

# (NP-kovaa) Optimointia Deklaratiivisesti

Nuorten Tiedeakatemian Akatemiaklubi  
Jeremias Berg

26.1.2020



# Kuka minä olen?

- Tietojenkäsittelytieteen tutkijatohtori.
  - Väittely, alkukesä 2018.
    - Helsingin Yliopisto.
  - Ulkomailla 2019.
  - Takaisin HY:lle 2020.
- Harrastan kiipeilyä ja frisbegolfia (lämpimällä säällä).



# Mistä aion puhua?

- Mitä on optimointi?
  - Miksi optimointi on tärkeää?
- Mitä tarkoittaa NP-kova?
  - Mitä haasteita NP-kovien optimointiongelmiin ratkaisussa on?
- Miten ja miksi NP-kovia ongelmia ratkotaan deklaraatiivisesti?
- Minkälaiset tutkimuskysymykset ovat keskeisiä?

# Mistä aion puhua?

- Mitä on optimointi?
  - Miksi optimointi on tärkeää?
- Mitä tarkoittaa NP-kova?
  - Mitä haasteita NP-kovien optimointiongelmiä ratkaisussa on?
- Miten ja miksi NP-kovia ongelmia ratkotaan deklaratiiivisesti?
- Minkälaiset tutkimuskysymykset ovat keskeisiä?

# Mistä aion puhua?

- Mitä on optimointi?
  - Miksi optimointi on tärkeää?
- Mitä tarkoittaa NP-kova?
  - Mitä haasteita NP-kovien optimointiongelmiin ratkaisussa on?
- Miten ja miksi NP-kovia ongelmia ratkotaan deklaratiiivisesti?
- Minkälaiset tutkimuskysymykset ovat keskeisiä?

# Mistä aion puhua?

- Mitä on optimointi?
  - Miksi optimointi on tärkeää?
- Mitä tarkoittaa NP-kova?
  - Mitä haasteita NP-kovien optimointiongelmiin ratkaisussa on?
- Miten ja miksi NP-kovia ongelmia ratkotaan deklaratiiivisesti?
- Minkälaiset tutkimuskysymykset ovat keskeisiä?

# Mistä aion puhua?

- Mitä on optimointi?
  - Miksi optimointi on tärkeää?
- Mitä tarkoittaa NP-kova?
  - Mitä haasteita NP-kovien optimointiongelmiin ratkaisussa on?
- Miten ja miksi NP-kovia ongelmia ratkotaan deklaratiiivisesti?
- Minkälaiset tutkimuskysymykset ovat keskeisiä?

**Optimointiongelma**  
Löydä paras ratkaisu.

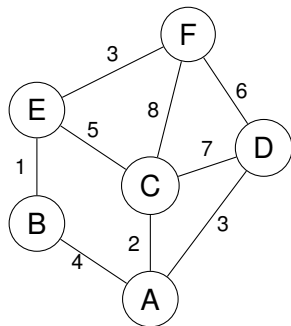


**Optimointiongelma**  
Löydä **lyhyin** ratkaisu.

**Optimointiongelma**  
Löydä **halvin** ratkaisu.

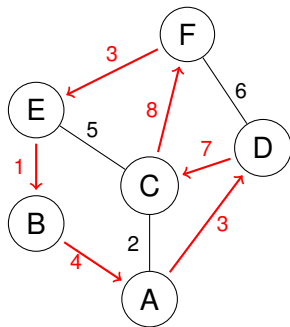
**Optimointiongelma**  
Löydä **nopein** ratkaisu.

**Optimointiongelma**  
Löydä paras ratkaisu.



**Kauppamatkustajan ongelma:**  
Löydä *lyhyin* reitti joka kiertää  
kaikki kaupungit.

**Optimointiongelma**  
Löydä paras ratkaisu.

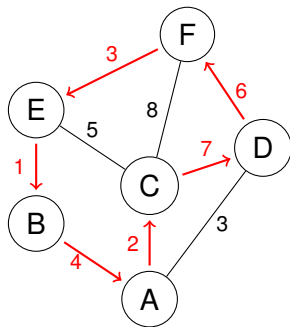


**Kauppamatkustajan ongelma:**

Löydä *lyhyin* reitti joka kiertää kaikki kaupungit.

**Pituus: 26**

**Optimointiongelma**  
Löydä paras ratkaisu.



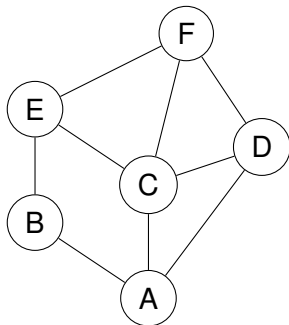
**Kauppamatkustajan ongelma:**

Löydä *lyhyin* reitti joka kiertää kaikki kaupungit.

**Pituus:** 23

## Miksi tämä on vaikeaa?

- Mahdollisten reittien määrä kasvaa nopeasti.



