Indhold

[Skemalægning 1](#_Toc349305177)

[Definition 1](#_Toc349305178)

[Skoleskemalægning 2](#_Toc349305179)

[Det simple skema 2](#_Toc349305180)

[De hårde skemaer 2](#_Toc349305181)

[Kompleksitet 3](#_Toc349305182)

[Løsningsmetoder 3](#_Toc349305183)

[Eksakte 3](#_Toc349305184)

[Heuristikker 3](#_Toc349305185)

[Dataformat 4](#_Toc349305186)

[XHSTT Formatet 4](#_Toc349305187)

[Times 4](#_Toc349305188)

[Resources 5](#_Toc349305189)

[Events 5](#_Toc349305190)

[Constraints 5](#_Toc349305191)

[Omkostningsfunktion 6](#_Toc349305192)

[Dataarkiv 6](#_Toc349305193)

[Tidligere undersøgelser 6](#_Toc349305194)

[Før ITC2011 6](#_Toc349305195)

[ITC2011 6](#_Toc349305196)

[Tabu search 6](#_Toc349305197)

# Skemalægning

Jeg vil i dette kapitel definere skemalægning, kigge på tidligere UNDERSØGELSER? og så modellere problemet matematisk.

## Definition

Skemalægning er planlægningen af tidspunkterne, for møder af ressourcer, under givne forudsætninger.

Ressourcer vil typisk være lokaler, undervisere, klasser og/eller studerende.

Mødet af ressourcer, herefter kaldet et event, vil i undervisningssammenhænge typisk vil være en undervisningstime, men kan også være et lærermøde eller lignende.

Forudsætningerne er opdelt i hårde og bløde betingelser. De hårde betingelser skal overholdes og de bløde skal så vidt muligt overholdes.

Den vigtigste hårde betingelse er at skemaet er fysisk muligt. Altså at ingen ressource skal være to steder på samme tid, samt at alle events skal planlægges, altså have en tid tildelt.

Visse ressourcer være forud tildelt specifikke events, f. eks. kan det være bestemt at Value Chain Design skal undervises af Ole Olesen. Det kan også være et krav at en delmængde af en ressource skal bruges til et event, f.eks. skal have øvelses hold i IT foregå i et IT-lokale og ikke hvilket som helst lokale. Sådanne specifikationer vil også være hårde betingelser.

De bløde betingelser kan også ses som præferencer, som f.eks. ’undgå så vidt muligt timer fredag formiddag’ eller ’begræns antallet af mellemtimer’.

I ovenstående har jeg brugt eksempler fra undervisningsområdet, men vagtplanlægning på en skadestue, vil også gå ind under ovenstående definition. Der vil de enkelte events have forud definerede tider og så vil problemet bestå i at tildele ressourcerne. Ressourcerne vil så være personalet og lokalet være givet.

Jeg vil i dette speciale fokusere på skoleskemaer, i det følgende vil jeg se på tidligere undersøgelser.

## Skoleskemalægning

Skoleskemalægningsproblemers sværhedsgrad følger til en hvis grad, sværhedsgraden i uddannelsesinstitutionerne.

I 0. klasse bliver eleverne i samme lokale og har kun 1 lærer. Dvs. der er faktisk ikke noget problem i skemalægningen. Så begynder eleverne at få forskellige fag, undervist af forskellige lærere, men bliver de først par år stadig i samme lokale. Dette ændrer sig med fag som hjemkundskab, fysik og idræt, der kræver specielle lokaler. På de største klassetrin indføres valgfag, hvor eleverne ikke længere nødvendigvis er fra samme klasse, men fra samme årgang. Valgfag komplicerer tingene betydeligt, da de skal grupperes sammen og klasserne ikke længere kan ses som ens enheder. Grupperingen af valgfag skal ske, så eleverne stadig kan have obligatoriske fag, med deres egen klasse. På gymnasiet er problemstillingen den samme, men med større volumen. Flere valgfag og flere klasser.

På universitetet

**Folkeskole:** I folkeskolen består problemet i at skoleklasser skal parres med skolelærere uden at der er overlap. Lokaler er ofte helt ude af problemet, da de anses som faste for klassen.

**Universitet:** På universitetet er klassetilhørsforholdet mindre og de skemaet skal lægges så de studerende har mulighed for at vælge forskellige kurser.

**Eksamener:** Eksamensdatoerne skal spredes mest muligt, så de studerende har nogenlunde lige meget forberedelsestid, til hver eksamen.

### Denne opdeling Det simple skema

Det mest basale problem vurderer om det i et givent tidsrum er muligt for et sæt lærere at undervise et sæt klasser. Mere formelt, givet:

sæt af klasser

sæt af lærere

undervisningsmatrix hvor er antallet af lektioner klasse skal undervises af lærer

Findes et acceptabelt skema med højst tidsenheder, således at alle lektioner er tildelt.

Acceptabelt defineres i det simple skema som at det fysisk kan lade sig gøre, altså at ingen lærer eller klasse skal være to steder på samme tid.

