

Hemtentamensuppgift 1: Linjär programmering och Simplex-metoden

Niclas Börlin

10 april 2001

Målet med denna uppgift är att ni ska få implementera simplex-algoritmen i Matlab och se att det ibland räcker med en enkel algoritm för att lösa många svåra problem.

1 Simplex-metoden

- Implementera simplex-algoritmen i Matlab för lösning av problemet

$$\begin{aligned} \min_x \quad & c^T x \\ \text{då} \quad & Ax \geq b. \end{aligned}$$

Funktionen ska ta följande inparametrar: matrisen A , vektorn b , gradientvektorn c som beskriver problemet, ett starthörn x_0 , en tolerans eps , och maximalt antal iterationer n_{Max} . Toleransen behövs eftersom det p g a avrundningsfel, etc., inte går att utföra jämförelsen $A*x==b$ för likhet, utan man måste i stället skriva $\text{abs}(A*x-b) < \text{eps}$. På samma sätt måste $A*p < 0$ skrivas om till $A*p < -\text{eps}$. Varvtalsbegränsningen n_{Max} behövs för att hindra att funktionen går in i en oändlig loop.

- Funktionens utparametrar skall minst vara lösningen x , antal iterationer n och en vektor λ med Lagrange-multiplikatorer i lösningpunkten. För att underlätta felsökning rekommenderas att ni "spårar" $x^{[k]}$ och $\lambda^{[k]}$ och returnerar två matriser xx och ll med $x^{[k]}$ resp $\lambda^{[k]}$ lagrade kolumnvis.
- Funktionen ska kontrollera att given startpunkt är ett tillåtet hörn.
- Er algoritm ska kunna hantera degenererade hörn. Om ni kommer fram till ett degenererat hörn med $m \geq n$ aktiva bivillkor, så plocka ut n bivillkor slumpmässigt. Fel delmängd kommer att ge $\alpha = 0$, men algoritmen fungerar fortfarande.
- För att ta fram ett tillåtet starthörn kan ni göra något av följande:
 - Plocka ut n st rader ur A och b och lös för x . Kontrollera om punkten är tillåten. Annars välj ut n andra rader och upprepa.
 - Använda den utdelade funktionen `allcorners`, som ger er alla hörn till ett problem $Ax \geq b$. **OBS! Rekommenderas endast på små problem.**
 - Om ni lätt kan finna en tillåten punkt, använd utdelad funktion `findvertex` för att finna tillåtet hörn.

2 Tvåfas-simplex

Implementera en funktion som löser startproblemet i tvåfas-simplexmetoden, dvs givet problemet

$$\begin{aligned} \min_x \quad & c^T x \\ \text{då} \quad & Ax \geq b \end{aligned}$$

och x_0 ej tillåten punkt, formulerar och löser

$$\min_{x,v} \sum_{i=1}^m v_i$$

då

$$\begin{bmatrix} A & I_m \\ 0 & I_m \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \geq \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Ni får anropa er egen simplex-funktion samt `findvertex` för att lösa detta problem.

3 Grundalgoritm för Simplex-metoden

Antag att problemet saknar degenererade hörn och låt $x^{[k]}$ beteckna den aktuella punkten efter n iterationer. Givet ett hörn $x^{[k]}$ blir då algoritmen för simplex-metoden:

- Bestäm en tillåten nedförsriktning p i $x^{[k]}$:
 - Identifiera den aktiva mängden $A_{\mathcal{A}}$ i $x^{[k]}$.
 - Lös $A_{\mathcal{A}}^T \lambda = c$.
 - Välj i sådan att $\lambda_i < 0$. Om alla $\lambda_i \geq 0$, avsluta.
 - Bestäm sökriktningen p som lösningen till $A_{\mathcal{A}} p = e_i$.
- Bestäm steglängden $\alpha = \min_j \sigma_j$, där

$$\sigma_j = \begin{cases} \frac{-r_j(x^{[k]})}{a_j^T p} & \text{om } j \in \mathcal{D} \\ +\infty & \text{annars} \end{cases}.$$

- $x^{[k+1]} = x^{[k]} + \alpha p$.

Deluppgifter

- (10 poäng)
Implementera simplexmetoden beskriven i sektion 1 och 3. Koden behöver endast fungera om given startpunkt är ett tillåtet hörn. Redovisa kod och algoritmer. Strategin för val av negativ Lagrange-multiplikator i steg 1c ska ingå. Redovisa testkörningar på ett "enkelt" problem, t ex The Diet problem eller A Work Scheduling problem. Redovisa problemformuleringar samt dessa omskrivna till standardform.
- (10 poäng)
Implementera tvåfas-simplex beskriven i sektion 2. Redovisa kod och algoritmer. Koden ska fungera oberoende av startpunkt. Redovisa testkörningar på samma problem som i uppgift a men med startpunkter som visar att er tvåfas-kod fungerar.
- (4 poäng)
Redovisa testkörningar på ett "svårare" problem med minst ett likhetsbivillkor, t ex transshipmentproblemet eller shortest-path-problemet. Redovisa problemformuleringar samt dessa omskrivna till standardform. Om ni ej löst uppgift b behöver ni endast starta i tillåtna hörn, annars ska ni också starta i icke tillåtna punkter.
- (3 poäng)
Utred eller resonera kring: Hur påverkar likhetsbivillkor antal iterationer? Ledning: Likhetsbivillkor introducerar degenererade hörn.
- (3 poäng)
Utred eller resonera kring: Hur påverkar valet av starthörn antalet iterationer?