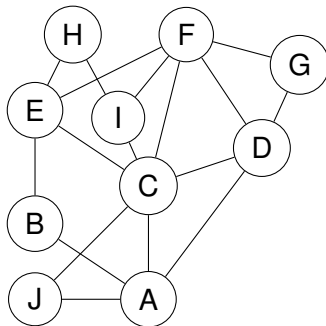
**Paikkojen määrä: 6**

**Reittien määrä: 720**

**Kaikkien kokeileminen: 0.72s**

## Miksi tämä on vaikeaa?

- Mahdollisten reittien määrä kasvaa nopeasti.



**Paikkojen määrä:** 10

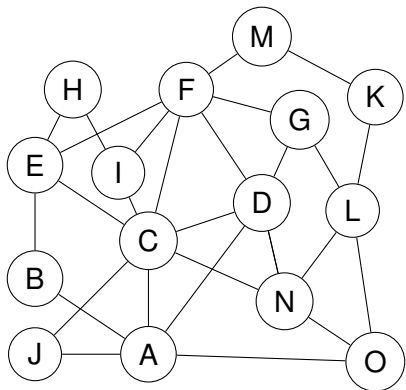
**Reittien määrä:** ~ 3.6 miljoonaa

**Kaikkien kokeileminen:** ~ 1h



## Miksi tämä on vaikeaa?

- Mahdollisten reittien määrä kasvaa nopeasti.

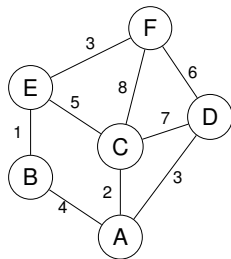


**Paikkojen määrä:** 15

**Reittien määrä:** > 1.3 biljoonaa

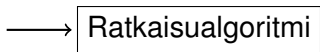
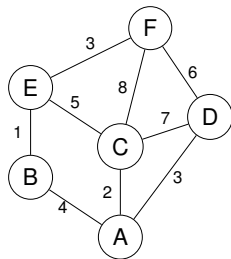
**Kaikkien kokeileminen:** ~ 10 vuotta

# Optimointiongelmien ratkaisu



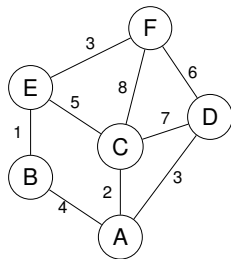
Ongelma

# Optimointiongelmien ratkaisu



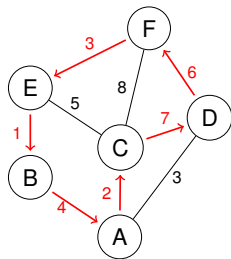
Ongelma

# Optimointiongelmien ratkaisu



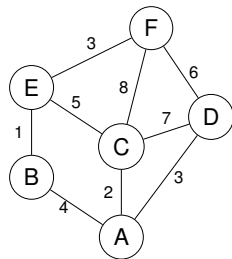
Ongelma

→ Ratkaisualgoritmi →



Ratkaisu

# Optimointiongelmien ratkaisu

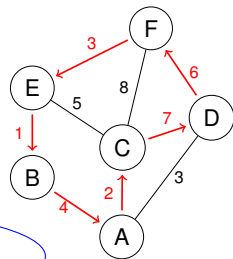


Ongelma

Sovelluksia sekä lukujärjestysten,  
että syöpähoitojen suunnittelussa

Ratkaisualgoritmi

Suurin osa tutkimuksestani  
keskittyy näihin



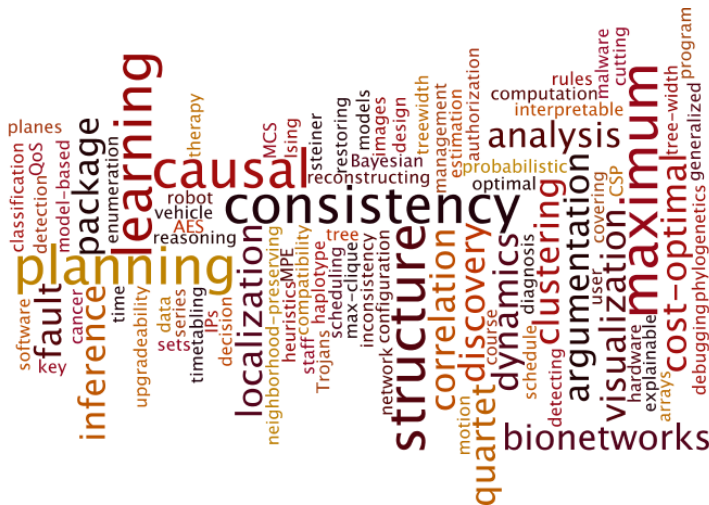
Ratkaisu

# Optimointiongelmat ovat yleisiä

- Suunnitteluongelmat
  - Itseohjaavat autot
- Aikataulutus
  - Minimoi hyppytunnit lukujärjestyksessä
- Ohjelmistonkehitys
  - Löydä pienin mahdollinen selitys bugille
- Data Analyysi
  - Klusterointi
  - Visualisointi
- Koneoppiminen
  - Todennäköisin paikka kasvoille kuvassa
  - Paras suositus katsottavaksi elokuvaksi
- ⋮

# Optimointiongelmat ovat yleisiä

Kuva: Ruben Martins



# Optimointiongelmat ovat yleisiä





## Mitä tarkoittaa NP-kovuus?

Ongelma on **NP**:ssä jos on olemassa epädeterministinen Turingin kone joka ratkaisee sen polynomisessa ajassa

Ongelma on **NP-kova** jos jokainen **NP**:ssä oleva ongelma redusoituu siihen polynomisessa ajassa.

