Videregående algoritmik

# Projektopgave 2

David Pisinger, Blok 2, 2007-08

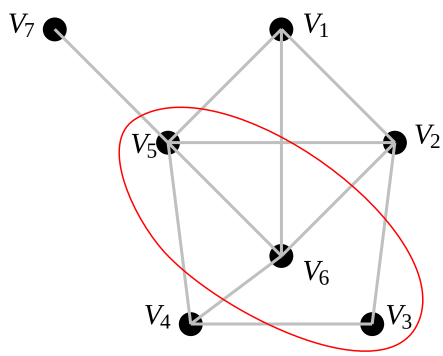
Dette er den anden obligatoriske projektopgave på kurset “Videregående algorimik”. Opgaven stilles 11. december 2007 og skal afleveres senest 9. januar 2008 kl. 12.00 i DIKU’s førstedelsadministration. For at blive godkendt skal der være gjort et reelt forsøg på at løse samtlige spørgsmål. Besvarelsen skal udarbejdes i grupper på to til tre deltagere. Grupper med 1 deltager kræver accept fra instruktoren. Læs venligst hele opgaveformuleringen igennem inden du går i gang. Hints til opgaverne kan fås ved øvelserne, hvor der er afsat en øvelsesgang til at arbejde med projektopgaven.

## MAX-CUT problemet

Givet en ikke-orienteret graf $G = (V,E)$ er MAX-CUT problemet defineret som

$$\text{MAX-CUT} = \{< G >: \text{ find et snit } S,T \text{ i grafen } G \text{ som maksimerer antal kanter over snittet}\}$$

F.eks. har følgende instans af MAX-CUT den optimale løsning $S = \{3,5,6\}$, $T = \{1,2,4,7\}$ med løsningsværdi 9.




Det tilhørende afgørlighedsproblem MAX-CUT-DECISION kan formuleres som følger: Lad $n = |V|$ være antal knuder i $G$. Lad de binære variable $x_i$ for $i = 1,\dots,n$, være $x_i = 1$ hvis knude $i \in S$, og $x_i = 0$ hvis knude $i \in T$. Lad endvidere konstanterne $e_{ij} = 1$ hvis kant $(i, j) \in E$, og $e_{ij} = 0$ ellers. Sidst, angiver $k$ en nedre grænse for antallet af kanter over snittet $(S,T)$.

$$\text{MAX-CUT-DECISION} = \left\{ < G,k > : \text{der findes} (x_1,\dots,x_n) \in \{0,1\}^n \text{så} \sum_{i\in V} \sum_{j\in V} e_{ij} x_i (1-x_j) \geq k \right\}$$

Det er velkendt at maximering af en objektfunktion $f$ over domænet $X$ givet som problemet

$$\max_{x\in X} \ f(x)$$

kan omskrives til et ækvivalent problem der minimerer objektfunktionen

$$\min_{x\in X} \ -f(x)$$

**Opgave 1** Kan denne ækvivalens benyttes til at løse MAX-CUT ved hjælp af MIN-CUT (f.eks. ved at skifte fortegn på alle kantvægte)? Argumenter for dit svar. ■

## Kvadratisk 0-1 programmering

Betragt det kvadratiske 0-1 optimeringsproblem QP:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in N}\sum_{j\in N} d_{ij}x_i x_j \\ \text{subject to} \quad & x_j \in \{0,1\}, \quad j \in N \end{aligned} \tag{1}$$

defineret på mængden $N = \{1,\ldots,n\}$, og med $d_{ij} \in \mathbb{R}$ for $i, j \in N$. Uden tab af generalitet kan man antage at $(d_{ij})$ matricen er symmetrisk. Følgende eksempel viser en instans med $n = 4$ variable givet ved $(d_{ij})$ matricen

| $i$ \ $j$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 3 | 0 | 1 | 2 |
| 2 | 0 | -9 | 1 | 3 |
| 3 | 1 | 1 | -5 | 2 |
| 4 | 2 | 3 | 2 | 0 |

Den optimale løsning er at vælge $x_1 = x_3 = x_4 = 1$ hvilket giver en løsningsværdi på 8.

