Föreläsning 7: Linjärprogrammering

Exempel:
Maximera

$$2x_1 - x_2 + x_3$$

under villkoren

$$\begin{cases} x_1 + x_2 \leq 1 \\ -2x_1 + x_3 \leq 2 \\ x_1 + x_2 + x_3 \leq 5 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Ett linjärprogrammeringsproblem på *kanonisk form* har följande form:
A är en $m \times n$-matris, $\bar{b}$ en n-vektor och $\bar{c}$ en m-vektor.
Maximera

$$\bar{c}^T \bar{x}$$

under bivillkoren

$$\begin{cases} A\bar{x} \leq \bar{b} \\ \bar{x} \geq \bar{0} \end{cases}$$

Många problem kan på ett naturligt sätt uttryckas som linprog-problem.
Exempel: Flödesproblem.
Vi låter x_e vara värdet på flödet i kant e. Vi har då villkoren $0 \leq x_e \leq c(e)$ för alla kanter e. För varje nod x utom s och t gäller

$$\sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e$$

Vi sätter

$$v = \sum_{e \in Ut(s)} x_e$$

Problemet kan då skrivas som
Maximera v
under villkoren

$$\begin{cases} v = \sum_{e \in Ut(s)} x_e & \\ \sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e & \text{för alla x utom s,t} \\ 0 \leq x_e \leq c(e) & \text{för alla kanter} \end{cases}$$

Dietproblemet: Ett klassiskt exempel är problemet för en person som skall köpa mat. Han kan välja mellan n matprodukter. Vi antar att varje produkt har ett visst innehåll av var och en av m livsnödvändiga näringsämnen. Låt

a_{ij} = innehåll av näringsämne i i matprodukt j
b_i = årsbehovet av näringsämne i.

x_j = årskonsumtionen av matprodukt j.
c_j = styckkostnaden för matprodukt j.

Problemet för konsumenten är då att minimera årskostnaden samtidigt som näringsbehovet tillgodoses. Det ger:

Minimera

$$\bar{c}^T \bar{x}$$

Under villkoren

$$\begin{cases} A\bar{x} \geq \bar{b} \\ \bar{x} \geq \bar{0} \end{cases}$$

Lägg märke till att dessa problem inte passar i standardformen. I första problemet har vi inte bara har olikheter utan även likheter bland villkoren. I andra problemet har vi en minimering istället för en maximering och olikheter vända åt fel håll. Men man kan lätt ordna om så man får standardform. Följande exempel visar tekniken.
Exempel:
Minimera

$$x_1 + 2x_2 - x_3$$

under villkoren

$$\begin{cases} x_1 + x_3 = 1 \\ x_2 - x_3 \geq 3 \end{cases}$$

Man kan byta minimering mot maximering genom att byta tecken på funktionen.
Olikheter "år fel håll" kan ändras genom teckenbyten.
Likheter kan göras om genom att man inför två olikheter.
I vårt fall får man
Maximera

$$-x_1 - 2x_2 + x_3$$

under villkoren

$$\begin{cases} x_1 + x_3 \leq 1 \\ -x_1 - x_3 \leq -1 \\ x_3 - x_2 \leq -3 \end{cases}$$

Linprog-problem går att lösa med **Simplexmetoden**. Den är inte polynomiell (!) men fungerar i medeltal tillfredsställande.
Det finns dock polynomiella algoritmer för Linprog-problem t.ex. ellipsoid-algoritmen och Karmakars algoritm.

Reduktioner

Vi har ett problem A med indata x. Vi har ett annat problem B A kan reduceras till B om vi har en funktion $R(x)$ som givet x bildar indata $R(x)$ till B. Vi använder B:s lösningsalgoritm med indata $R(x)$. Låt oss anta att vi får svaret y. Detta skall då vara svaret till A också om reduktionen är korrekt. (Eventuellt kan man tillåta att svaret på A är $Q(y)$ där Q är någon känd funktion.)
För att reduktionen skall vara bra krävs att R är *enkel*.

Reduktion av problem till LP-problem

Exempel: Hitta kortaste väg $s \to t$ i viktad graf G.

Maximera d_t

Under villkoren

$$\begin{cases} d_u \le d_v + w(u,v) & \text{för alla kanter}(u,v) \\ d_s = 0 \end{cases}$$

Duala formen

Givet ett LP-problem P på standardform (med maximering) kan det s.k. it duala problemet P^* ställas upp:
Minimera

$$\bar{b}^T \bar{y}$$

under villkoren

$$\begin{cases} A^T \bar{y} \ge \bar{c} \\ \bar{x} \ge \bar{0} \end{cases}$$

En känd sats säger att lösningen (det optimala värdet) till det *primala* problemet P är samma som lösningen till det *duala* problemet P^*.
Om P inte är på standardform blir översättningen primalt-dualt inte lika riktigt lika enkelt.

Exempel på dualt problem

Flödesproblemet kan översättas på dual form:
Vektorn $\bar{y}$ består av $|V| + |E|$ tal. De är g_i för varje nod v_i och γ_j för varje kant e_j.
Minimera

$$\sum_j \gamma_j c_j$$

under villkoren

$$\begin{cases} g_i - g_j + \gamma_k \ge 0 & \text{om} \quad e_k = (v_i, v_j) \\ g_n - g_1 \ge 1, \quad \gamma_j \ge 0 & \text{för alla} \quad j \end{cases}$$

Lösningen till detta problem motsvarar ett minimalt snitt $(S, V - S)$ och en tilldelning av värden $g_i = 0$ om $v_i \in S$, $g_i = 0$ annars. $\gamma_j = 1$ om e_j går från S till $V - S$, $\gamma_j = 0$ annars.

Översättning av dietproblemet till dual form ger

Maximera

$$\bar{b}^T \bar{y}$$

under villkoren

$$\begin{cases} A^T \bar{y} \leq \bar{c} \\ \bar{x} \geq \bar{0} \end{cases}$$

Hur kan vi tolka $\bar{y}$? En möjlighet är att tänka sig att vi har en ett läkemedelsbolag som vill tillverka piller som innehåller de olika näringsämnena. Det betyder att vi har m olika piller. Vi antar att de alla innehåller en enhet av näringsämnena. Vilket pris y_i skall vi sälja pillret som ger ämne i till? Vi vill att pillren skall vara konkurrenskraftiga gentemot t.ex. matprodukt j. Det ger $\sum_i y_i a_{ij} \leq c_j$ för alla j. (Annars vore det bättre för en konsument att delvis köpa matprodukt j.) Samtidigt vill bolaget maximera sin vinst som måste vara $\bar{b}^T \bar{y}$.

Översättningen av kortaste vägen-problemt till dual form ger:

Maximera

$$\sum_e x_e w(e)$$

under villkoren

$$\begin{cases} 1 = \sum_{e \in Ut(s)} x_e & \\ \sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e & \text{för alla x utom s,t} \\ 0 \leq x_e \leq 1 & \text{för alla kanter} \end{cases}$$

Genom att studera samband mellan primala problemet och det duala kan man ofta utveckla algoritmer som är effektivare än simplexalgoritmen. Maxflödesalgoritmen är ett exempel på en sådan algoritm.