# Mitä tarkoittaa NP-kovuus?

NP-kovien optimointi ongelmien ratkaisussa  
valitaan **laadun** ja **nopeuden** välillä

# Mitä tarkoittaa NP-kovuus?

NP-kovien optimointi ongelmien ratkaisussa valitaan **laadun** ja **nopeuden** välillä

## Nopeus:

- Niin hyvä ratkaisu kun mahdollista *rajatussa ajassa*.

## Laatu

- **Paras** mahdollinen ratkaisu *kunhan käytössä on tarpeeksi aikaa*.

# NP-kovien optimointiongelmien ratkaisutavat

**Nopeus**

**Laatu**

# NP-kovien optimointiongelmien ratkaisutavat

## Nopeus

## Laatu

### Approximaatio

- Nopeita
- Matemaattinen laatutakuu saaduille ratkaisulle.

## Nopeus

## Laatu

### Approximaatio

- Nopeita
- Matemaattinen laatutakuu saaduille ratkaisulle.

### Paikallishaku

- Vieläkin nopeampia
- Ei laatutakuuta ratkaisuille.

# NP-kovien optimointiongelmien ratkaisutavat

## Nopeus

### Approximaatio

- Nopeita
- Matemaattinen laatutakuu saaduille ratkaisulle.

### Paikallishaku

- Vieläkin nopeampia
- Ei laatutakuuta ratkaisuille.

## Laatu

### Ongelmaspecifinen ratkaisualgoritmi

- Paras mahdollinen ratkaisu (hitaammin)
- Eri algoritmi jokaiselle ongelmalle

# NP-kovien optimointiongelmien ratkaisutavat

## Nopeus

### Approximaatio

- Nopeita
- Matemaattinen laatutakuu saaduille ratkaisulle.

### Paikallishaku

- Vieläkin nopeampia
- Ei laatutakuuta ratkaisuille.

## Laatu

### Ongelmaspecifinen ratkaisualgoritmi

- Paras mahdollinen ratkaisu (hitaammin)
- Eri algoritmi jokaiselle ongelmalle

### Deklaratiivinen ratkaisutapa

- Mallinnetaan ongelma matemaattisina rajoitteina.
- Ratkaistaan rajoitejoukko.



# Deklaratiivinen tapa ratkoa optimointiongelmia

## Mitä on rajoite?

$$x \vee y \vee z$$

$$(x \vee \neg y) \wedge z$$

$$(x \wedge y) \rightarrow z$$

## Mitä on rajoite?

$$x \vee y \vee z$$

*x tai y tai z on tosi*

$$(x \vee \neg y) \wedge z$$

$$(x \wedge y) \rightarrow z$$

## Mitä on rajoite?

$$x \vee y \vee z$$

*x tai y tai z on tosi*

$$(x \vee \neg y) \wedge z$$

*joko x on tosi tai y on epätosi ja z on tosi*

$$(x \wedge y) \rightarrow z$$

## Mitä on rajoite?

$$x \vee y \vee z$$

*x tai y tai z on tosi*

$$(x \vee \neg y) \wedge z$$

*joko x on tosi tai y on epätosi ja z on tosi*

$$(x \wedge y) \rightarrow z$$

*jos x ja y ovat tosia niin z on tosi*

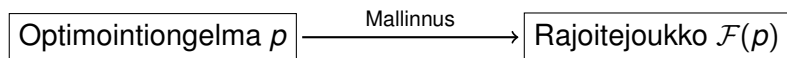
# Deklaratiivista Optimointia

Optimointiongelma  $p$

## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

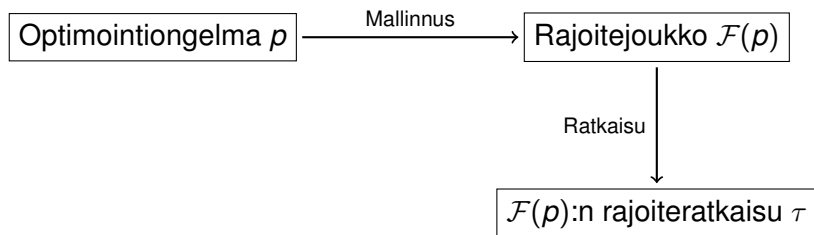
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

