P2-projektforslag
Kombinatorik: grafteori og optimering*

Vejledere: Bo Hove†, Diego Ruano‡, Jan B. Sørensen§

1. februar 2012

1 Indledning

Temaet for projekter på 2. semester af matematik-studiet og matematik-økonomi-studiet er kombinatorik: grafteori og optimering. Ordet "kombinatorik" har omtrent samme betydning som udtrykket "diskret matematik", der navnet på et af kurserne på 2. semester og den lærebog [1] vi bruger i kurset.

Kombinatorik er en gren af matematikken hvor man studerer endelige strukturer eller strukturer, der er "næsten endelige". Udtrykket kombinatorik bruges ofte i betydningen optælling af elementer i en endelig struktur. Men her bruges ordet altså i en bredere betydning, der omfatter grafteori og kombinatorisk (eller diskret) optimering.

En graf er i denne sammenhæng en endelig struktur, der består af et antal punkter (også kaldet knuder) og forbindelser mellem punkterne. Punkterne tegnes tit som små cirkler. Forbindelserne, der kaldes kanter, tegnes som kurver, der forbinder to punkter. Kanterne kan eventuelt have en orientering (retning). Se figur 1 og 2.

Grafer kan benyttes i forbindelse med matematisk modellering af fysiske netværk som f.eks. elektriske netværk, vejnet, eller lign., men de benyttes også ved modellering af mere abstrakte relationer mellem objekter.

---

*Oprindelig sammensat af Leif K. Jørgensen

†bh@thisted-gymnasium.dk

‡diego@math.aau.dk

§js@aalborgstudenterkursus.dk

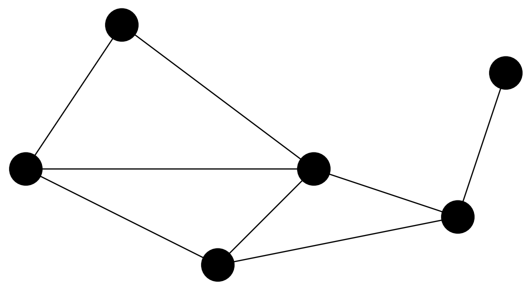

Figure 1: En ikke-orienteret graf

På figur 3 vises f.eks. en graf hvor hvert punkt svarer til en mængde af tal. To punkter er forbundet med en kant hvis fællesmængden af de to tilhørende mængder er tom.

Et andet eksempel er en graf, der beskriver relationer mellem et antal personer. Her vil hvert punkt svare til en person og der er en kant mellem to personer, der kender hinanden.

Veje og kredse er begreber der ofte bruges i studiet af grafer. En vej fra et punkt A til et punkt B består af en række kanter som man kan følge for at komme fra A til B. I en orienteret graf skal man naturligvis følge kanternes retning. Der kan desuden være krav om at hver kant kun bruges én gang eller at vejen ikke må gå gennem et punkt to gange. Den orienterede graf i figur 2 har f. eks. 4 veje fra A til B.

En kreds er en vej der vender tilbage til det punkt hvor den startede – altså $B = A$.

Optimering handler om at bestemme maksimum eller minimum for en funktion. F.eks. kender I en metode til bestemmelse af maksimum og minimum for en differentiabel funktion $f : \mathbb{R} \mapsto \mathbb{R}$. I kombinatorisk optimering skal vi finde maksimum eller minimum for en funktion $f : S \mapsto \mathbb{R}$, hvor S er en endelig (eller "næsten" endelig) mængde med en bestemt struktur. S kan f.eks. være mængden af veje fra A til B i en graf. Man er så interesseret i at finde den korteste vej fra A til B. I optimering beskæftiger man sig ofte med vægtede grafer. Det vil sige at hver kant har tilknyttet et tal (en vægt), som f.eks. kan angive længden af kanten. Længden af en vej vil så være summen af længder af de kanter, der gennemløbes.