**Opgave 2** Omformuler QP til et afgørlighedsproblem QP-DECISION. ■

**Opgave 3** Vis at QP-DECISION ligger i klassen $\mathcal{NP}$. ■

**Opgave 4** Vis at QP-DECISION er $\mathcal{NP}$-fuldstændig ved reduktion fra MAX-CUT-DECISION. ■

## Nemme instanser af kvadratisk 0-1 programmering

Selv om QP-DECISION er $\mathcal{NP}$-fuldstændigt betyder det som bekendt ikke at alle instanser er svære at løse. Betragt følgende klasse af instanser for QP:

- $d_{ij} \in \mathbb{R}_0^+$ for alle $i, j \in N$, $i \neq j$
- $d_{ii} \in \mathbb{R}$ for alle $i \in N$

dvs. alle elementer udenfor diagonalen i matricen $(d_{ij})$ skal være ikke-negative.

**Opgave 5** Vis at instanser af QP som overholder disse to krav kan løses i polynomiel tid som følger:

a) Konstruer en instans $(V,E,c)$ af MAXIMUM-FLOW ved at sætte $V = N \cup \{s,t\}$ og $E = \{s\} \times N \,\cup\, N \times N \,\cup\, N \times \{t\}$. Kapaciteten af kanterne sættes til:

$$\begin{array}{ll} c_{si} = \max\{0, \sum_{j\in N} d_{ij}\}, & i \in N \\ c_{ij} = d_{ij}, & i, j \in N, i \neq j \\ c_{ii} = 0, & i \in N \\ c_{it} = \max\{0, -\sum_{j\in N} d_{ij}\} & i \in N \end{array}$$

Argumenter for at dette er en gyldig instans af MAXIMUM-FLOW problemet.

b) Antag at den optimale strømning $f^*$ fundet med MAXIMUM-FLOW har strømningsværdien $|f^*|$, og at det tilhørende minimale snit er $S, T$. Lad $x_i = 1$ hvis $i \in S$ og lad $x_i = 0$ hvis $i \in T$. Opskriv værdien af et minimalt snit udtrykt ved disse beslutningsvariable.

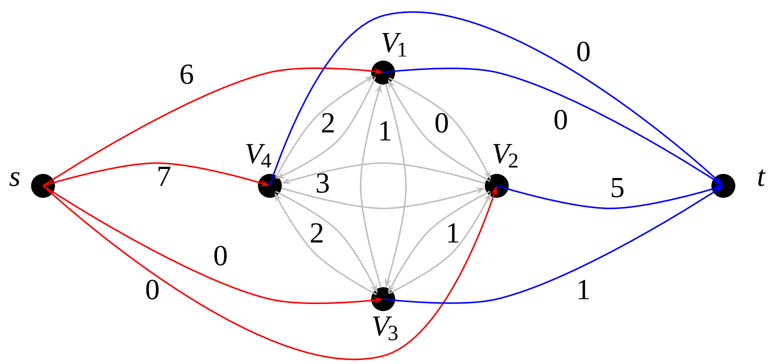
c) Vis at $c_{si} - c_{it} = \sum_{j \in N} c_{ij} + d_{ii}$ for alle $i \in N$.

d) Vis at løsningsværdien til QP kan findes som $\sum_{i \in N} c_{si} - |f^*|$.

e) Angiv køretiden for ovenstående algoritme.

■

Et eksempel på ovenstående transformation med datasættet fra forrige side er:

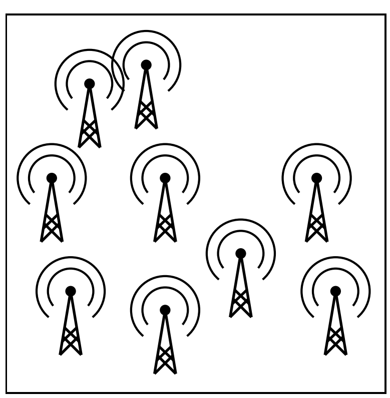



Da $(d_{ij})$ matricen er symmetrisk har de grå kanter parvist samme kapacitet. Den maksimale strømningsværdi i grafen er $|f^*| = 5$, og det tilhørende minimale snit er $S = \{s, 1, 3, 4\}$ og $T = \{t, 2\}$.

**Opgave 6** Implementer en algoritme, som kan løse de ovenstående nemme instanser af QP problemet. Brug algoritmen til at løse instanserne på hjemmesiden. Rapporter den optimale løsningsværdi for hvert af problemerne. ■

## Kvadratisk knapsack problemet

Kvadratisk knapsack problemet har adskillige anvendelser. Betragt f.eks. et teleselskab som skal etablere et antal radiomaster rundt omkring i landet. De $n$ mest interessante byer udvælges, og der gennemføres en nøje analyse af hver radiomasts fordele og ulemper. En radiomast $j$ koster et givet beløb $w_j$ at etablere, og teleselskabet har et budget på $c$. Enhver radiomast vil i sig selv give et vist afkast $p_{jj}$ mens man får afkastet $p_{ij} + p_{ji}$ for telefoni mellem to byer $i$ og $j$.



