j \end{cases}\right).$$

Fie matricea coeficienţilor c_{ij} dată de $\begin{bmatrix} 1 & -1 & 0 \\ 0.1 & -5 & 2 \\ 0 & -1 & -0.5 \end{bmatrix}$, adica n=m=3. Să se rezolve problema de mai înainte cu ajutorul funcţiei `linprog`.

Rezolvare. Iniţial compunem matricea C şi vectorul d ce formează constrângerile pentru a putea apela funcţia `linprog` în forma standard:

```
function [x,f]=alocare()
 c=[1 -1 0 0.1 -5 2 0 -1 -0.5];
 A1=[ones(1,3) zeros(1,6);...
     zeros(1,3) ones(1,3) zeros(1,3);...
     zeros(1,6) ones(1,3)];
 A2=[eye(3) eye(3) eye(3)];
 C = [A1; A2; -eye(9)]; d= [ones(6,1);zeros(9,1)];
 [x f] = linprog(c,C,d);
 end
```

Exemplul 9 (*Proiectare filtre*)**.** Una dintre cele mai importante probleme ale ingineriei o reprezintă proiectarea filtrelor cu răspuns la impuls prestabilit. Prin noţiunea de filtru înţelegem un *sistem dinamic liniar invariant în timp*, care de cele mai multe ori este definit de relaţia de *convoluţie* dintre două semnale discrete/continue:

$$y(t) = \int_{-\infty}^{\infty} u(s)h(t-s)ds \qquad\qquad y(k) = \sum_{i=-\infty}^{\infty} u(i)h(k-i),$$

în care h este răspunsul la impuls corespunzător sistemului. Echivalent, în domeniul frecvenţial, transferul intrare-ieşire pentru cazul continuu anterior se reprezintă sub forma:

$$Y(\omega) = H(\omega)U(\omega),\ -\infty < \omega < \infty,$$

unde $Y(\omega), H(\omega), U(\omega)$ sunt transformatele Fourier ce corespund funcţiilor $y(t), h(t), u(t)$. Problema de sinteză a unui filtru presupune determinarea configuraţiei optime globale a unui bloc (colecţie) de filtre cu funcţii de tranfer cunoscute (Fig. 1.6), astfel încât funcţia de transfer globală rezultată să coincidă cu una prestabilită. Notând funcţia de transfer globală prestabilită cu $T(\omega)$, considerăm măsurarea calităţii funcţiei de transfer pe segmentul $[\omega_{\min}, \omega_{\max}]$. Rezultă astfel următoarea problemă de optimizare:

$$\min_{x\in\mathbb{R}^n} \sup_{\omega_{\min}\le\omega\le\omega_{\max}} |T(\omega) - \sum_{j=0}^{n-1} x_j H_j(\omega)|,$$

unde $H_j(\omega)$ reprezintă funcţiile de transfer aferente filtrelor ce compun blocul. O versiune mai abordabilă este următoarea aproximare a problemei precedente:

$$\min_{x\in\mathbb{R}^n} \max_{1\le i\le m} |T(\omega_i) - \sum_{j=0}^{n-1} x_j H_j(\omega_i)|, \tag{1.5}$$

în care s-a recurs la alegerea unui număr finit m de puncte în intervalul $[\omega_{\min}, \omega_{\max}]$.

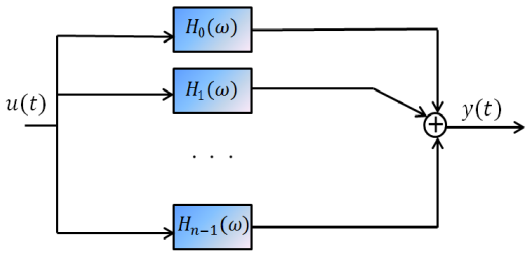

Figura 1.6: *Exemplu bloc de filtre*

Problema (1.5) se poate reformula în termenii unei probleme de minimizare cu funcţia obiectiv liniară şi având de asemenea constrângeri

liniare de inegalitate, i.e. un program liniar:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n,t} f(x,t) && (=t)\\ &\text{s.l.: } C\begin{bmatrix}x\\t\end{bmatrix} \le d && \left(|T(\omega_i) - \sum_{j=0}^{n-1} x_j H_j(\omega_i)| \le t \quad \forall i = 1,\dots,m\right). \end{aligned}$$