I optimering vil vi være interesserede i at finde systematisk fremgangmåde til at bestemme en optimal løsning. En sådan fremgangsmåde kaldes en

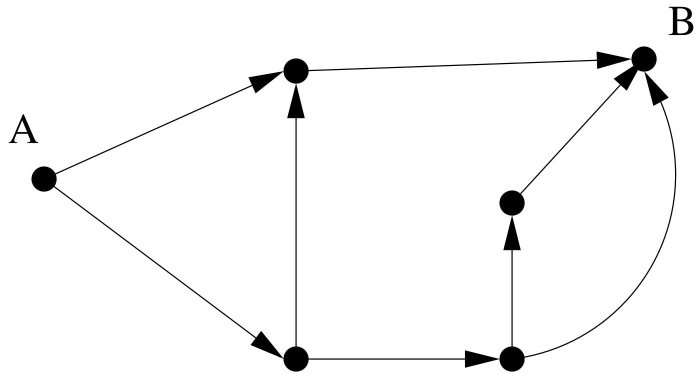

Figure 2: En orienteret graf

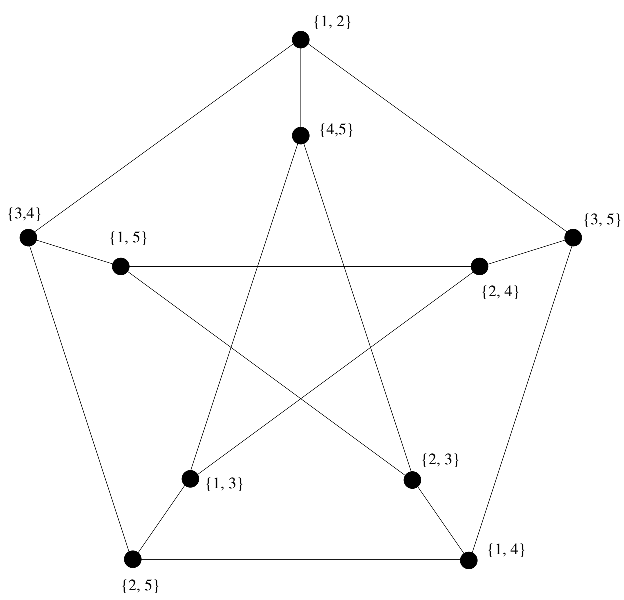

Figure 3: Petersen grafen

algoritme. Når vi så har konstrueret en algoritme, der finder f.eks. korteste veje i en graf, så vil det være interessant at vide hvor mange udregninger, der kræves for at finde en optimal løsning. Antallet af udregninger vil blandt andet afhænge af grafens størrelse. En øvre grænse for antal udregninger kan altså angives som en funktion af antallet af punkter i grafen. Den funktion kaldes algoritmens kompleksitet. Hvis en algoritme bygger på "afprøvning af alle muligheder" – f.eks. find alle veje fra A til B og bestem længden af hver af dem, så vil algoritmens kompleksitet måske være af størrelsesorden 2^n, hvor n er antal punkter i grafen. En sådan algoritme kan i praksis kun anvendes på små grafer. Men hvis man finder en algoritme, der har kompleksitet f.eks. n^2, så kan man også løse optimeringsproblemet for ret store grafer. Algoritmer og kompleksitet er altså vigtige begreber i forbindelse med optimering.

På P2 skal I arbejde med projekter inden for grafteori og optimering.

- Matematik-økonomi studerende skal vælge et projektemne inden for optimering, som giver mulighed for at se på anvendelser i økonomi.
- Matematik studerende kan enten vælge et projektemne inden for optimering eller et emne som er et teoretisk studie af grafer og deres struktur.

Projektarbejdet skal resultere i udarbejdelsen af en P2-rapport og en P2-procesanalyse. P2-rapporten skal behandle matematiske/matematik-økonomiske aspekter og kontekstuelle aspekter. I vejledes i disse emner af henholdvis hovedvejleder og bivejleder.

I de følgende afsnit beskrives den matematik I kan arbejde med i projekterne. I dette projektkatalog giver vi dog kun løse beskrivelser af begreberne. Det overlades til jeres projektarbejde at give præcise matematiske definitioner.

2 Euler-kredse: ruter gennem alle kanter.

Grafteori er en matematisk disciplin, der har fået sit store gennembrud i det tyvende århundrede. Der er dog også ældre problemstillinger i grafteori. Man kan sige at grafteori blev grundlagt i 1736 af den svejtsiske matematiker Leonhard Euler, der er en af de største matematikere gennem tiderne.

Euler havde hørt om et problem fra den østprøjsiske by Königsberg (Kaliningrad): Floden Pregel løber gennem Königsberg og deler byen i to flodbreder og to øer. Disse fire områder er forbundet med syv broer. Kan man gå en tur i byen sådan at man passerer over hver af de syv broer præcis én gang. Euler indså hurtigt at det kan man ikke.

