Basis

Definer P (sprog der kan defineres i polynomiel tid af en TM)
Polynomial Church-Turing thesis
Definer NP (problemer hvis løsnigner kan findes ved exhaustive search) (side 9)

Introducer NP-hårdhed (alle problemer i NP reducerer til disse) (hvorfor er det interessant?)
Introducer reduktioner (polynomiel mapping fra L1 til L2)
Introducer NPC (hvorfor er det interessant?)

Hvis L1 er NP-hårdt og L1 reducerer til L2, så er L2 også NP-hårdt
Problemet er at vi ikke har et sprog at starte med
Cook viste at SAT $\in$ NPC så vi har et sted at starte

1. P, NP and NPC

Beslutningsproblemer
Polynomiel tid
NP-hardness
Reduktioner
CIRCUIT SAT ∈ NPC
CIRCUIT SAT ≤ SAT

Beslutningsproblemer
- input er bitstrenge $x \in \{0,1\}^*$
- output JA eller NEJ $y \in \{0,1\}$
- Ikke restriktivt
  - Vi kan kun give en computer bitstrenge
  - Vi kan lave funktion $f : \{0,1\}^* \rightarrow \{0,1\}^*$

P er sprog der kan afgøres i polynomiel tid
Hvad er polynomiel tid?
- Vi kan bruge en TM som beregningsmodel grundet Church-Turing thesis
- En TM tager max $p(|x|)$ for at løse problemet for et poly. p
- Dette kan siges grundet Polynomial Church-Turing thesis

NP er beslutningsproblemer hvortil en mulig løsning kan afgøres i polynomiel tid
- Opskriv formel
- Hvordan kan vi se at $P \subseteq NP$ (ignorer y i formel)

Hvis $P \neq NP$ er vi interesserede i NPC
- NP-hard er klassen af problemer som alle problemer i NP kan reduceres til
- Der er en interessant fællesmængde mellem NP og NP-hard som hedder NPC
- NPC indeholder de sværeste problemer i NP

Hvad er en reduktion, $L1 \leq L2$?
- Polynomiel beregnelig funktion fra et sprog til et andet
- Hvis $L1 \leq L2$
  - hvis $x \in L1 \Rightarrow r(x) \in L2$
  - hvis $L2 \in P \Rightarrow L1 \in P$, grundet en algoritme til løsning af L1 er $f_{L2}(r(x))$
  - hvis $L1 \in$ NP-hard $\Rightarrow L2 \in$ NP-hard, grundet reduktioner er transitive

Hvordan finder vi det første problem i NPC som vi kan reducere fra?
- Cook's theorem → SAT ∈ NPC

2. Cook's theorem and the complexity of variants of SAT

Cook's theorem siger SAT ∈ NPC

[*Basis*] (**P, NP og NPC**)

CIRCUIT SAT ∈ NPC
CIRCUIT SAT ≤ SAT
Vis at 3SAT og NAESAT følger direkte

2SAT ∈ P
3SAT ≤ MAX2SAT

3. NP-complete graph problems

Beskriv undirected graphs

[*Basis*] (**P, NP og NPC**)

3SAT ≤ INDEPENDENT SET

Vis korollar CLIQUE er i NPC

Vis korollar NODE COVER er i NPC

3SAT ≤ HAMILTON PATH

4. NP-complete problems involving sets and numbers

[*Basis*] (**P, NP og NPC**)

Introducer BIPARTITE MATCHING ($\in$ P, da det er special case af MAX FLOW)
Krydser over til NPC ved TRIPARTITE MATCHING (ligesom 2SAT til 3SAT)
3SAT ≤ TRIPARTITE MATCHING

Vis korollar EXACT COVER BY 3-SETS er i NPC
Vis korollar SET COVERING er i NPC

EXACT COVER BY 3-SETS ≤ KNAPSACK
Pseudopolynomial algorithms and strong NP-completeness

5. Local search heuristics for TSP

Tour construction algorithms

Nearest Neighbour
Greedy
Clarke-Wright
Christofides

2-Opt og 3-Opt

Tabu search
Lin-Kernighan algorithm

Simulated annealing

6. Approximation algorithms

Motivation for approximations algoritmer (**P, NP og NPC**)

VERTEX COVER (2-approximation algorithm)

Randomization and linear programming

MAX-3-CNF
WEIGHTED VERTEX COVER

TSP

Minimum spanning tree
Dette er kortere end optimal tur
Vi traverser alle kanter 2 gange
Herefter shortcutter vi, som vi kan da trekantsuligheden gælder
Derfor er vores tur mindre end 2 * c(C*)