# Deklaratiivista Optimointia

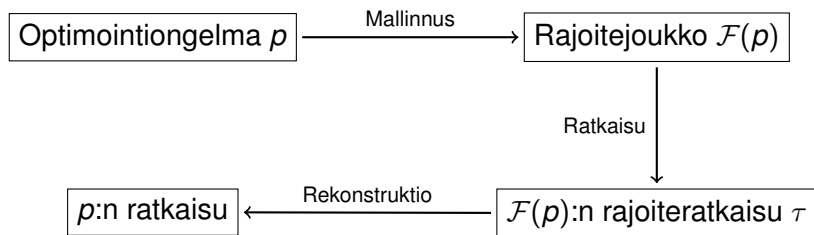


## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin



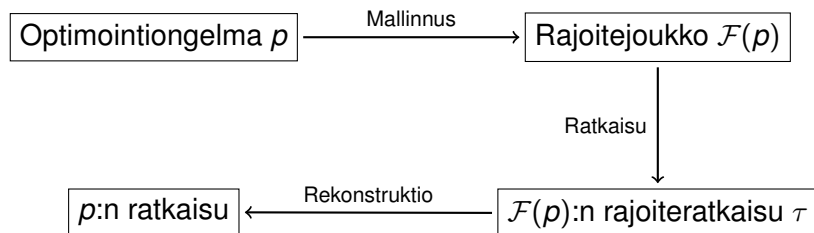
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

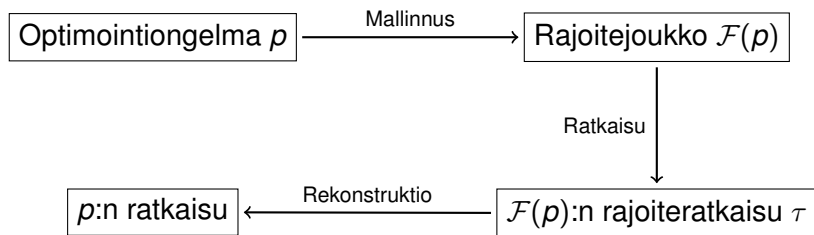
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

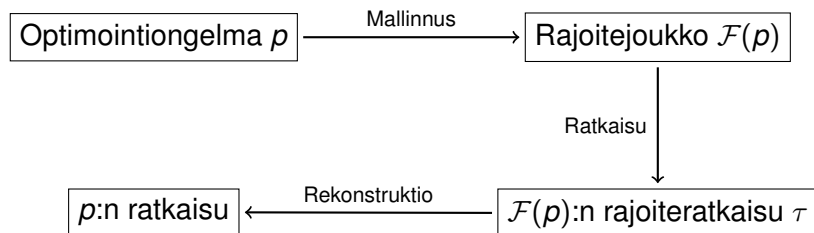
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

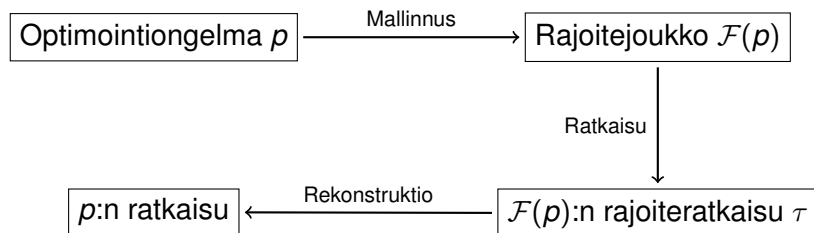
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

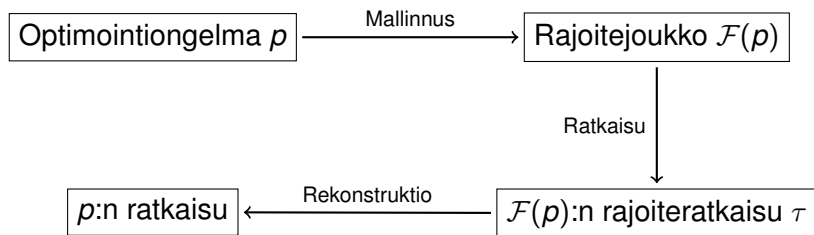
# Deklaratiivista Optimointia



## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

# Deklaratiivista Optimointia

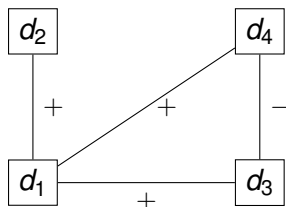


## Miksi deklaratiiivisesti?

- Yleistä, samat rajoitealgoritmit moniin eri ongelmiin
- Joustavaa, rajoitemallit ovat helposti muokattavissa
- Tehokasta, rajoitealgoritmit ovat tehokkaita
- Helppoa (ainakin helpompaa), mallin kehittäminen helpompaa kuin algoritmin

# Deklaratiivista Optimointia

## Esimerkkiongelma

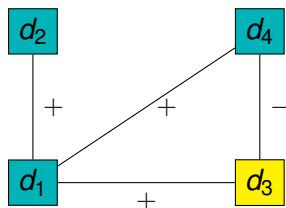


## Korrelaatioklusterointi

- Klusteroi (ryvästä) joukko dataa:
  - Samanlaiset pisteet  $\rightarrow$  samaan klusteriin
  - Erilaiset pisteet  $\rightarrow$  eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .

# Deklaratiivista Optimointia

## Esimerkkiongelma



Klusteri 1  
Klusteri 2  
Klusteri 3  
Klusteri 4

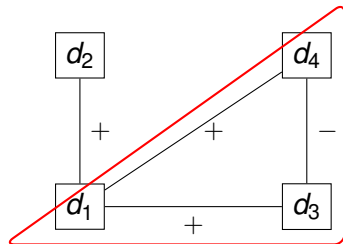
## Korrelaatioklusterointi

- Klusteri (ryvästä) joukko dataa:
  - Samanlaiset pisteet → samaan klusteriin
  - Erilaiset pisteet → eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .



# Deklaratiivista Optimointia

## Esimerkkiongelma



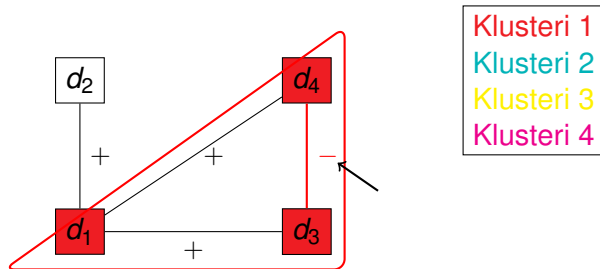
Klusteri 1  
Klusteri 2  
Klusteri 3  
Klusteri 4

## Korrelaatioklusterointi

- Klusteri (ryvästä) joukko dataa:
  - Samanlaiset pisteet → samaan klusteriin
  - Erilaiset pisteet → eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .

# Deklaratiivista Optimointia

## Esimerkkiongelma

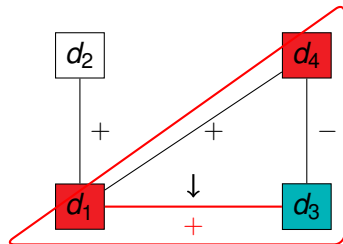


## Korrelaatioklusterointi

- Klusteri (ryvästä) joukko dataa:
  - Samanlaiset pisteet → samaan klusteriin
  - Erilaiset pisteet → eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .

# Deklaratiivista Optimointia

## Esimerkkiongelma



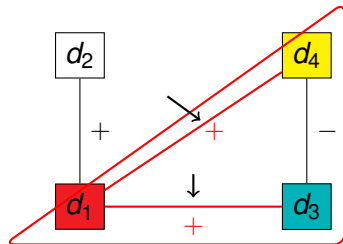
Klusteri 1  
Klusteri 2  
Klusteri 3  
Klusteri 4

## Korrelaatioklusterointi

- Klusteri (ryvästä) joukko dataa:
  - Samanlaiset pisteet → samaan klusteriin
  - Erilaiset pisteet → eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .

# Deklaratiivista Optimointia

## Esimerkkiongelma

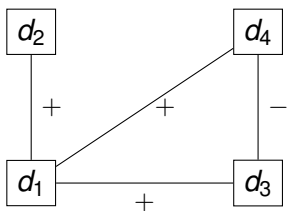


Klusteri 1  
Klusteri 2  
Klusteri 3  
Klusteri 4

## Korrelaatioklusterointi

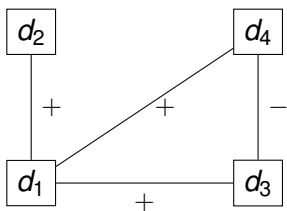
- Klusteri (ryvästä) joukko dataa:
  - Samanlaiset pisteet → samaan klusteriin
  - Erilaiset pisteet → eri klusteriin
- bioinformatiikka, informaatiotiede, data analyysi . . .

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



**Rajoitepohjainen ratkaisu:**

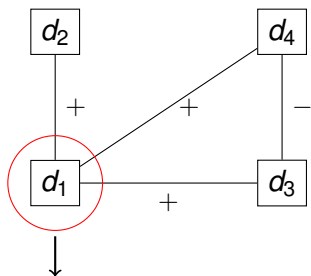
# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

1) Mallinna

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

1) Mallinna

### Muuttujat:

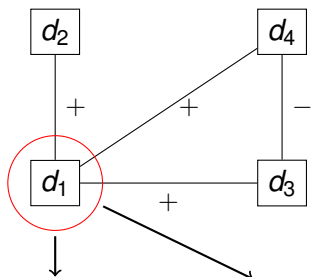
$s_1^1, \dots, s_1^4$

$s_1^1 = \text{tosi}$

jos  $d_1$  laitetaan

klusteriin 1

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

1) Mallinna

### Muuttujat:

$s_1^1, \dots, s_1^4$

$s_1^1 = \text{tosi}$

jos  $d_1$  laitetaan  
klusteriin 1

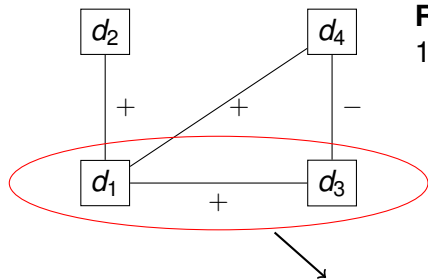
### Rajoitteet:

**Kovat:**  $d_1$  täsmälleen yhteen klusteriin.

$$(s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4) \wedge \bigwedge_{i \neq j} \neg (s_1^i \wedge s_1^j)$$



# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

1) Mallinna

### Muuttujat:

$$s_1^1, \dots, s_1^4$$

$$s_1^1 = \text{tosi}$$

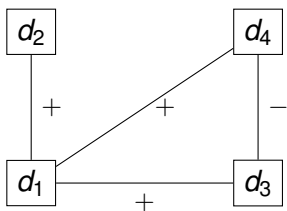
jos  $d_1$  laitetaan  
klusteriin 1

### Rajoitteet:

**Pehmeät:** suosi  $d_1$  ja  $d_3$   
samaa klusteriin.

$$\bigvee_{k=1}^4 (s_1^k \wedge s_3^k)$$

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

1) Mallinna

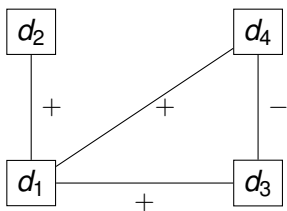
### Rajoitejoukko:

$$(s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4) \wedge \bigwedge_{i \neq j} \neg(s_i^j \wedge s_j^i) \wedge (s_2^1 \vee s_2^2 \vee s_2^3 \vee s_2^4) \wedge \bigwedge_{i \neq j} \neg(s_i^j \wedge s_j^i) \wedge \dots$$

$$\bigvee_{k=1}^4 (s_1^k \wedge s_3^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_4^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_2^k) \wedge \bigwedge_{k=1}^4 \neg(s_3^k \wedge s_4^k) \wedge \dots$$

⋮

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

- 1) Mallinna
- 2) Ratkaise

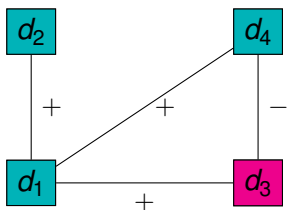
## Rajoitejoukko:

$$(s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4 \vee s_1^4) \wedge \bigwedge_{i \neq j} \neg(s_i^j \wedge s_j^i) \wedge (s_2^1 \vee s_2^2 \vee s_2^3 \vee s_2^4 \vee s_2^4) \wedge \bigwedge_{i \neq j} \neg(s_i^j \wedge s_j^i) \wedge \dots$$

$$\bigvee_{k=1}^4 (s_1^k \wedge s_3^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_4^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_2^k) \wedge \bigwedge_{k=1}^4 \neg(s_3^k \wedge s_4^k) \wedge \dots$$

⋮

# Deklaratiivinen ratkaisu korrelaatioklusterointiin



## Rajoitepohjainen ratkaisu:

- 1) Mallinna
- 2) Ratkaise
- 3) Rekonstruktio

## Rajoitejoukko:

$$(s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4 \vee s_1^4) \wedge \bigwedge_{i \neq j} \neg(s_i^i \wedge s_j^j) \wedge (s_2^1 \vee s_2^2 \vee s_2^3 \vee s_2^4 \vee s_2^4) \wedge \bigwedge_{i \neq j} \neg(s_2^i \wedge s_2^j) \wedge \dots$$

$$\bigvee_{k=1}^4 (s_1^k \wedge s_3^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_4^k) \wedge \bigvee_{k=1}^4 (s_1^k \wedge s_2^k) \wedge \bigwedge_{k=1}^4 \neg(s_3^k \wedge s_4^k) \wedge \dots$$

⋮

# Keskeiset Kysymykset

## Rajoitealgoritmien Kehitys

- Mikä rajoitekieli kannattaa valita?
- Miten saadaan rajoitealgoritmit tarpeeksi nopeiksi?
- Miten tehdään työkaluista helppokäyttöisiä?

## Rajoitealgoritmien Sovellus

- Minkälaisiin ongelmiin deklarativista lähestymistapaa kannattaa soveltaa?
- Miten eri ongelmia kannattaa mallintaa rajoitteiksi?

# Keskeiset Kysymykset

## Rajoitealgoritmien Kehitys

- Mikä rajoitekieli kannattaa valita?
- Miten saadaan rajoitealgoritmit tarpeeksi nopeiksi?
- Miten tehdään työkaluista helppokäyttöisiä?

## Rajoitealgoritmien Sovellus

- Minkälaisiin ongelmiin deklarativista lähestymistapaa kannattaa soveltaa?
- Miten eri ongelmia kannattaa mallintaa rajoitteiksi?