Han oversatte så problemet til et grafteoretisk problem: Kan man finde en rute (en vej) i en graf som går gennem hver kant præcis én gang. Dette problem løste han ved at give en karakterisation af de grafer, der har en sådan rute. Senere har man så opkaldt disse ruter efter Euler. Man kan tale om en Euler-kreds hvis ruten vender tilbage til start-punktet. Ellers taler man om en Euler-vej. Det er i dag et standard emne i grafteori-lærebøger at bevise Eulers karakterisation og at angive en algoritme der finder ruten.

Hvis der ikke er en Euler-kreds i grafen så står man med et mere kompliceret problem: find en korteste rute som går gennem hver kant mindst én gang. Problemet studeres som regel i vægtede grafer hvor hver kant har en længde.

Et postbud skal gå igennem alle gader i sit distrikt mindst én gang. Hvis han vil planlægge ruten så han går kortest muligt, så har han netop det ovenstående problem. Problemet kaldes derfor det kinesiske postbuds problem.

Samme problemstilling opstår for skraldebilen, sneploven eller andre, der skal igennem alle gader.

3 Graffarvning: firefarve problemet.

Et andet klassisk problem, der har givet anledning til teoretiske studier af grafer er det såkaldte firefarve problem. Firefarve problemet spørger om man kan farve landene i et vilkårligt landkort med fire forskellige farver sådan at to lande, der har en fælles grænse skal have forskellig farve. Lande der ikke grænser op til hinanden må gerne få samme farve.

Ud fra et landkort kan vi konstruere to grafer, der begge er planare. At de er planare betyder at de kan tegnes i planen uden kanter der krydser hinanden. Den ene af disse grafer har grænserne som kanter og punkterne er de steder hvor tre eller flere lande mødes. Den anden graf har et punkt for hvert land med en kant mellem to punkter hvis de tilsvarende lande har en fælles grænse.

Firefarve problemet kan nu formuleres som et grafteoretisk problem: kan punkterne i en vilkårlig planar graf tildeles farver sådan at to punkter, der er forbundet med en kant, har forskellig farve.

Dette problem har beskæftiget mange matematikere gennem det tyvende århundrede og i sidste halvdel af nittende århundrede. Deres arbejde har resulteret i mange delresultater. Men det endelige bevis for firefarve *sætningen* har gjort intensivt brug af computere til at analysere utallige tilfælde. Desuden bygger beviset på de mange tidligere delresultater. Sætningen blev første gang bevist i 1976. Et mere gennemarbejdet bevis kom i 1997. I 1976 var det helt nyt at man kunne bevise matematiske sætninger ved brug af computere. Beviset førte derfor til en debat om hvad man forstår ved et bevis: skal det f.eks. kunne læses af mennesker.

Som eksempel på en sætning, der blev bevist tidligt uden brug af computer kan nævnes at landkort kan farves med to farver hvis og kun hvis grafen, der består af grænser, har en Euler-kreds.

4 Korteste veje.

Et af de mest grundlæggende emner i kombinatorisk optimering er korteste vej problemet. Dette har en umiddelbar anvendelse i trafikken hvis man skal finde en korteste rute fra punkt A til punkt B. Men der mange andre anvendelser af korteste veje. F.eks. indgår problemet i løsningen af visse andre optimeringsproblemer, som f.eks. det kinesiske postbuds problem. I visse anvendelser vil længden af en kant altid være et positivt tal, men i andre anvendelser tillader man negative kant-længder. Der findes et stort antal algoritmer til bestemmelse af korteste veje. Hvilken algoritme, der er bedst egnet til en konkret anvendelse afhænger blandt af om der er kanter med negativ længde.

5 Lineær programmering.

Lineær programmering kan delvis opfattes som et kombinatorisk optimeringsproblem, idet vi ved at maksimum/minimum skal findes i et af de endeligt mange hjørnepunkter. Men grunden til at lineær programmering nævnes her er at en del af de optimeringsproblemer, der vedrører grafer, kan formuleres som lineære programmerings-problemer.

Det gælder f.eks. korteste vej. Betragt en orienteret graf med kanter

$e_1, \ldots, e_m$ med længder henholdsvis $\ell_1, \ldots, \ell_m$. Vi skal finde en korteste vej fra punktet A til punktet B. Vi indfører variable $x_1, \ldots, x_m$, der skal angive hvilke kanter vejen går gennem. Hvis vejen går gennem kanten e_i så sætter vi $x_i = 1$ og ellers $x_i = 0$.