Exerciţiul 1. Se consideră cazul particular ce cuprinde funcţiile: $T(\omega) = \cos\omega^2 + \omega^3$, $H_0(\omega) = \frac{\omega^2}{1+\omega}$, $H_1(\omega) = \frac{2\omega}{1+\omega}$, $H_2(\omega) = \omega^2 + \omega + 1$ şi vectorul $\bar{\omega} = [0.1\ 0.5\ 1\ 5\ 10]$. Ţinând cont că valorile ω_i reprezintă componentele vectorului $\bar{\omega}$, să se scrie modelul problemei anterioare sub forma standard a unei probleme de programare liniară şi să se rezolve problema cu ajutorul funcţiei Matlab `linprog`.

1.2.4 Programare pătratică

Funcţia `quadprog` rezolvă probleme de *programare pătratică*, ce vizează minimizarea unei funcţii obiectiv pătratice, în prezenţa unui set de constrângeri liniare. O problemă de programare pătratică se formulează în general după cum urmează:

$$\begin{aligned} \min_{x\in\mathbb{R}^n}\ & \frac{1}{2}x^TQx + q^Tx\\ \text{s.l.: }\ & Cx \le d,\ \ l \le x \le u,\\ & Ax = b, \end{aligned}$$

unde $Q = Q^T \in \mathbb{R}^{n\times n}$, $q, l, u \in \mathbb{R}^n, d \in \mathbb{R}^m, b \in \mathbb{R}^p, C \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{p\times n}$. Sintaxa funcţiei `quadprog` cuprinde:

```
x = quadprog(Q,q,C,d,A,b,l,u,x0,optiuni)
[x,fval,exitflag,output,lambda] = quadprog(. . . ).
```

Pentru mai multe detalii, se poate apela comanda: `help` quadprog

Exemplul 10 (*Problema Google*)**.** Ca urmare a progreselor tehnologice recente, motoarele de căutare (e.g. Google, Yahoo) au devenit un punct central de interes în optimizare, urmărind dezvoltarea de algoritmi eficienţi ce execută o căutare cât mai rapidă. Constatând că o căutare eficientă presupune o clasificare cât mai strategică a paginilor web, rezultă o reducere a problemei de căutare la una de clasificare a acestor

pagini. Problema de clasificare (*ranking*) se formulează uşor în termeni de grafuri ponderate (Fig. 1.7). De aceea, paginile web sunt interpretate ca noduri ale unui graf, muchiile ca legături aferente dintre ele, iar fiecărei muchii îi este asociată o pondere.

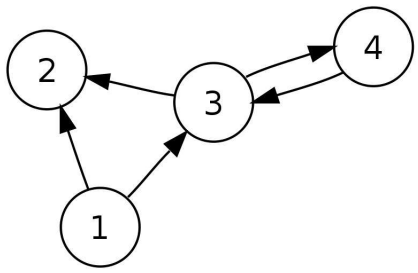

Figura 1.7: *Exemplu graf orientat*

Dacă notăm cu E matricea de adiacenţă a grafului, elementele acesteia respectă următoarea regulă: E_{ij} este nenul dacă există o muchie direcţionată de la i la j, altfel este nul. Dacă ţinem cont de faptul că graful este ponderat şi că suma ponderilor este egală cu 1, atunci matricea E va avea suma pe linii egală cu 1. Matematic, problema precedentă presupune găsirea unui vector propriu x corespunzător valorii proprii 1, i.e. să se determine $x \in \mathbb{R}^n$ astfel încât $Ex = x$. Reformulând mai departe problema în termeni de optimizare avem:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) && \left(= \frac{1}{2}\|Ex - x\|^2\right) \\ &\text{s.l.: } Cx \leq d && (x \geq 0) \\ &\qquad Ax = b && \left(\sum_{i=1}^{n} x_i = 1\right). \end{aligned}$$

De cele mai multe ori, matricea are dimensiuni foarte mari (e.g. $10^9 \times 10^9$) ceea ce face problema foarte greu de rezolvat (vezi Secţiunea 7.2.3).