CT1

Lad såfremt klasse *i* mødes med lærer *j* mødes til tid *k* ellers 0 så kan problemet formuleres således:

|  |  |  |
| --- | --- | --- |
| *Find* |  |  |
| *s.t.* |  | (1) |
|  |  | (2) |
|  |  | (3) |
|  |  | (4) |

[DETTE FORMULERES MATEMATISK OG VISES SOM GRAFFARVNINGS PROBLEM]

## De hårde skemaer

[HER SES PÅ UDVIDELSER AF DEN SIMPLE MODEL]

## Kompleksitet

## Løsningsmetoder

### Eksakte

### Heuristikker

# Dataformat

Practice and Theory of Automated Timetabling (PATAT) er en komite, der siden 1995 holdt konferencer om skemalægning. På konferencerne mødes forskere og skemalæggere og udveksler forskningsresultater og ideer.

PATAT er blevet enige om en standard for at angive et skemalægningsproblem, for nemmere at kunne udveksle resultater og problemer. Endvidere har det muliggjort konkurrencer i skemalægning, hvor alle har mulighed for at anvende deres skemalægningsprogram på det samme datasæt. Dataformatet vil jeg gennemgå i de følgende afsnit.

Den tredje konkurrence startede i 2011 og sluttede til PATAT konferencen i Norge, august 2012. På konferencen præsenterede deltagerne artikler om deres algoritmer, som jeg vil gennemgå senere. Datasættene der blev brugt i konkurrencen er offentliggjort, sammen med de vindende løsninger, som jeg forhåbentlig kan lave bedre. Danmark blev repræsenteret af 2 Ph.d.-studerende fra DTU, der vandt bronze.

## XHSTT Formatet

XHSTT står for XML High School TimeTabling. XML er et filformat, indeholdende data i en træ-struktureret måde. High School angiver skole-niveauet, Danmark havde data med fra 3 gymnasier, som dog ikke var med i selve konkurrencen.

Nedenfor ses hovedstrukturen i XML-filen.

* Archives
  + Instances
    - Times
    - Resources
    - Events
    - Constraints
  + Solution Groups
    - Solutions

Formatet er lavet fleksibelt, så det kan bruges til forskellige landes skolers forskellige skemaproblemer. Denne fleksibilitet ses ved at betingelserne har fået sin egen node. Dermed kan nye problemstillinger medtages ved blot at indføre nye typer af betingelser, men mere om det i Constraints afsnittet. Times-og Ressource-noderne beskriver henholdsvis tids- og ressource-forholdene, mens Events-noden beskriver lektionerne og deres behov for ressourcer og tid.

### Times

Tidsforholdene er essensen af skemalægningen

* Times
  + TimeGroups
    - Week
    - Day
    - TimeGroup
  + Time
    - Name
    - Week
    - Day
    - TimeGroups
      * TimeGroup

### Resources

### Events

### Constraints

Betingelserne angives under Constraints-noden. For hver betingelse angives om det er Required eller ej, dvs. om betingelsen er hård eller blød. Endvidere angives funktionen og vægten, som afvigelser skal koste. Vægten må højst være 1000 og funktionstyperne er *Sum*, *SumSquare*, *SquareSum*, *SumStep* og *StepSum*. Funktionen skal som input have afvigelsen, som beregnes individuelt for den enkelte betingelsestype. Nedenfor listes de typer, der kan angives

Assign resource constraints

* AssignResourceConstraint Id
  + Name
  + Required
  + Weight
  + CostFunction
  + AppliesTo
    - EventGroups
      * EventGroup
    - Events
      * Event
  + Role

*Assign resource constraint* knyttes til en samling af Events, enten angivet enkeltvis under Events-noden eller vha. gruppe angivelse under EventGroups-noden. I Role-noden angives hvilken rolle ressourcen skal udfylde. I den enkelte Event kan det ses hvilken ressourcetype der kan antage den angivne rolle.

Afvigelsen beregnes ud fra den samlede tidslængde, at de tilknyttede Events ikke har en ressource tilknyttet i den angivne rolle.

Denne betingelse er ofte tilknyttet en *EventGroup*, der indeholder alle Events og under *Role* angives lokale eller lærer og anses som vigtig for løsningen og derfor hård.

Assign time constraints

Split events constraints

Distribute split events constraints

Prefer resources constraints

Prefer times constraints

Avoid split assignments constraints

Spread events constraints

Link events constraints

Order events constraints

Avoid clashes constraints

Avoid unavailable times constraints

Limit idle times constraints

Cluster busy times constraints

Limit busy times constraints

Limit workload constraints

## Omkostningsfunktion

En løsning består af en række Events, hvor det for hver enkelt Event angives starttid, tidslængde og resourcer. Løsningens omkostning består af en *infeasibility* værdi og *objective* værdi. Førstnævnte er den samlede omkostning for overtrædelser af de hårde betingelser, tilsvarende er sidstnævnte for overtrædelsen af de bløde betingelser.

## Dataarkiv

# Tidligere undersøgelser

## Før ITC2011

## ITC2011

# Tabu search

En tredje ændring