Følgende vil så være opfyldt:

- For ethvert $i = 1, \ldots, m$ er $0 \leq x_i \leq 1$.
- For ethvert punkt v i grafen, undtagen A og B er
$$\sum_{i \in \mathrm{Ind}(v)} x_i = \sum_{i \in \mathrm{Ud}(v)} x_i,$$
hvor der i første sum summeres over de tal i, der opfylder at kanten e_i er orienteret ind i v og i anden sum summeres over tal i, der opfylder at e_i er orienteret ud fra v.
- Desuden er
$$\sum_{i \in \mathrm{Ind}(A)} x_i = 0, \qquad \sum_{i \in \mathrm{Ud}(A)} x_i = 1$$
og
$$\sum_{i \in \mathrm{Ind}(B)} x_i = 1, \qquad \sum_{i \in \mathrm{Ud}(B)} x_i = 0.$$

Længden af vejen er $x_1\ell_1 + \ldots + x_m\ell_m$.

Vi skal altså minimere funktionen $x_1\ell_1 + \ldots + x_m\ell_m$ under betingelsen at overstående uligheder og ligninger skal være opfyldt. Egentlig er der det yderligere krav at alle variable x_i skal have en heltallig værdig. Problemet er altså et heltals lineært programmeringsproblem. Heltals lineære programmeringsproblemer er i almindelighed vanskelige. Men i det konkrete problem kan man overbevise sig om at der en optimal løsning til vores lineære programmerings problem hvor alle x_i er hele tal – altså enten 0 eller 1.

6 Transport i netværk.

Antag vi skal transportere nogle varer fra A til B gennem et netværk (altså: en orienteret graf med vægte på kanterne). Vægten på en kant skal i dette tilfælde opfattes som en kapacitet af kanten (altså: det største antal varer, der kan kan transporteres gennem denne kant). En sådan transport kaldes en strømning. Det skal naturligvis være sådan at antal varer, der strømmer ind i et punkt v er lig med antal varer, der strømmer ud af v, undtagen hvis $v = A$ eller $v = B$. Vi skal nu bestemme det største antal varer, der kan transporteres fra A til B. Der findes flere gode algoritmer til at finde en maksimal strømning. Nogle af dem skal undervejs bestemme korteste veje.

Hvis den maksimale strømning fra A til B har værdi s (Der sendes altså s varer fra A til B.) så kan man bevise at der findes en mængde af kanter hvis samlede kapacitet er s og som opfylder at hvis man fjerner dem fra grafen så er der ikke længere nogen vej fra A til B. Dette resultat kaldes "max-flow-min-cut" sætningen.

Max-flow-min-cut sætningen er også et fundamentalt resultat i det teoretiske studium af grafers struktur: Hvis alle kanters kapacitet sættes til 1, udtaler sætningen sig om grafens "sammenhængsgrad" og sætningen kaldes i dette tilfælde Mengers sætning.

Strømning i netværk er et problem, der naturligt kan formuleres som et lineært programmerings problem. Max-flow-min-cut sætningen kan da opfattes som et korollar til dualitetssætningen i lineær programmering.

Strømning har har også anvendelse i forbindelse med optimale parringer i to-delte grafer (se afsnit 9, nedenfor) og i forbindelse med det kinesiske postbuds problem i orienterede grafer.

7 Hamilton-kredse og den handelsrejsendes problem.

En Hamilton-kreds i en graf er en kreds, der besøger hvert punkt præcis én gang – bortset fra at startpunkt=slutpunkt. Definitionen af en Hamilton-kreds minder altså lidt om definitionen af en Euler-kreds. Der er dog en væsentlig forskel når vi går dybere i teorien. Der findes et simpelt kriterium, der afgør om en graf har Euler-kreds. Men det er meget svært at afgøre om en graf har en Hamilton-kreds.

Antag vi har en vægtet ikke-orienteret graf; helst med en kant mellem hvert par af punkter så vi er sikre på at der mange Hamilton-kredse. Så vil vi gerne finde den korteste Hamilton-kreds i grafen. Dette kaldes den handelsrejsendes problem, idet vi forestiller os en handelsrejsende, der skal besøge et antal byer. Der er uden betydning i hvilken rækkefølge byerne besøges, men vi vil gerne gøre det så den samlede rejse bliver så kort som muligt. Det forudsættes at vi kender afstanden mellem alle par af byer.

Det samme problem optræder i mange andre sammenhænge. Man kan f.eks. forestille sig en industrirobot, der skal bore huller på bestemte punkter i en genstand. Den skal helst bevæge sig kortest muligt og vende tilbage til startpositionen så den er klar til et nyt emne.

Der kendes ikke nogen effektiv algoritme til løsning af den handelsrejsendes problem. Man kunne forsøge at konstruere samtlige Hamilton-kredse i grafen og beregne længden af hver af dem. Men antallet af Hamilton-kredse er alt for stort til det er en realistisk fremgangsmåde, undtagen i små grafer.