Exerciţiul 2. Fie matricea $E = \begin{bmatrix} 0.1 & 0 & 0 & 0.9 \\ 0 & 0 & 0.5 & 0.5 \\ 0.2 & 0 & 0.2 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Să se rezolve problema cu ajutorul funcţiei `quadprog`, folosind mai întâi doar constrângeri de egalitate şi apoi incluzând şi constrângerile de inegalitate. Ce se observă?

1.3 Probleme rezolvate de seminar

1.3.1 Diferenţiere multivariabilă

Problema 1. Fie funcţiile $f : \mathbb{R}^3 \to \mathbb{R}$ şi $g : \mathbb{R}^2 \to \mathbb{R}$ definite de:

$$f(x) = x_1^3 x_2 + 2x_3^2 x_1 - x_2 x_3,$$
$$g(x) = e^{x_1 - x_2} + e^{2x_1 - 1} - e^{2x_2 - 2}.$$

(i) Să se determine expresiile gradienţilor $\nabla f(x), \nabla g(x)$.

(ii) Să se determine expresiile Hessianelor $\nabla^2 f(x), \nabla^2 g(x)$.

Rezolvare. Pentru o funcţie diferenţiabilă de 2 ori $f : \mathbb{R}^n \to \mathbb{R}$, reamintim definiţiile gradientului şi Hessianei:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}, \qquad \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

(i) Expresiile gradienţilor funcţiilor din enunţ au forma:

$$\nabla f(x) = \begin{bmatrix} 3x_1^2 x_2 + 2x_3^2 \\ x_1^3 - x_3 \\ 4x_3 x_1 - x_2 \end{bmatrix}, \qquad \nabla g(x) = \begin{bmatrix} e^{x_1 - x_2} + 2e^{2x_1 - 1} \\ -e^{x_1 - x_2} - 2e^{2x_2 - 2} \end{bmatrix}.$$

(ii) Expresiile Hessianelor funcţiilor din enunţ au forma:

$$\nabla^2 f(x) = \begin{bmatrix} 6x_1 x_2 & 3x_1^2 & 4x_3 \\ 3x_1^2 & 0 & -1 \\ 4x_3 & -1 & 4x_1 \end{bmatrix},$$

$$\nabla^2 g(x) = \begin{bmatrix} e^{x_1 - x_2} + 4e^{2x_1 - 1} & -e^{x_1 - x_2} \\ -e^{x_1 - x_2} & e^{x_1 - x_2} - 4e^{2x_2 - 2} \end{bmatrix}.$$

Problema 2. Fie funcţia multidimensională $h : \mathbb{R}^3 \to \mathbb{R}^2$,

$$h(x) = \begin{bmatrix} x_2^2 + x_3 \\ x_1^2 + x_2 \end{bmatrix}.$$

(i) Determinaţi expresia Jacobianului $\nabla h(x)$.

(ii) Arătaţi că, în orice vector $x \in \mathbb{R}^3$, Jacobianul calculat are liniile liniar independente.

Rezolvare. (i) Din definiţia Jacobianului pentru o funcţie $h : \mathbb{R}^3 \to \mathbb{R}^2$, i.e. $\nabla h(x) = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x_1} & \frac{\partial h_1(x)}{\partial x_2} & \frac{\partial h_1(x)}{\partial x_3} \\ \frac{\partial h_2(x)}{\partial x_1} & \frac{\partial h_2(x)}{\partial x_2} & \frac{\partial h_2(x)}{\partial x_3}. \end{bmatrix}$, avem $\nabla h(x) = \begin{bmatrix} 0 & 2x_2 & 1 \\ 2x_1 & 1 & 0 \end{bmatrix}$.

(ii) Din definiţia proprietăţii de liniar independenţă avem: doi vectori $u, v \in \mathbb{R}^n$ sunt liniar independenţi dacă nu există un scalar $\alpha \neq 0$ astfel încât $u = \alpha v$. Această proprietate este evidentă observând că egalitatea:

$$\begin{bmatrix} 2x_1 \\ 1 \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 0 \\ 2x_2 \\ 1 \end{bmatrix}$$

nu poate avea loc dacă $\alpha \neq 0$.

