Johdatus verkkoteoriaan luento 10.4.18

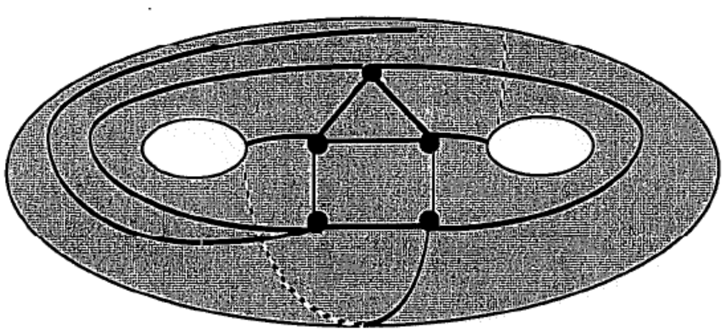
Netspace

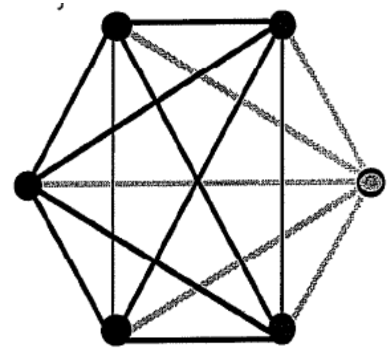
Verkko eli graafi, tasoverkko, solmut, välit, alueet , suunnatut verkot, isomorfiset verkot, verkon duaali, verkon upottaminen, verkon genus, verkon komplementti, aliverkko, täydellinen verkko, solmun ja alueen asteluku, väliverkko, säännöllinen verkko, Eulerin ja Hamiltonin verkot, kauppamatkustajan ongelma, verkon kromaattinen luku, kaksijakoinen verkko, erilaiset polut ja silmukat, verkon yhtenäisyys, puurakenteet, projektiverkko ja sen kriittinen polku …

(punaisella olleet käsitteet käsitelty)

(sinisellä olevat mainittu vielä 10.4.18)

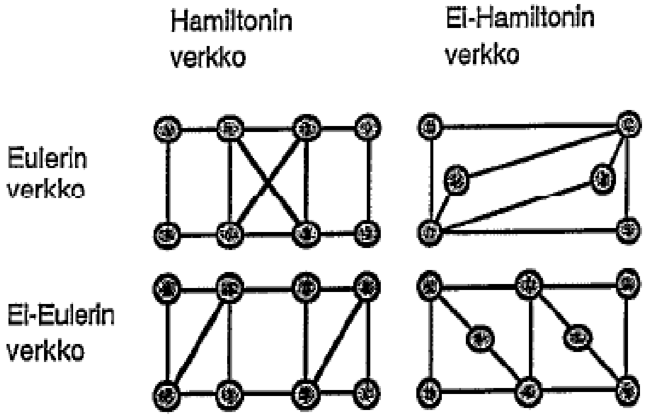
Verkon upottaminen
eli verkon sijoittaminen erilaisille topologisille pinnoille





