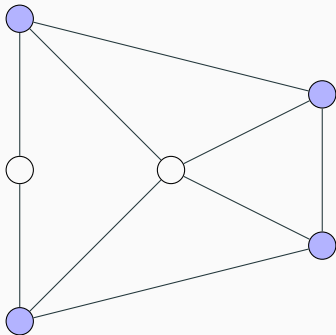




Algoritmikerho 02.05.2024

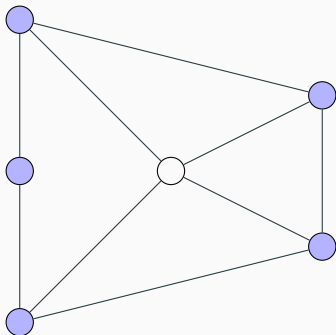
Juha Harviainen

Solmupeite Joukko solmuja, jolla ainakin toinen kunkin kaaren päistä kuuluu solmupeitteeseen

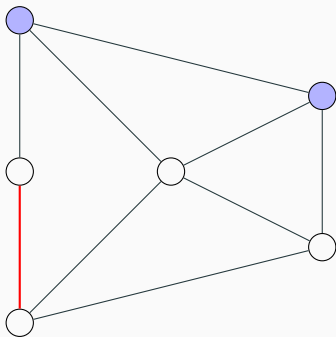


Pienet peitteet

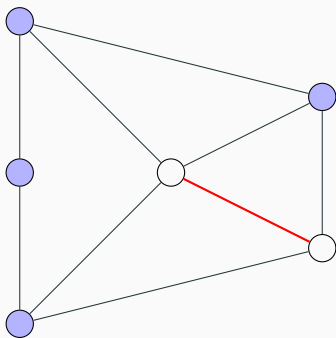
- Pienimmän peitteen löytäminen vaikeaa
- Helpompi löytää keskinertainen ratkaisu



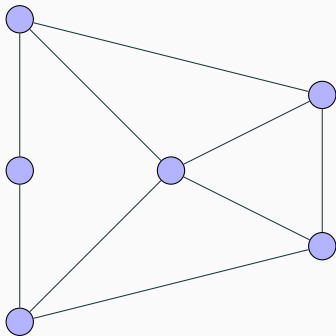
- Toisen pään jokaisesta kaaresta on oltava solmupeitteessä
- Etsitään jokin solmupeite rekursiivisesti
- Niin kauan kuin löytyy kaari, jonka kumpikaan pää ei ole ratkaisussa, lisätään molemmat päät ratkaisuun



- Toisen pään jokaisesta kaaresta on oltava solmupeitteessä
- Etsitään jokin solmupeite rekursiivisesti
- Niin kauan kuin löytyy kaari, jonka kumpikaan pää ei ole ratkaisussa, lisätään molemmat päät ratkaisuun

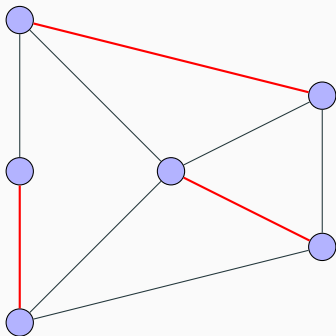


- Toisen pään jokaisesta kaaresta on oltava solmupeitteessä
- Etsitään jokin solmupeite rekursiivisesti
- Niin kauan kuin löytyy kaari, jonka kumpikaan pää ei ole ratkaisussa, lisätään molemmat päät ratkaisuun



Keskinkertaisuus?

- Toisen pään jokaisesta kaaresta on oltava solmupeitteessä
- Kaikkien algoritmin valitsemien kaarten päät erilliset
- Kaikista näistä kaarista vähintään toinen pää valittava
- Parhaan ratkaisun koko vähintään puolet löydetystä!
- Saatiin 2-approksimaatio



- Usein vaativien (NP-vaikeiden) ongelmien kohdalla tyytyväisiä, jos paras ratkaisu vakiokertojen päässä optimista
- Toisinaan kiinnostavat mielivaltaisen hyvät approksimaatiot, toisinaan voidaan näyttää niiden mahdottomuus
- Tarkistettavien tehtävien teko vaikeaa
- Lineaariohjelmointi

$$\begin{aligned} & \text{minimize} && \sum_{v \in V} c(v)x_v \\ & \text{subject to} && x_u + x_v \geq 1, \quad (u, v) \in E \\ & && x_v \geq 0, \quad v \in V \end{aligned}$$

Lineaariohjelmointi

- Pythoniin MIP-paketti: `pip install mip`
- Tavoitefunktio: maksimoi $2x + y$
- Rajoitteet: $0 \leq x \leq y$
 $x + y \leq 10$
- Ratkaisu: $x = y = 5$

```
1 from mip import *
2
3 model = Model()
4
5 x = model.add_var()
6 y = model.add_var()
7
8 model += x <= y
9 model += x + y <= 10
10 model += x >= 0
11 model += y >= 0
12
13 model.objective = maximize(2 * x + y)
14 model.optimize()
15
16 print(x.x, y.x)
```

<https://cses.fi/algo/list>