1.3.2 Probleme de optimizare generale

Problema 3. Fie următoarea problemă de optimizare:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \; & x_1^3 + x_2^3 \\ \text{s.l.:} \; & (x_1 + x_2)^2 \leq 0, \\ & x_1 x_2^3 = -1. \end{aligned}$$

(i) Să se precizeze care sunt constrângerile de egalitate şi care sunt cele de inegalitate.

(ii) Să se determine analitic punctul de optim şi valoarea optimă.

Rezolvare. (i) Se observă că inegalitatea $(x_1 + x_2)^2 \leq 0$ este echivalentă cu $(x_1 + x_2)^2 = 0$. Concluzionăm că nu există constrângeri de inegalitate şi avem două constrângeri de egalitate: $h_1(x) = x_1 + x_2 = 0$ şi $h_2(x) = x_1 x_2^3 + 1 = 0$.
(ii) Din punctul (i) rezultă că $x_1 + x_2 = 0$ şi făcând sistem cu cea de-a doua constrângere, concluzionăm că mulţimea fezabilă este formată din două puncte, anume $(x_1, x_2) = (1, -1)$ şi $(x_1, x_2) = (-1, 1)$. Punctele de optim global vor fi $(x_1^*, x_2^*) = (1, -1)$ şi $(x_1^*, x_2^*) = (-1, 1)$, din moment ce valoarea optimă $f^* = 0$ este aceeaşi pentru ambele.

Problema 4. Fie funcţia $f : \Delta_n \to x, f(x) = \ln x^T A x$, unde Δ_n este o mulţime, numită mulţimea simplex, şi este dată de $\Delta_n = \{x \in \mathbb{R}^n :$

$\sum_{i=1}^{n} x_i = 1, x_i \geq 0 \ \ \forall i = 1, \ldots, n\}$, $A \in \mathbb{R}^{n \times n}$ este o matrice simetrică şi pozitivă (i.e. toate elementele matricei sunt ne-negative).

(i) Să se determine expresia gradientului $\nabla f(x)$ şi a Hessianei $\nabla^2 f(x)$.

(ii) Să se determine constanta Lipschitz a gradientului funcţiei f.

Rezolvare: (i) Prin calcule simple, obţinem că expresia gradientului este dată de $\nabla f(x) = \frac{2Ax}{x^T Ax}$, iar a hessianei $\nabla^2 f(x) = \frac{2Ax^T Ax - (2Ax)(2Ax)^T}{(x^T Ax)^2} = \frac{2A}{x^T Ax} - \frac{4Ax}{x^T Ax}\left(\frac{Ax}{x^T Ax}\right)^T$.

(ii) Gradientul este continuu în sens Lipschitz dacă următoarea inegalitate are loc:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2 \ \ \forall x, y \in \Delta_n.$$

Echivalent, această proprietate se exprimă şi în termenii matricei Hessiane:

$$\|\nabla^2 f(x)\|_2 \leq L \ \ \forall x \in \Delta^n.$$

Obervăm că matricea $\frac{(2Ax)(2Ax)^T}{(x^T Ax)^2}$ este pozitiv semidefinită, de aceea avem următoarea mărginire a normei Hessianei:

$$\|\nabla^2 f(x)\| \leq \|\frac{2A}{x^T Ax}\| \ \ \forall x \in \Delta_n.$$

Mai mult, observăm că

$$\min_{x \in \Delta_n} x^T Ax = \min_{x \in \Delta_n} \sum_{i,j} A_{ij} x_i x_j \geq \min_{x \in \Delta_n} \left(\min_i A_{ii}\right) \|x\|^2 = \frac{1}{N}\left(\min_i A_{ii}\right).$$

De aici obţinem o aproximare a constantei Lipschitz corespunzătoare gradientului funcţiei din enunţ:

$$\|\nabla^2 f(x)\| \leq N\|\frac{2A}{\min\limits_i A_{ii}}\| = L.$$

Problema 5. Fie $P \in \mathbb{R}^{n \times n}$ o matrice simetrică şi inversabilă. Să se arate că funcţia pătratică:

$$f(x) = \frac{1}{2} x^T Px + x^T b$$

are valoare minimă (i.e. este mărginită inferior) dacă şi numai dacă $P \succeq 0$ (i.e. este pozitiv semidefinită). Mai mult, în cazul în care este mărginită, arătaţi că punctul de minim x^* este unic şi determinat de expresia $x^* = -P^{-1}b$.