Idet vi indfører beslutningsvariable $x_j \in \{0,1\}$ til at angive om en radiomast bygges eller ej, kan vi formulere problemet som følgende kvadratiske knapsack problem (QKP):

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n}\sum_{i=1}^{n} p_{ij}x_i x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j x_j \leq c \\ & x_j \in \{0,1\}, \quad j = 1,\dots,n. \end{aligned} \tag{2}$$

hvor objektfunktionen angiver at profitten skal maximeres, mens knapsack begrænsningen betyder at etablerings-budgettet på $c$ ikke må overskrides. Det antages at alle værdier af $p_{ij}$ og $w_j$ er ikke-negative heltal.

Afgørlighedsproblemet svarende til QKP er $\mathcal{NP}$-fuldstændigt, hvilket kan vises ved reduktion fra KNAPSACK (diagonalelementerne i QKP sættes til profitterne i KNAPSACK, mens alle elementer udenfor diagonalen sættes til nul).

Følgende eksempel viser en instans af QKP for syv byer.

| | $i \backslash j$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| | 1 | 8 | 9 | 4 | 3 | 3 | 4 | 7 |
| | 2 | 9 | 7 | 1 | 2 | 6 | 3 | 9 |
| | 3 | 4 | 1 | 8 | 4 | 7 | 5 | 4 |
| $p_{ij}$ | 4 | 3 | 2 | 4 | 1 | 0 | 3 | 7 |
| | 5 | 3 | 6 | 7 | 0 | 2 | 4 | 0 |
| | 6 | 4 | 3 | 5 | 3 | 4 | 8 | 9 |
| | 7 | 7 | 9 | 4 | 7 | 0 | 9 | 5 |
| $w_j$ | | 2 | 7 | 8 | 9 | 5 | 3 | 9 |

$n = 7$, $c = 20$.

Den optimale løsning er at bygge radiomaster i byerne $1,2,5,6$ hvilket giver en profit på 83.

## Øvre grænseværdi

**Opgave 7** Hvis man Lagrange-relaxerer knapsack begrænsningen i (2) fremkommer problemet QKP$(\lambda)$ givet ved

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n}\sum_{i=1}^{n} p_{ij}x_i x_j - \lambda\left(\sum_{j=1}^{n} w_j x_j - c\right) \\ & x_j \in \{0,1\}, \quad j = 1,\dots,n. \end{aligned} \tag{3}$$

For hvilke værdier af $\lambda$ er (3) en relaxering af (2)? ■

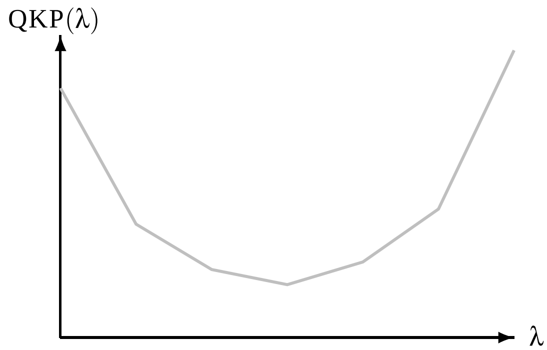
Betegn med $L$ den lovlige mængde af $\lambda$-værdier.

**Opgave 8** Vis at for et givet $\lambda \in L$ kan det Lagrange-relaxerede problem (3) skrives på formen

$$\begin{aligned} \text{maximize} \quad & \left(\sum_{i\in N}\sum_{j\in N} d_{ij}x_i x_j\right) + k \\ \text{subject to} \quad & x_j \in \{0,1\}, \quad j = 1,\dots,n \end{aligned} \tag{4}$$

hvor $k$ er en konstant. Vis endvidere at dette problem kan løses i polynomiel tid. ■

Det vides at $\text{QKP}(\lambda)$ er en konveks funktion af $\lambda$, der ser ud som følger:




Det Lagrange duale problem går ud på at finde den værdi af $\lambda$ som resulterer i den strammeste grænseværdi. Dvs. vi søger

$$\min_{\lambda \in L} \text{QKP}(\lambda) \tag{5}$$

**Opgave 9** Angiv en nedre og øvre grænse for hvor stor $\lambda$ kan blive i ovenstående udtryk. Udtrykket $\lambda < \infty$ er ikke et tilfredsstillende svar. ■

**Opgave 10** Beskriv en effektiv algoritme som løser problemet (5). ■

**Opgave 11** Implementer algoritmen til løsning af (5) og afprøv den på instanserne fra hjemmesiden. ■

# Branch-and-bound algoritme

Vi vil nu konstruere en rekursiv branch-and-bound algoritme til løsning af QKP problemet (2). I hvert skridt forgrener vi på den sidste beslutningsvariabel $x_n$ således at det tilbageværende problem herefter har størrelse $n-1$.