Men der findes hurtige algoritmer, der ikke finder optimal løsning, men en Hamilton-kreds der kun er lidt længere end den optimale. F.eks. findes der algoritmer, hvor man kan bevise at den konstruerede Hamiltonkreds er højst 2 eller $\frac{3}{2}$ gange så lang som den optimale. Disse algoritmer bygger på minimum vægt udspændende træer, parringer og Euler-kredse.

8 Minimum vægt udspændende træer.

Et udspændende træ i en ikke-orienteret graf består af alle grafens punkter og en delmængde af dens kanter, udvalgt sådan at der er veje mellem alle par af punkter, men er ingen kredse. Et træ altså en minimal måde at forbinde punkterne. Der kan være mange forskellige udspændende træer i en graf. Hvis grafen har n punkter og der er en kant mellem hvert par af punkter, så er der præcis n^{n-2} udspændende træer. For andre grafer kan antallet f.eks. udtrykkes som determinanten af en bestemt matrix.

Et minimum vægt udspændende træ i en vægtet graf er et udspændende træ, hvor summen af vægtene af træets kanter er minimalt. Til trods for at antallet af udspændende træer i en graf kan være meget stort, så findes der faktisk effektive algoritmer, der finder det træ som har mindst vægt.

Minimum vægt udspændende træer finder anvendelse hvor skal forbinde et antal punkter så billigt som muligt. Der er også anvendelser f.eks. ved opdeling af en mængde af objekter i grupper af objekter, der ligner hinanden.

9 Parring.

En parring i en ikke-orienteret graf er en inddeling af punkterne i par af to punkter, som er forbundet med en kant. Hvert punkt er i højst (evt. præcis)

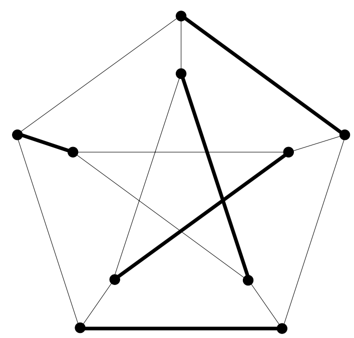

Figure 4: Petersen grafen fra figur 3 med en parring. Kanter tegnet med en tyk linie betyder at punkterne er parret. I dette eksempel er alle punkter parret.

ét af disse par.

I optimering er man tit interesseret i bestemme en parring i en vægtet graf med størst mulig samlet vægt (eller mindst mulig samlet vægt).

I mange tilfælde har grafen to typer af punkter og hver kant forbinder to punkter af forskellig type. F.eks.: visse punkter svarer til jobs, der skal udføres og de andre punkter svarer til personer. En kant mellem et job og en person angiver at personen kan udføre dette job. Kantens vægt kan være prisen for udførelsen. Man er så interesseret i få alle jobs udført (af forskellige personer) sådan at prisen bliver så lille som muligt.

Der findes særlige algoritmer, der finder en optimal parring i disse såkaldt to-delte grafer.

Man er også interesseret i at finde en optimal parring i generelle grafer. Altså grafer der ikke er to-delte. Dette har f.eks. anvendelse i forbindelse med løsning det kinesiske postbuds problem og den handelsrejsendes problem. Der findes også en algoritme, der løser problemet generelle grafer, men den er naturligvis mere kompliceret end den algoritme, der kun skal løse problem for to-delte grafer.

10 Projekterne

- Eulerkredse og det kinesiske postbuds problem
- Hamiltonkredse og den handelsrejsendes problem

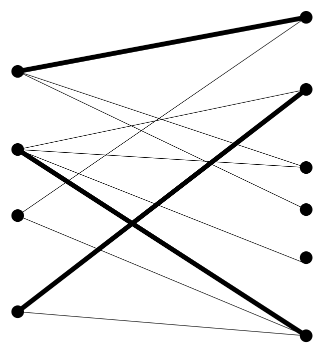

Figure 5: En todelt graf med parring. Eksempel: punkter til venstre er personer der vil donere en nyre. Punkter til højre er patienter med behov for ny nyre. En kant angiver at donerens nyre er egnet til patienten. Parringen viser hvordan man vælger at foredele nyrerne.

- Strømning i netværk
- Farvelægning: firefarve problemet
- Parring
- Udspændende træer

References

[1] Kenneth H. Rosen, Discrete Mathematics and its Applications, sixth edition, McGraw-Hill 2007.