Optimering 2013 - Noter

Mathias Bak Bertelsen

31. marts 2013

Start med at løse de mekaniske opgaver, da modelleringsopgaverne tager længere tid.

Indhold

1 Max flow

A flow must satisfy the following properties

- **Capacity constraint:** For all $u, v \in V$, we require $f(u,v) \le c(u,v)$.
- **Skew symetry:** For all $u, v \in V$, we require $f(u,v) = -f(v,u)$.
- **Flow conservation:** For all $u, v \in V - \{s,t\}$, we require

$$\sum_{v \in V} f(u,v) = 0$$

1.1 Cuts

Net flow across a cut is calculated by adding all the flow from S to T, and subtracting the flow from T to S.

Capacity cut is calculated by adding the capacities on all arcs going from S to T and ignoring the arcs from T to S.

2 Min cost flow

Opgaveforklaring: Forklar hvad flowet er og hvad nodes betyder. Beskriv hvad costs og balances er med mindre det vises på grafen. Referer til ”Integrality theorem” og forklar hvorfor det er vigtigt at resultatet er integral.

Augmenting cycle: A cycle where the total cost is negative. It must not repeat any vertices.

Negative capacity arc: An arc with negative capacity must me saturated for a flow to be feasible. A negative capacity arc from node A to node B yields a lower bound for the flow from B to A.

Balance: The balance on a node N means that N produces that amount of flow. A negative balance means consumption of flow.

3 Linear Programming

Basis: The vairables on on the left hand side of a dictionary. This is initially the slack variables.
Leaving variable: The variable that leaves the basis when we pivot.
Entering variable: The variable that enters the bases when we pivot.

3.1 The Simplex Method

Introducer slack variabler: Ryk hele udtrykket over på højresiden af uligheden. Udskift derefter venstresiden med en slackvariabel og lav uligheden til en lighed. Objektfunktionen ændres ikke.

Hver constraint	$ax_1 + bx_2 + cx_3 \leq k$
Bliver til	$x_4 = k - ax_1 - bx_2 - cx_3$

Find en feasible solution: I en feasible solution skal alle variable være større end 0 (hvis programmet er på standardform). Forsøg først med 0. Hvis 0 ikke er en løsning, se s. 39 i Vasek Chvatal.

Pivot: I det følgende er entering variable x_e og leaving variable x_l.

- **Vælg entering variable:** Vælg variablen med det laveste index der er positiv i objektfunktionen.
- **Vælg leaving variable:** For hver constraint hvor x_e er negativ beregn hvor meget den kan hæves. **Eksempel:** Hvis x_2 er entering variable i eksemplet ovenfor, vil x_2 kunne hæves til k/b. Rækken der kan hæves mindst (mest stringente) er pivotrækken.
- **Udfør pivot:** Træk x_l over på højresiden og x_e over på venstresiden. Divider derefter hele højresiden med konstanten foran x_e. Derefter fjernes x_e i alle de andre ligninger inklusive objektfunktionen vha. substitution.

Termination: Algoritmen terminerer når alle variable i objektfunktionen har en negativ konstant.

3.2 Det Duale Program

Find det duale program: Antallet af variable i det primale program er lig med antallet af constraints i det duale. Antallet af constraints i det primale program er lig med antallet af variable i det duale.

- maximization programmer bliver til minimization og omvendt

- Koefficienterne for x_1 bliver koefficienterne i den første constraint i det duale. Koefficienterne for x_2 bliver koefficienterne i den anden constraint i det duale og så videre.
- Alle uligheder vendes
- Højresiden af ulighederne er givet ved koefficienterne i objektfunktionen
- Objektfunktionen fåes ved at bruge konstanterne på højresiden af constraints som koefficienter til variablene

Eksempel:
Primalt program

$$\begin{aligned} \text{maximize} \quad & 100x_1 + 300x_2 \\ \text{subject to} \quad & 3x_1 + 6x_2 \leq 40 \\ & -x_1 + 3x_2 \leq 0 \\ & x_1 + 4x_2 \leq 16 \end{aligned} \tag{1}$$

Dualt program

$$\begin{aligned} \text{minimize} \quad & 40y_1 + 16y_3 \\ \text{subject to} \quad & 3y_1 - y_2 + y_3 \geq 100 \\ & 6y_1 + 3y_2 + 4y_3 \geq 300 \end{aligned} \tag{2}$$

3.3 Dual løsning fra primalt program

Eksempelløsning på primalt program

$$\begin{aligned} x_1 &= 1 + 4x_2 + x_4 - 3x_6 - 9x_7 \\ x_5 &= 2 + x_2 + 2x_4 + 2x_6 + 2x_7 \\ x_3 &= 3 + 3x_2 + x_4 + x_6 + 3x_7 \\ z &= -3x_2 - 3x_4 - 4x_6 - 7x_7 \end{aligned} \tag{3}$$

Løsningen på det duale program $\vec{y}$ er givet ved at negere koefficienterne på slackvariablerne i objektfunktionen z. Løsningen på det duale program til eksempel (3) er altså $\vec{y} = (0, 4, 7)$.