Initielt sættes $x_i = 0$ for $i = 1, \ldots, n$ og $z^* = 0$. Herefter kaldes algoritmen

$$\texttt{quadknap}(0, c, n);$$

Den rekursive del af algoritmen kan skitseres som:

$\texttt{quadknap}(P, \overline{c}, \overline{n})$
  **if** $\overline{c} < 0$ **then return**
  **if** $P > z^*$ **then** $z^* \leftarrow P$; $x_i^* \leftarrow x_i$ for $i = 1, \ldots, n$
  **if** $\overline{n} = 0$ **then return**
  find en øvre grænseværdi $\mathcal{U}$ for problemet defineret på $(p_{ij})$, $i, j = 1, \ldots, \overline{n}$, og med kapacitet $\overline{c}$
  **if** $\mathcal{U} + P > z^*$ **then**
    $x_{\overline{n}} \leftarrow 1$; modificer $(p_{ij})$ matricen; $\texttt{quadknap}(P + p_{\overline{n}\overline{n}}, \overline{c} - w_{\overline{n}}, \overline{n} - 1)$
    $x_{\overline{n}} \leftarrow 0$; retabler $(p_{ij})$ matricen; $\texttt{quadknap}(P, \overline{c}, \overline{n} - 1)$

Her angiver $\overline{n}$ antallet af frie variable (dvs. variable som der endnu ikke er forgrenet på), $\overline{c}$ angiver den tilbageværende kapacitet, mens $P$ angiver profitten af de allerede valgte radiomaster.

Ved modification af $(p_{ij})$ matricen sætter vi

$$p_{ii} \leftarrow p_{ii} + p_{i\overline{n}} + p_{\overline{n}i}$$

for hvert $i = 1, \ldots, \overline{n} - 1$. Dette betyder, at hvis radiomast $i$ på et senere tidspunkt bliver valgt, vil man automatisk indkassere profitten af al kommunikation mellem byerne $i$ og $\overline{n}$. Når matricen skal retableres subtraherer vi atter disse bidrag ved at sætte

$$p_{ii} \leftarrow p_{ii} - p_{i\overline{n}} - p_{\overline{n}i}$$

for hvert $i = 1, \ldots, \overline{n} - 1$.

**Opgave 12** Implementer en fuldstændig version af `quadknap` som benytter grænseværdien (5). ■

**Opgave 13** Kør algoritmen på instanserne fra hjemmesiden og angiv køretid, antal branch-and-bound knuder, grænseværdi i rodknuden, samt optimal løsningsværdi. ■

# Noter

Det kan i opgaven antages at MAX-CUT-DECISION er $\mathcal{NP}$-fuldstændig. $\mathcal{NP}$-fuldstændighed af MAX-CUT-DECISION bliver vist ved forelæsningerne som følger (se plancherne): 3CNF-SAT kan reduceres i polynomiel tid til 3CNF-NAE-SAT. Sidstnævnte problem kan igen reduceres i polynomiel tid til MAX-CUT-DECISION.

Til implementerings-opgaverne benyttes et rammeprogram skrevet i C++. Rammeprogrammet kan indlæse instanser fra hjemmesiden og rummer en simpel implementering af EDMONDS-KARP algoritmen til løsning af MAXIMUM-FLOW problemet. Rammeprogrammet findes på kursets hjemmeside sammen med alle instanser.

De benyttede datatilfælde er en blanding af virkelige data og tilfældigt genererede data som angivet i følgende tabel:

| Beskrivelse | filnavne |
|---|---|
| Radiotelefoni | `tele7` |
| Compiler design | `comp30 comp45 comp47` |
| Geometrisk $p$-dispersion | `geo10 geo20 geo30 geo40` |
| Random 25% densitet | `rand10.25 rand20.25 rand30.25 rand40.25` |
| Random 100% densitet | `rand10.100 rand20.100 rand30.100 rand40.100` |
| Klike i en graf | `clique10 clique20 clique30 clique40` |

Densiteten af et datatilfælde angiver hvor mange procent af matricens elementer der er forskellige fra nul. Compiler design problemerne er foreslået af Helmberg, Rendl, Weismantel.