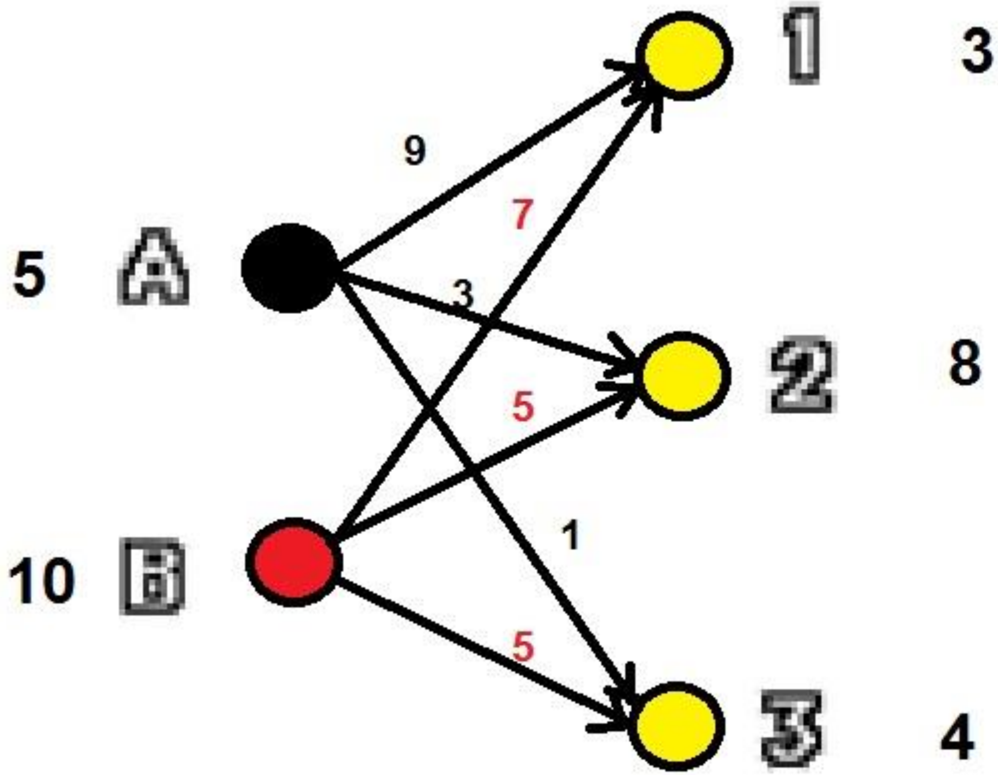
Hamiltonin polku ja silmukka

* *Hamiltonin silmukka* on suljettu polku, joka kulkee verkon kaikkien solmujen kautta siten, että kussakin solmussa käydään vain kerran. Hamiltonin silmukka on siis verkon kaikki solmut käsittävä suljettu polku eli useimmissa tapauksissa verkon kaikkien solmujen kautta kulkeva kehä. Tunnettu *kauppamatkustajan ongelma* (-> aiheesta erillinen luku) on tyypillinen esimerkki tehtävästä, jossa haetaan Hamiltonin silmukkaa. *Hamiltonin verkko* on verkko, joka sisältää Hamiltonin silmukan eli siis verkko, joka sisältää virittävän kehän (kuva 15).



Kuva 15. Hamiltonin ja Eulerin verkkoja

Kuljetusongelma


5
A
10
B
9
7
3
5
1
5
1
3
2
8
3
4

Kertaus demotehtävän pohjalta

Oheisessa kuljetusongelmassa on alimmalla rivillä kysyntä ja oikeanpuolimmaisella sarakkeella tarjonta. Muut luvut kuvaavat kuljetuskustannuksia. Hae kustannusmatriisin kantaratkaisu luoteiskulmamenetelmällä ja Vogelin menetelmällä.

Jatka tehtävä loppuun saakka kuljetusalgoritmia käyttäen.

100	100	100	49	149
498	400	251	200	351
350	299	299	400	250
199	153	199	199	750

1° Luoteiskulmamenetelmä (lähdetään “luoteiskulmasta” ja sijoitetaan ruutuun suurin mahdollinen tavaramäärä, poistetaan <u>yksi</u> rivi tai sarake):

2° Vogelin menetelmä (lasketaan kullekin riville ja sarakkeelle sakkoluku - kahden pienimmän arvon erotus - ja valitaan suurimman sakkoluvun riviltä/sarakkeelta pienimmän kustannuksen ruutu, sijoitetaan ruutuun suurin mahdollinen tavaramäärä):