Rezolvare. Observăm că

$$\frac{1}{2}\left(x+P^{-1}b\right)^T P\left(x+P^{-1}b\right)=\frac{1}{2}x^TPx+b^Tx+\frac{1}{2}b^TP^{-1}b.$$

De aici rezultă,

$$f(x)=\frac{1}{2}x^TPx+b^Tx=\frac{1}{2}\left(x+P^{-1}b\right)^T P\left(x+P^{-1}b\right)-\frac{1}{2}b^TP^{-1}b.$$

Presupunem că P are o valoare proprie negativă, e.g. $-\lambda$ (unde $\lambda>0$) şi notăm vectorul propriu asociat valorii proprii negative cu u. Atunci, pentru orice $\alpha\in\mathbb{R},\alpha\neq 0$, considerând $x=\alpha u-P^{-1}b$ şi ţinând cont de relaţia $Pu=-\lambda u$, avem:

$$\begin{aligned}f(x)&=\frac{1}{2}\left(x+P^{-1}b\right)^T P\left(x+P^{-1}b\right)-\frac{1}{2}b^TP^{-1}b\\&=\frac{1}{2}\alpha u^TP\alpha u-\frac{1}{2}b^TP^{-1}b\\&=-\frac{1}{2}\lambda\alpha^2\|u\|_2^2-\frac{1}{2}b^TP^{-1}b.\end{aligned}$$

În final, observăm că $\lambda>0$ şi avem libertatea de a alege α oricât de mare, de unde rezultă că funcţia nu are minimum (i.e. minimum se atinge la $-\infty$). Pentru ca funcţia să aibă minimum este necesar ca $P\succeq 0$. Pentru a arăta ultima parte a rezultatului, din reformularea lui $f(x)$ anterioară avem:

$$f(x)=\frac{1}{2}\left(x+P^{-1}b\right)^T P\left(x+P^{-1}b\right)-\frac{1}{2}b^TP^{-1}b.$$

Observăm că $\frac{1}{2}\left(x+P^{-1}b\right)^T P\left(x+P^{-1}b\right)\geq 0$, deci minimul funcţiei se atinge atunci când $x+P^{-1}b=0$.

Problema 6. Fie problema de optimizare :

$$\begin{gathered}\min_{x\in\mathbb{R}^n}\ f(x)\quad\left(=\frac{c^Tx+d}{e^Tx+f}\right)\\\text{s.l. } Gx\leq h,\ Ax=b,\end{gathered}$$

cu $\text{domf}(x)=\left\{x\in\mathbb{R}^n:e^Tx+f>0\right\}$. Arătaţi că această problemă poate fi adusă la forma de programare liniară.

Rezolvare. Notăm $u(x) = e^T x + f > 0$. Împărţind relaţia prin $u(x)$, aceasta devine: $\frac{e^T x}{u(x)} + f\frac{1}{u(x)} = 1 > 0$. De asemenea, folosim notaţia $y(x) = \frac{x}{u(x)} \in \mathbb{R}^n$ şi $z(x) = \frac{1}{u(x)} \in \mathbb{R}$. Reformulând funcţia obiectiv avem:

$$f(x) = \frac{c^T x + d}{u(x)} = c^T \frac{x}{u(x)} + d\frac{1}{u(x)} = c^T y(x) + dz(x).$$

Folosind schimbarea de variabilă precedentă, putem aduce problema de minimizare după x la una ce presupune minimizarea după $y(x)$ şi $z(x)$. Pentru a schimba variabila constrângerilor, împarţim prin u ultimele două seturi de egalităţi/inegalităţi. Astfel, obţinem un LP:

$$\begin{aligned} \min_{y\in\mathbb{R}^n, z\in\mathbb{R}} \; & c^T y + dz \\ \text{s.l.: } & e^T y + fz = 1, \;\; Ay = bz \\ & Gy \leq hz. \end{aligned}$$