# Keskeiset Kysymykset

## Rajoitealgoritmien Kehitys

- Mikä rajoitekieli kannattaa valita?
- Miten saadaan rajoitealgoritmit tarpeeksi nopeiksi?
- Miten tehdään työkaluista helppokäyttöisiä?

## Rajoitealgoritmien Sovellus

- Minkälaisiin ongelmiin deklarativista lähestymistapaa kannattaa soveltaa?
- Miten eri ongelmia kannattaa mallintaa rajoitteiksi?

## Deklaratiivinen Optimointi

- Mallinna ratkaistava ongelma rajoitteina.
- Ratkaise rajoitemalli.

## Miksi deklaratiiivisesti?

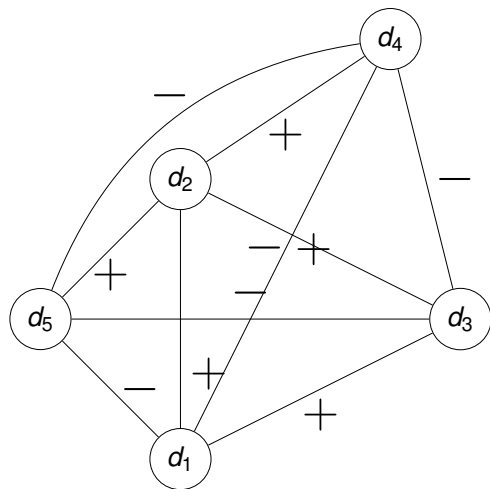
- Rajoitemallin kehittäminen helpompaa kuin ratkaisualgoritmin.
- Tehokasta.
- Helposti muokattavissa eri ongelmille.



Kiitos!