Luoteiskulmamenetelmällä saatiin kantaratkaisu, jonka ulkopuolelle jäivät ruudut A3 ja B1. Kuljetusalgoritmissa lasketaan sakot näille kantaratkaisun ulkopuolelle jääneille ruuduille, joita esimerkkitapauksessa on vain kaksi. Sakon arvo saadaan kiertämällä ruutua vastaava polku (ks. sivun alin kuva) kannassa ja vuorotellen summaamalla ja vähentämällä kustannukset. Polku kierretään liikkumalla vuoroin pysty- ja vaakasuoraan, kiertosuunta ei vaikuta tulokseen. Saadaan: $S(A3) = A3 - B3 + B2 - A2 = 1 - 5 + 5 - 3 = -2$ ja $S(B1) = B1 - B2 + A2 - A1 = 7 - 5 + 3 - 9 = -4$. Algoritmi pysähtyy jos yksikään lasketuista sakoista ei ole negatiivinen. Mikäli negatiivisia sakkolukuja löytyy niin valitaan **kantaan itseisarvoltaan suurimman arvon omaava ruutu**; tässä tapauksessa siis B1. Kuormitus uudelle kantaruudulle saadaan kiertämällä sama polku jolta sakko laskettiin ja siirtämällä polulta kyseiseen kantaruutuun suurin mahdollinen kuorma. Esimerkkitapauksessa se on $\min\{3,6\} = 3$. Kun polku kierretään ympäri niin siirretään kuormia ruudusta toiseen siten että lopputilanteessa on tasapaino jälleen saavutettu eli rivi ja sarakesummat täsmäävät. Siis seuraavasti: B1 ↵ +3 ⇒ 3, A1 ↵ -3 ⇒ 0, A2 ↵ +3 ⇒ 5, B2 ↵ -3 ⇒ 3. Se ruutu, jonka kuormitus nollautuu (A1) poistuu kannasta. Jos useamman ruudun kuorma tulee nollaksi niin kannasta poistetaan ruutu, jolla on korkeimmat kustannukset.

”**kantaan itseisarvoltaan suurimman arvon omaava ruutu” = ruutu, jossa on suurin negatiivinen luku**

Sijoitusongelma
pikkukertaus

SIJOITUSONGELMAssa on kaksi yhtä suurta joukkoa ja kummankin joukon jokaiselle alkiolle on haettava vastinalkio toisesta joukosta. On esimerkiksi jaettava k kpl tehtäviä k:n henkilön kesken s.e. kullekin tulee täsmälleen yksi (sopivin) tehtävä. Sijoitusongelma on *kuljetusongelman erikoistapaus*. Siis sellainen kuljetusongelma, jossa tuottajia ja hankkijoita on yhtä paljon ja jossa kukin tuottaja vie optimiratkaisussa tavaraa ainoastaan täsmälleen yhdelle hankkijalle, joten optimiratkaisussa on alkioita n kpl (kuljetusongelmassahan määrä oli 2n-1 kun m=n). Mikäli tuottajien ja hankkijoiden määrä ei ole sama niin lisätään varjomuuttujia (dummy person), joille kustannuskerroin (cij) on nolla. Sijoitusongelma voidaan esittää LP-ongelmana alla olevassa muodossa (vertaa vastaavaan kuljetusongelman esitykseen):

$$\text{Min } z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{kun } i = 1 \ldots n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{kun } j = 1 \ldots n$$