Problema 7. Fie următoarea problemă de optimizare:

$$\begin{aligned} \min_{x\in\mathbb{R}^2} \; & 2x_1^2 + x_2^2 - x_1 x_2 \\ \text{s.l.: } & 2^{x^T x - 1} \leq 2, \;\; (a^T x - b)^2 \leq 0 \\ & 5^{|x_1|} \leq 1, \; 3^{x_1 + x_2} = 1, \end{aligned}$$

unde $a \in \mathbb{R}^2$ şi $b \in \mathbb{R}$ sunt daţi. Să se arate că toate funcţiile care descriu problema de optimizare sunt fie liniare, fie pătratice. Să se precizeze care sunt constrângerile de egalitate şi care sunt cele de inegalitate.
Rezolvare. Se observă că funcţia obiectiv $f(x) = 2x_1^2 + x_2^2 - x_1 x_2$ este pătratică, i.e.

$$f(x) = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Constrângerea $2^{x^T x - 1} \leq 2$ este echivalentă cu $x^T x - 1 \leq 1$, i.e. $x^T x \leq 2$. Deci prima constrângere este una de inegalitate descrisă de o funcţie pătratică $g_1(x) = x^T x - 2 \leq 0$.
Constrângerea $(a^T x - b)^2 \leq 0$ este echivalentă cu $a^T x - b = 0$. Deci a doua constrângere este de egalitate descrisă de o funcţie liniară $h_1(x) = a^T x - b = 0$.

Constrângerea $5^{|x_1|-1} \leq 1$ este echivalentă cu $|x_1| - 1 \leq 0$. Deci a treia constrângere este de inegalitate, descrisă de două funcţii liniare $g_2(x) = x_1 - 0 \leq 0$ şi $g_3(x) = -x_1 - 1 \leq 0$.
Constrângerea $3^{x_1+x_2} = 1$ este echivalentă cu $x_1 + x_2 = 0$. Deci a patra constrângere este de egalitate, descrisă de o funcţie liniară $h_2(x) = x_1 + x_2 = 0$.
Problema 8. O funcţie se numeşte *monomială* dacă se prezintă sub forma:

$$f(x) = cx_1^{a_1}x_2^{a_2}...x_n^{a_n},$$

în care $c > 0$, $a_i \in \mathbb{R}$ şi $x \in \mathbb{R}^n_{++}$, i.e. $x_i > 0$ pentru orice $i = 1, \ldots, n$. Considerând problema de programare geometrică :

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &\text{s.l.: } g_i(x) \leq 1 \ \ \forall i = 1, \ldots, m \\ &\qquad h_i(x) = 1 \ \ \forall i = 1, \ldots, p, \end{aligned}$$

cu f_i şi h_i funcţii monomiale. Arătaţi că o astfel de problemă de programare geometrică poate fi scrisă sub forma unui LP.

Rezolvare. Din teoria optimizării ştim că punctul de optim al unei probleme de optimizare cu funcţia obiectiv $f(x)$, este acelaşi cu cel al problemei cu funcţia obiectiv $\log f(x)$. Într-adevăr, în cazul neconstrâns condiţiile suficiente de optimalitate pentru problema originală (cu funcţia obiectiv $f(x)$) sunt $\nabla f(x^\star) = 0$. Observăm că pentru cazul compunerii cu funcţia logaritm (cu funcţia obiectiv $\log f(x)$) condiţiile de optimalitate se transformă în $\frac{\nabla f(x^\star)}{f(x^\star)} = 0$, sau echivalent în $\nabla f(x^\star) = 0$. Precizăm că pentru cazul constrâns are loc o echivalenţă logaritmică similară, i.e având problema originală:

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} cx_1^{a_1}x_2^{a_2}...x_n^{a_n} \\ \text{s.l.: } c_i x_1^{b_1^i}x_2^{b_2^i}\ldots x_n^{b_n^i} \leq 1 \quad \forall i = 1, \ldots, m \\ \qquad e_j x_1^{d_1^j}x_2^{d_2^j}\ldots x_n^{d_n^j} = 1 \quad \forall j = 1, \ldots, p, \end{cases}$$

compunând funcţia obiectiv şi constrângerile cu funcţia logaritm, obţinem o problemă de optimizare cu acelaşi punct de optim:

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} \log(cx_1^{a_1}x_2^{a_2}...x_n^{a_n}) \\ \text{s.l.: } \log(c_i x_1^{b_1^i}x_2^{b_2^i}\ldots x_n^{b_n^i}) \leq 0 \quad \forall i = 1, \ldots, m \\ \qquad \log(e_j x_1^{d_1^j}x_2^{d_2^j}\ldots x_n^{d_n^j}) = 0 \quad \forall j = 1, \ldots, p. \end{cases}$$

Folosind schimbarea de variabilă $\log x_i = y_i$, observăm următoarea reformularea liniară a unei funcţii monomiale:

$$f(x) = \log(cx_1^{a_1} \dots x_n^{a_n}) = \log c + \sum_{i=1}^{n} a_i \log x_i = \log c + \sum_{i=1}^{n} a_i y_i.$$

Astfel, rezultă o formă LP a problemei în variabila y.

Problema 9. Fie funcţia pătratică $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x) = x_1^2 - x_2^2 + x_1 x_2$. Să se arate ca funcţia f are gradient Lipschitz în raport cu norma Euclidiană, i.e. există $L > 0$ astfel încât:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^2.$$

şi să se determine constanta Lipschitz corespunzătoare.

Rezolvare. Remarcăm că funcţia f se poate formula ca $f(x) = \frac{1}{2}x^T Q x$, unde $Q = \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}$. Verificând relaţia de continuitate Lipschitz avem:

$$\|Qx - Qy\|_2 = \|Q(x - y)\|_2 \leq \|Q\|_2 \|x - y\|_2, \quad \forall x, y \in \mathbb{R}^2.$$

Inegalitatea reiese din definiţia normei matriceale induse. Evident, constanta Lipschitz este dată de $L = \|Q\|_2 = \lambda_{\max}(Q)$.

1.4 Probleme propuse

Problema 1. Să se arate că problema de optimizare (1.1) este problemă pătratică (QP). Evidenţiaţi matricea Hessiană şi precizaţi dacă este sau nu pozitiv definită.

Problema 2. Fie funcţia neliniară $g(x) = -5e^{2x} + \cos x - 1$. Să se aproximeze cu un polinom de gradul 3 funcţia g după modelul descris în Exemplul 3 pentru punctele $y_1 = -1, y_2 = -1/2, y_3 = 0, y_4 = 1/2$ şi $y_5 = 1$. Să se rezolve problema de optimizare cu ajutorul funcţiilor `fminunc` şi `quadprog`.

Problema 3. Fie următoarea problemă de optimizare constrânsă:

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \ & \frac{1}{2}\|Ax\|^2 \\ \text{s.l.:} \ & \|x\|_1 \leq 1, \end{aligned}$$

unde $A \in \mathbb{R}^{m\times n}$. Să se reformuleze problema precedentă ca una pătratică (QP) şi să se rezolve apoi această formulare cu ajutorul funcţiei Matlab `quadprog`.

Problema 4. Să se calculeze expresiile gradientului şi Hessianei corespunzătoare următoarelor funcţii:

(i) $f(x) = x_1^2 - x_2^2 - 2x_3^2 + 2x_2x_3 + 3x_1x_2$ (ii) $f(x) = x_1^3 + x_2x_3^2$

(iii) $f(x) = e^{x_1x_2} + e^{x_2x_3} + e^{x_3x_1}$ (iv) $f(x) = x_1 \ln x_1 + x_2 \ln x_2 + x_3 \ln x_3$.

Problema 5. Să se arate că funcţia pătratică $f(x) = x_1^2 + 2x_2^2 - 2x_3^2 - x_2x_3 + 3x_1x_2$ are gradientul continuu în sens Lipschitz în raport cu norma Euclidiană. Să se determine constanta Lipschitz L.

Problema 6. Se consideră următoarea problemă de optimizare:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & f(x_1, x_2) \\ \text{s.l.:} \ & 2x_1 + x_2 \geq 1, \ x_1 + 3x_2 \geq 1 \\ & x_1 \geq 0, \ x_2 \geq 0. \end{aligned}$$

Să se traseze grafic mulţimea fezabilă. Pentru următoarele funcţii obiectiv, să se determine mulţimea optimă şi valoarea optimă:

(i) $f(x_1, x_2) = x_1 + x_2$. (ii) $f(x_1, x_2) = -x_1 - x_2$.

(iii) $f(x_1, x_2) = x_1$. (iv) $f(x_1, x_2) = \max\{x_1, x_2\}$.