# Korrelaatioklusterointi

Kun data on harvaa

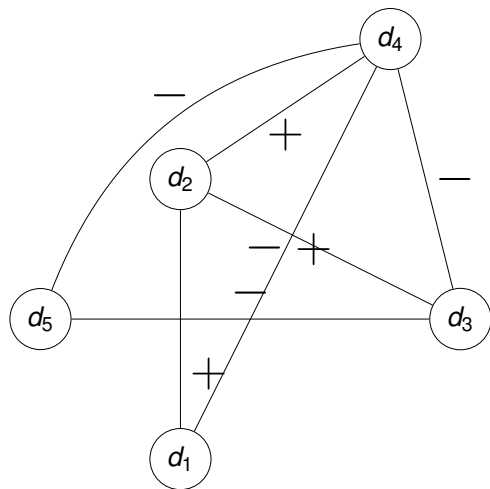


Pidä  $p * 100\%$  kaarista

$p = 1.0$

# Korrelaatioklusterointi

Kun data on harvaa

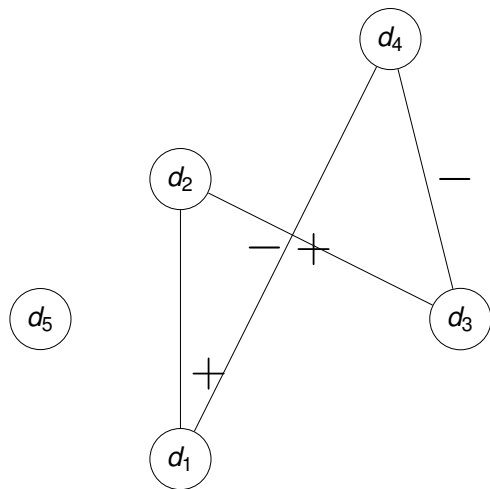


Pidä  $p * 100\%$  kaarista

$p = 0.7$

# Korrelaatioklusterointi

Kun data on harvaa

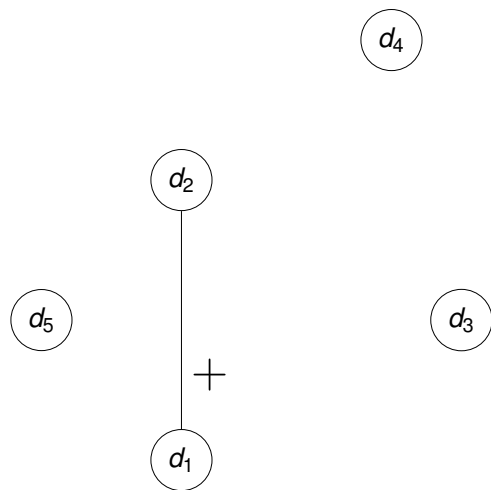


Pidä  $p * 100\%$  kaarista

$$p = 0.4$$

# Korrelaatioklusterointi

Kun data on harvaa

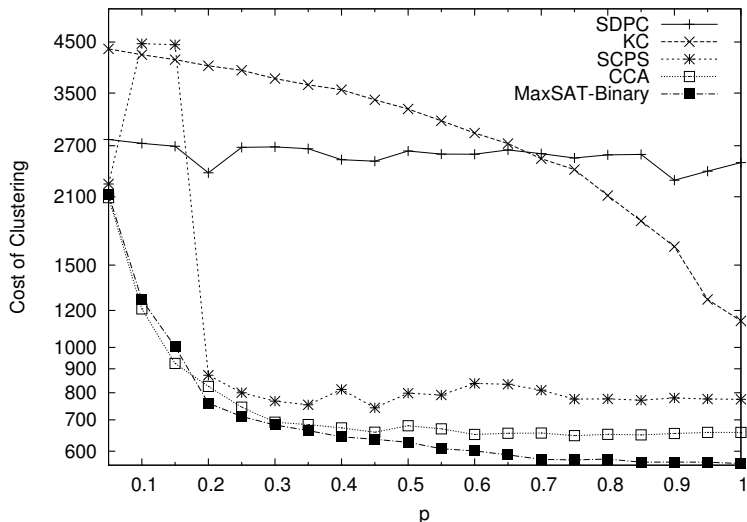


Pidä  $p * 100\%$  kaarista

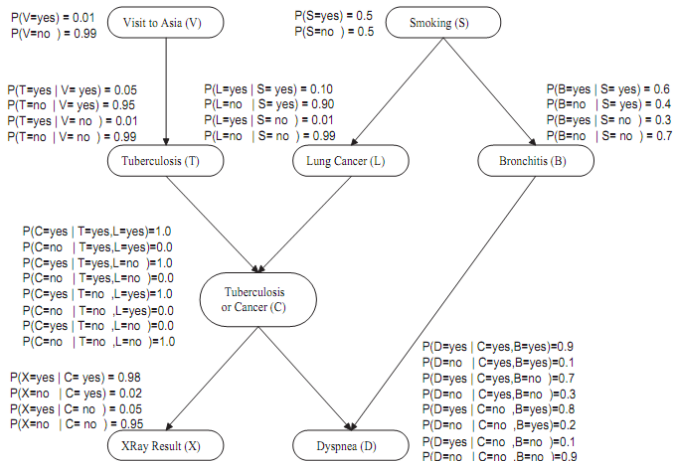
$$p = 0.1$$

# Kokeet harvalla datalla

## Tulokset



# Esimerkkiongelman 2: Bayesverkko

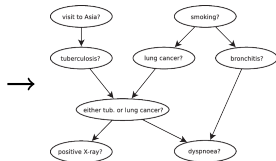


# Rajoitetun puuleveyden Bayesverkkojen rakenteen oppiminen

## Bounded Treewidth Bayesian Network Structure Learning (BTBNSL)

- Strukturioppiminen: selitä dataa verkolla.
- Käytä sitä uusien todennäköisyyksien päättelemiseen.
  - Molemmat ongelmat ovat NP-kovia.
- BTBNSL: Opi verkko jonka yli on tehokasta päätellä.

A	S	T	L	B	T&L	X	D
0	1	0	1	0	1	1	1
0	0	1	1	0	1	0	1
1	1	0	1	1	0	0	1
1	0	0	1	1	1	1	1
1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



$$P(S \mid T, L, B) = ?$$

$$P(X \mid D, A, B) = ?$$

$$P(A \mid T, L, B) = ?$$

⋮

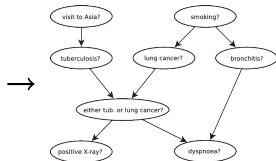


# Rajoitetun puuleveyden Bayesverkkojen rakenteen oppiminen

## Bounded Treewidth Bayesian Network Structure Learning (BTBNSL)

- Strukturioppiminen: selitä dataa verkolla.
- Käytä sitä uusien todennäköisyyksien päättelemiseen.
  - Molemmat ongelmat ovat NP-kovia.
- BTBNSL: Opi verkko jonka yli on tehokasta päätellä.

A	S	T	L	B	T&L	X	D
0	1	0	1	0	1	1	1
0	0	1	1	0	1	0	1
1	1	0	1	1	0	0	1
1	0	0	1	1	1	1	1
1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



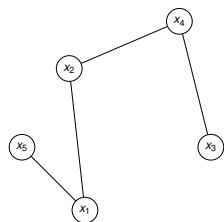
$$P(S \mid T, L, B) = ?$$

$$P(X \mid D, A, B) = ?$$

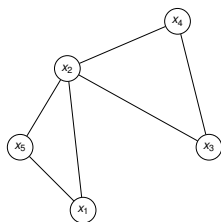
$$P(A \mid T, L, B) = ?$$

⋮

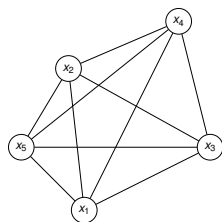
# Puuleveys



Puuleveys: 1



Puuleveys: 2



Puuleveys: 4

- Puuleveys mittaa, kuinka lähellä verkko on puuta.
- Bayesilainen päättely on tehokasta, jos verkolla on rajoitettu puuleveys.

# BTBNSL

Tuloksia