$$x_{ij} = 0 \text{ tai } 1$$

Sijoitusongelma voitaisiin kuljetusongelman erikoistapauksena myös ratkaista sellaisena, mikä kuitenkin on turhan työlästä. On syytä muistaa, että tuolloin saadaan 2n-1 ratkaisua, joista n valitaan (loput n-1 alkiota ovat aina nollia). Sijoitusongelman ratkaisemiseksi on kehitetty ns. unkarilainen menetelmä, joka sisältää seuraavat vaiheet: 1° ongelmasta muodostetun nxn-matriisin kultakin riviltä vähennetään sen riviminimi, 2° sen jälkeen kultakin sarakkeelta vähennetään sarakeminimi, 3° yliviivataan kaikki 0-alkiot siten että viivoja on mahdollisimman vähän (aluksi eniten nollia sisältävät rivit/sarakkeet) 4° jos viivojen lukumäärä on suurempi tai yhtä suuri kuin n niin mennään viimeiseen askeleeseen (6°), 5° etsitään pienin alkio (merk. L), jota ei ole peitetty viivalla ja *vähennetään se kaikista <u>peittämättömien rivien</u> alkioista* ja *lisätään kaikkien <u>peitettyjen sarakkeiden</u> alkioihin* ja palataan askeleeseen 3°, 6° valitaan ratkaisuun tulevat nolla-alkiot siten että kultakin riviltä ja sarakkeelta tulee täsmälleen yksi alkio (huom! ratkaisu ei ole aina yksiselitteinen).

Sijoitusongelman ratkaisu
Unkarilaisella menetelmällä:

1. Ongelmasta muodostetun nxn-matriisin kultakin riviltä vähennetään sen riviminimi,
2. sen jälkeen kultakin sarakkeelta vähennetään sarakeminimi,
3. yliviivataan kaikki 0-alkiot siten että viivoja on mahdollisimman vähän, aluksi eniten nollia sisältävät rivit/sarakkeet,
4. jos viivojen lukumäärä on suurempi tai yhtäsuuri kuin n niin mennään viimeiseen askeleeseen 6.,
5. etsitään pienin alkio (merk. L), jota ei ole peitetty viivalla ja vähennetään se kaikista peittämättömien rivien alkioista ja lisätään kaikkien peitettyjen sarakkeiden alkioihin ja palataan askeleeseen 3.,
6. valitaan ratkaisuun tulevat nolla-alkiot siten että kultakin riviltä ja sarakkeelta tulee täsmälleen yksi alkio.

Huom! Ratkaisu ei ole aina yksiselitteinen.

	Vi	Vo	T	U	A
Gina	3	5	2	2	7
Nina	3	2	1	2	3
Tina	4	9	2	5	3
Xena	5	6	4	5	4

	Vi	Vo	T	U	A	
Gina	3	5	2	2	7	2
Nina	3	2	1	2	3	1
Tina	4	9	2	5	3	2
Xena	5	6	4	5	4	4
x	M	M	M	M	M	M

1	3	0	0	7
2	1	0	1	2
2	7	0	3	1
1	2	0	1	0
0	0	0	0	0

L = 1

0	2	0	0	6
1	0	0	1	1
1	6	0	3	0
1	2	1	2	0
0	0	1	1	0

2.

3.

0	2	0	0	6
1	0	0	1	1
1	6	0	3	0
1	2	1	2	0
0	0	1	1	0

5.

1.

4.

0	2	0	0	6
1	0	0	1	1
1	6	0	3	0
1	2	1	2	0
0	0	1	1	0

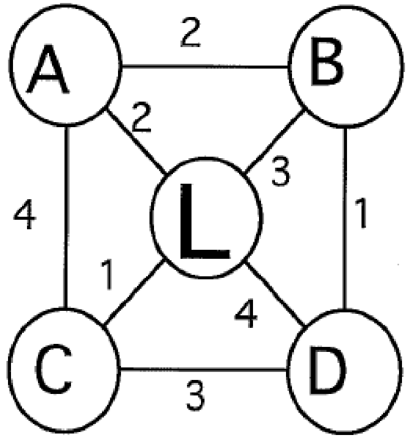
Kauppamatkustajan ongelma

Sijoitusongelma oli kuljetusongelman rajatapaus (joka oli tavallisen LP-ongelman rajatapaus) niin kauppamatkustajan ongelmaa voidaan puolestaan pitää sijoitteluongelman erikoismuotona. Tehtävänä on yksinkertaisesti hakea lyhin reitti joka kiertää verkon kaikkien solmujen kautta palaten lopuksi takaisin lähtösolmuun. Kyseessä voi olla esimerkiksi joukko firmoja, joista jokaisessa kauppamatkustajan on käytävä kertaalleen.