(v) $f(x_1, x_2) = x_1^2 + 9x_2^2$.

Problema 7. Se consideră problema de optimizare:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & (x_1 - 4)^2 + (x_2 - 2)^2 \\ \text{s.l.:} \ & 4x_1^2 + 9x_2^2 \leq 36, \quad 2x_1 \geq -3 \\ & x_1^2 + 4x_2^2 = 4. \end{aligned}$$

(i) Să se reprezinte grafic schema mulţimii fezabile, mulţimile izonivel ale funcţiei obiectiv şi identificaţi punctul de optim pe grafic.

(ii) Să se rezolve din nou punctul (i) unde în enunţul problemei minimizarea se înlocuieşte cu maximizarea.

Problema 8. O fabrică furnizează patru tipuri de produse. Unul dintre materialele brute necesare pentru fabricaţie se află în cantitate redusă, fiind disponibilă numai o cantitate R de material. Preţul de vânzare al produsului i este S_i per kilogram. Mai mult, fiecare kilogram de produs i consumă o cantitate a_i de material brut (cel în cantitate redusă). Costul

de producţie a x_i kilograme din produsul i, excluzând costul materialului în cantitate redusă, este de $k_i x_i^2$, unde $k_i > 0$ este cunoscut. Să se dezvolte un model matematic de optimizare pentru problemă şi apoi să se rezolve cu una din funcţiile Matlab precizate anterior.

Problema 9. Se presupune că cererile $d_1, \ldots, d_n$ pentru un anumit produs în intervalul a n perioade de timp sunt cunoscute. Cererea în timpul perioadei j poate fi satisfăcută din cantitatea produsă x_j pe parcursul perioadei sau din stocul magaziei. Orice exces de producţie poate fi stocat în magazie. Cu toate acestea, magazia are o capacitate limitată K, iar costul stocării unei unităţi de la o perioadă la alta este c. Costul producţiei pe parcursul perioadei j este $f(x_j)$, pentru $j = 1, \ldots, n$. Dacă stocul iniţial este I_0, să se formuleze problema ca un program neliniar.

Problema 10. Într-o localitate se urmăreşte amplasarea optimă a unui număr de depozite în vecinătatea magazinelor. Fie un număr de n magazine, cu poziţii şi cereri cunoscute. Magazinul i se află în poziţia (a_i, b_i) şi dispune de o cerere (grad de solicitare a produselor) r_i. Cererile vor fi satisfăcute cu ajutorul a m depozite cu capacităţi cunoscute. Folosim următoarele notaţii: (x_i, y_i) reprezintă poziţia necunoscută a depozitului şi c_i este capacitatea depozitului i pentru orice $i = 1, \ldots, m$; d_{ij} este distanţa de la depozitul i la magazinul j şi w_{ij} reprezintă unităţile transportate de la depozitul i la magazinul j pentru orice $i = 1, \ldots, m$ şi $j = 1, \ldots, n$.
Să se modeleze problema de găsire a poziţiilor optime corespunzătoare depozitelor. Pentru măsurarea distanţelor în plan, se poate folosi norma Euclidiană (norma 2) sau orice altă normă, cu condiţia să fie precizată.

Problema 11. Fie matricele $X, Y, Z \in \mathbb{R}^{n\times n}$. Să se arate egalitatea:

$$\mathrm{Tr}(XYZ) = \mathrm{Tr}(ZXY) = \mathrm{Tr}(YZX).$$

Problema 12. Fie matricea simetrică $Q \in \mathbb{R}^{n\times n}$. Să se determine expresia valorilor proprii extreme $\lambda_{\min}$ şi $\lambda_{\max}$ sub forma valorilor optime corespunzătoare unor probleme de optimizare.

Problema 13. Fie $x \in \mathbb{R}^n$, pentru norma $\|\cdot\|$, definim norma duală: $\|x\|^* = \min\limits_{\|y\|\leq 1} \langle x, y\rangle$. Să se demonstreze următoarele relaţii, pentru orice $x \in \mathbb{R}^n$:

$$\|x\|_1^* = \|x\|_\infty, \quad \|x\|_\infty^* = \|x\|_1, \quad \|x\|_2^* = \|x\|_2.$$