3.4 Standard form

- The objective must be to *maximize* a linear function
- All variables are restricted to *non-negative* values
- All constraints have to be on the form $d_1x_1 + d_2x_2 + \cdots + d_nx_n \leq e$

4 Integer Linear Programming

Relaxing: Remove the integrality constraints

4.1 The Branch-and-bound algorithm

Step 1: Relax the problem
Step 2: Solve the relaxed program, gives $(x*, v*)$ where $x*$ is the solution and $v*$ is the value of the objective function
Step 3: Branch on the variable that is fractional and repeat
Stop: When all the values are integral

Greedy algorithm: For knapsack problems use a greedy algorithm to solve the relaxed problem. First calculate the efficiency for every item which is value over weight ($E = V/W$). The greedy algorithm starts with the item with the highest effeciency and fills the knapsack. Repeat until there are no more space.

Når vi har med max af en funktion at gøre i objektfunktionen er løsningen at indføre en ny variabel Z og lave constraints for hver funktion inde i max. Disse skal være mindre end den nye variable Z og Z vil derefter blive vores nye objektfunktion.

4.2 Fjern "max" og "min" fra objektfunktion

Når vi har med max af en funktion at gøre i objektfunktionen er løsningen at indføre en ny variabel Z og lave constraints for hver funktion inde i max. Disse skal være mindre end den nye variable Z og Z vil derefter blive vores nye objektfunktion.

Eksempel:

$$\text{minimize} \quad \max_{1 \le j \le m} \sum_{i=1}^{n} c_{ij} x_j$$

Dette vil blive til

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & z - \sum_{i=1}^{n} c_{ij} x_j \geq 0 \quad \text{for } j = 1, \dots, m \\ & \text{z is unrestricted} \end{array}$$

Metoden er den samme for "min", dog skal uligheden vendes så den constraint der tilføjes er

$$z - \sum_{i=1}^{n} c_{ij} x_j \leq 0 \quad \text{for } j = 1, \dots, m$$

5 Matrix Games

http://www.math.ucla.edu/~tom/Game_Theory/mat.pdf

5.1 Saddelpunkt (giver spillets værdi med det samme)

Et saddelpunkt er en indgang i matricen der er sin rækkes minimum og sin søjles maximum. Værdien af saddelpunktet er spillets værdi og en optimal strategi er altid at vælge den række/søjle.

Saddelpunktet findes ved for hver række at se om det mindste tal også er det største tal i sin søjle. Det er ok hvis andre tal er lige så store, bare ikke større.

Eksempel: Her er indgang A_{12} saddelpunkt.

$$A = \begin{bmatrix} 5 & 3 & 4 \\ 2 & 1 & 1 \\ 6 & 1 & 2 \end{bmatrix} \tag{4}$$

5.2 Løsning af 2x2 matrix games

$$A = \begin{bmatrix} a & b \\ d & c \end{bmatrix}$$

1. **Test om der er et saddelpunkt**
2. **Hvis der ikke er et saddelpunkt, benyt følgende formler**

Rækkespillerens optimale strategi er $[p, 1-p]$, og søjlespillerens optimale strategi er $[q, 1-q]$. v er spillets værdi.

$$\begin{aligned} p &= \tfrac{c-d}{(a-b)+(c-d)} \\ q &= \tfrac{c-b}{(a-b)+(c-d)} \\ v &= ap + d(1-p) \end{aligned}$$

5.3 Fjern dominerede strategier

- Række a dominerer række b hvis alle indgange i a er **større end eller lig med** tilsvarende indgang i b.
- Søjle a dominerer søjle b hvis alle indgange i a er **mindre end eller lig med** tilsvarende indgang i b.

Eksempel: Søjle 2 dominerer søjle 3 da alle indgange er mindre. I den nye matrix dominerer række 3 række 1 da alle indgange er større.

$$\begin{bmatrix} 2 & 0 & 4 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 4 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 \\ 4 & 1 \end{bmatrix} \tag{5}$$

- En række (søjle) må fjernes hvis den er domineret af en sandsynligheds-fordeling af andre rækker (søjler).