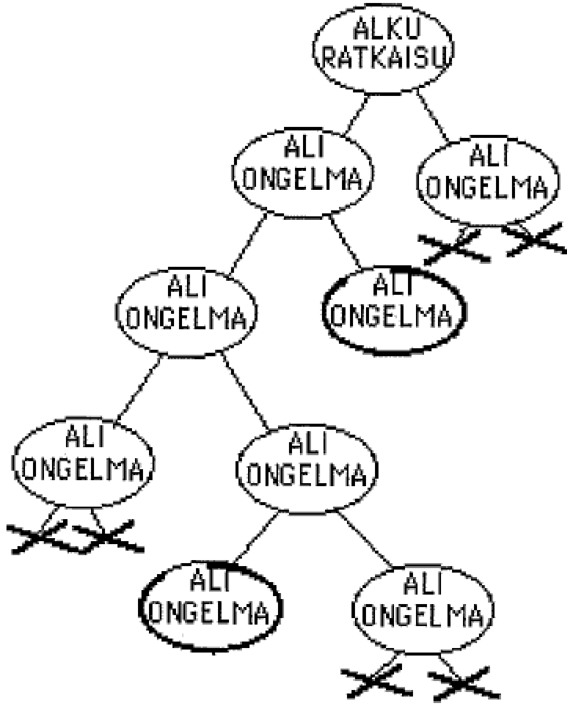
Kauppamatkustajan ongelma on eräs matematiikan perinteellisiä “suuria ongelmia”, jonka ratkaisemiseksi on kehitetty lukuisa määrä erilaisia algoritmeja. Ja säännöllisin väliajoin joku aina “keksii” sen parhaan menetelmän. Tyypillisesti mm. tätä ongelmaa ratkaistaan *branch-and-bound* menetelmää käyttäen. Branch-and-bound (“haaroitu ja rajoita”) soveltuu monien muidenkin kokonaislukuoptimointitehtävien ratkaisuun (esim. *knapsack problem*) Ajatuksena on ajatella tehtävää ensin sijoitusongelmana, jolle haetaan ratkaisu (esim. unkarilaista menetelmää käyttäen).

Ratkaisu ei kuitenkaan ole yleensä käypä (esimerkiksi koostuu useasta irrallisesta reitistä), jolloin jaetaan ongelma alitehtäviin poistamalla saadusta ratkaisusta kussakin alitehtävässä yksi alkio. Kukin alitehtävistä ratkaistaan jälleen sijoitusongelmana kunnes jossain haarassa saadaan käypä ratkaisu, minkä jälkeen muita haaroja jatketaan niin kauan kuin on mahdollista saavuttaa vielä parempi tulos.

Käsin tehtäviä ratkaisuja varten on kehitetty menetelmän sovellus, joka eliminoi ratkaisuun johtamattomat sivuhaarat jo menetelmän kuluessa. On huomattava, että lopullinen ratkaisu on aina arvoltaan suurempi tai yhtäsuuri kuin ensimmäinen sijoitusmenetelmällä haettu ratkaisu. Kauppamatkustajan ongelma LP-ongelmaksi muotoiltuna on muutoin aivan samanlainen kuin edellä esitetty sijoitusongelma, mutta lisäksi sen on sisällettävä ns. *rengasehto* eli alkioista on muodostuttava verkossa yksi yhtenäinen rengas.



	L	A	B	C	D	Min
L	M	2	3	1	4	1
A	2	M	2	4	M	2
B	3	2	M	M	1	1
C	1	4	M	M	3	1
D	4	M	1	3	M	1



Vähennetään riviminimit ja sarakeminimit. Lasketaan kullekin riville ja sarakkeelle sakko, joka on toiseksi pienimmän alkion (pienin on aina 0) arvo.

	L	A	B	C	D
L	M	1	2	0	3
A	0	M	0	2	M
B	2	1	M	M	0
C	0	3	M	M	2
D	3	M	0	2	M
Min:	0	1	0	0	0

M

	L	A	B	C	D	Sakko
L	M	0^{0}	2	0^{2}	3	0
A	0^{0}	M	0^{0}	2	M	0
B	2	0^{0}	M	M	0^{2}	0
C	$\boxed{0^{2}}$	2	M	M	2	2
D	3	M	0^{2}	2	M	2
Sakko:	0	0	0	2	2	$X_{CL} = 1$

Kullekin 0-alkiolle annetaan sakko (yläindeksi), joka on Max {rivisakko, sarakesakko}. Suurimman sakon saanut alkio otetaan kantaan (tässä xCL) ja samalla poistetaan matriisista vastaava rivi ja sarake. Käänteisalkio (xLC) eliminoidaan antamalla sille arvoksi M. Näin estetään mahdollisuus palata samaa reittiä takaisin. Samalla tavalla estetään myös pidempien silmukoiden muodostuminen, siis antamalla M sellaisten välien arvoiksi, joita pitkin voitaisiin palata syntyvälle reitille takaisin "liian aikaisin".

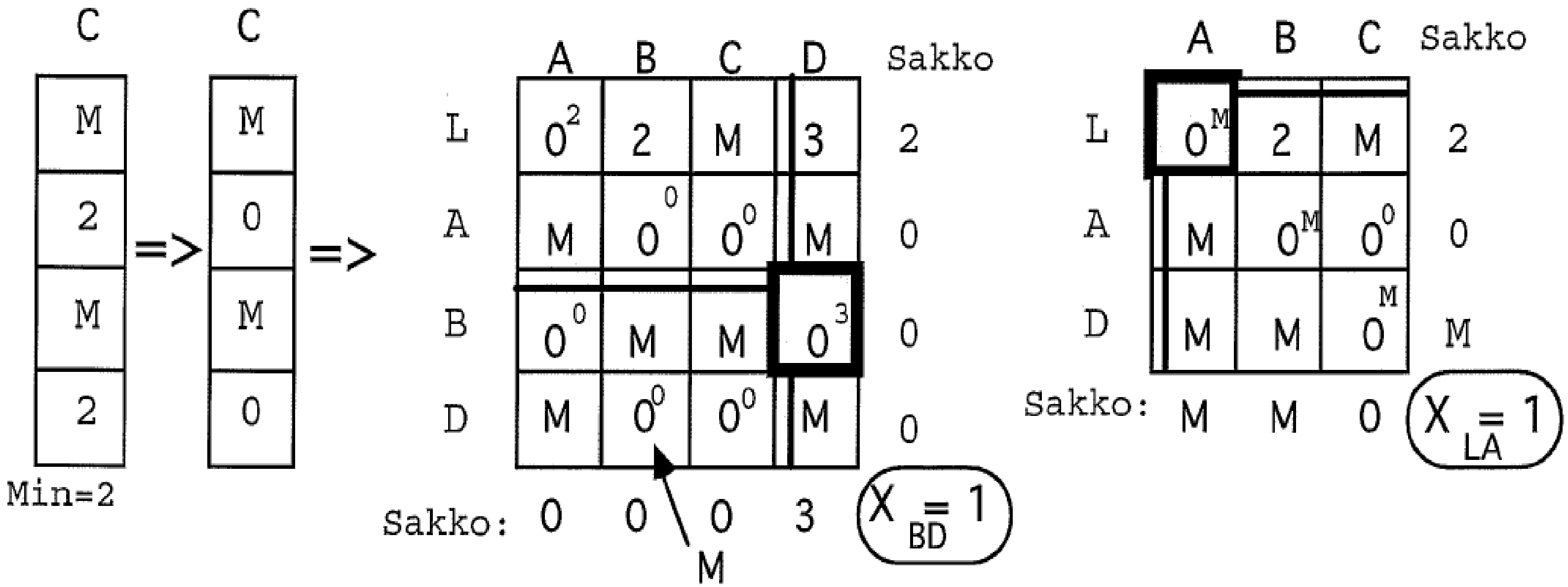
	L	A	B	C	D
L	M	1	2	0	3
A	0	M	0	2	M
B	2	1	M	M	0
C	0	3	M	M	2
D	3	M	0	2	M
Min:	0	1	0	0	0