Eksempel: Søjle 1 og 3 dominerer søjle 2 hvis hver har sandsylinghed 1/2. Herefter dominerer række 1 og 3 række 2 med sandsylinghed henholdsvist 1/3 og 2/3.

$$\begin{bmatrix} 0 & 4 & 6 \\ 5 & 7 & 4 \\ 9 & 6 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 6 \\ 5 & 4 \\ 9 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 6 \\ 9 & 3 \end{bmatrix} \tag{6}$$

5.4 Løsning af $2 \times n$ og $m \times 2$ matrix games

Denne type løses grafisk.

5.4.1 $2 \times n$ matrix games

Eksempel:

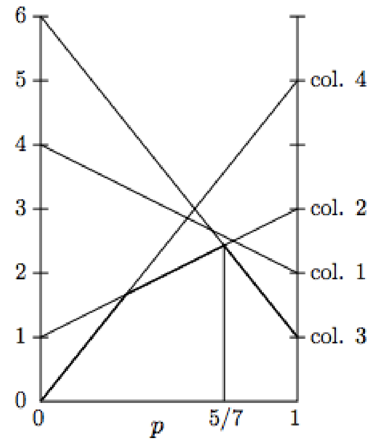
$$\begin{array}{r} p \\ p-1 \end{array} \begin{bmatrix} 2 & 3 & 1 & 5 \\ 4 & 1 & 6 & 0 \end{bmatrix} \tag{7}$$

Payoff funktionerne for denne matrix hvor række 1 vælges med sandsynlighed p og række 2 vælges med sandsynlighed $1-p$ er

$$2p + 4(1-p) \qquad \text{(col. 1)}$$
$$3p + 1(1-p) \qquad \text{(col. 2)}$$
$$1p + 6(1-p) \qquad \text{(col. 3)}$$
$$5p \qquad \text{(col. 4)}$$

Disse funktioner plottes i en graf hvor p er på den vandrette akse og *payoff* er på den lodrette. Husk at $0 \leq p \leq 1$.

Funktionerne plottes ved at sætte $p = 0$ og $p = 1$ og tegne en streg imellem. **Tip: når $p = 0$ er værdien tallet fra anden række i matricen og når $p = 1$ er værdien tallet fra første række.**



Figur 1: Plot af matricen (7)

p **er det højeste punkt under alle linjer.** Skæringen beregnes ved at sætte de skærende linjers ligninger lig med hinanden. I dette tilfælde

$$3p + 1(1-p) = 1p + 6(1-p) \Leftrightarrow p = 5/7$$

Værdien af spillet beregnes ved at indsætte værdien af p i en af de to ligninger der skar hinanden i $p = 5/7$, altså (col. 2) eller (col. 3) i dette eksempel.

$$v = 3p + 1(1-p) = 3(5/7) + 1(1-(5/7)) = 17/7$$

Den optimale strategi for rækkespilleren er altså $[\frac{5}{7}, \frac{2}{7}]$. Søjlespillerens optimale strategi findes ved at løse den 2×2 matrix der opstår ved at fjerne alle andre søjler end de to skærende som var (col. 2) og (col. 3). Det giver denne matrix

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 6 \end{bmatrix}$$

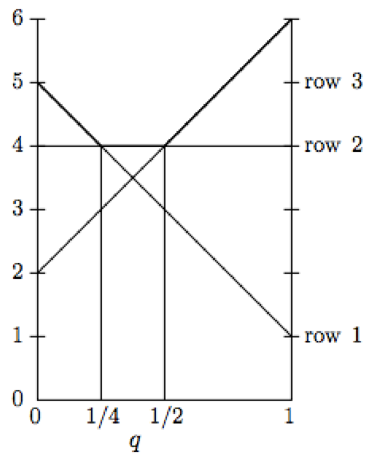
Den optimale strategi for søjlespilleren bliver da $[0, \frac{5}{7}, \frac{2}{7}, 0]$.

5.4.2 $m \times 2$ matrix games

Løsningen på disse er meget lig løsningen på $2 \times n$ matrix games. Tegn grafen, men for q i stedet for p (søjlespillerens løsning). Forkellen på grafen er at man skal fidne det **laveste** punkt der er **over** alle linjerne. Derefter løses den resulterende 2×2 matrix for at finde rækkespillerens optimale løsning.

Eksempel:

$$\begin{array}{cc} q & 1-q \\ \end{array}$$
$$\begin{pmatrix} 1 & 5 \\ 4 & 4 \\ 6 & 2 \end{pmatrix} \tag{8}$$



Figur 2: Plot af matricen (8)

Det ses at q har flere løsninger. $1/4 \leq q \leq 1/2$ er alle optimale løsninger.