M

	L	A	B	C	D	Sakko
L	M	0^0	2	0^2	3	0
A	0^0	M	0^0	2	M	0
B	2	0^0	M	M	0^2	0
C	$\boxed{0^2}$	2	M	M	2	2
D	3	M	0^2	2	M	2
Sakko:	0	0	0	2	2	

$X_{CL} = 1$

Sarakkeelle C ei jäänyt yhtään nollaa, joten vähennetään sarakeminimi (2) ja jatketaan pienentyneessä matriisissa kuten edellä. Näin jatketaan lisää välejä kantaan ottaen kunnes matriisi on supistunut ollen kokoa 2x2.

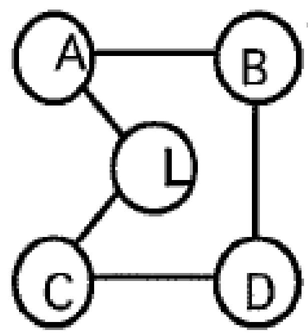


Näin jatketaan lisää välejä kantaan ottaen kunnes matriisi on supistunut ollen kokoa 2x2. Otetaan kantaan viimeiset kaksi väliä ja saadaan kauppamatkustajalle reitti, joka on esitetty kuvassa alla. Reitin pituus on 9 ja se voidaan luonnollisesti kiertää kumpaankin suuntaan.

	B	C	Sakko
A	0^{M}	0^{0}	0
D	M	0^{M}	M
Sakko:	M	0	

$X_{AB} = 1$

$X_{DC} = 1$



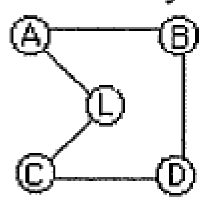
Jos samaa tehtävää lähdetään ratkaisemaan sijoitusongelmana niin saadaan neljä kantaratkaisua, joista kaksi on matriisin symmetrisyyden vuoksi samoja. Kummankin jäljelle jäävän ratkaisun arvo on 9, mutta toinen ei ole käypä ratkaisu koska reitti ei ole yhtenäinen (oikeanpuoleinen kuva alla). Branch-and-bound menetelmässä ratkaistaisiin seuraavaksi oikeanpuoleista verkkoa vastaava matriisi kolmena eri sijoitusongelmana, joista kussakin on yksi kolmion ACL väleistä eliminoitu antamalla sille matriisissa arvo M. Näin saaduista ratkaisuista jatkuu haarautuminen edelleen alitehtäviin kunnes on saatu rengasehdon täyttävä verkko tai kun ratkaisun (ei välttämättä käypä) arvo on suurempi kuin parhaan jo saadun käyvän ratkaisun. Tässä nimenomaisessa tapauksessa prosessi pysähtyy heti alussa sillä vasemmanpuoleinen verkko on ratkaisuna käypä ja sen arvo on jo vähintään yhtä hyvä kuin mikä menetelmää jatkamalla voitaisiin saavuttaa.

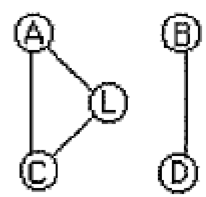
	L	A	B	C	D
L	M	0	2	0	3
A	0	M	0	2	M
B	2	0	M	M	0
C	0	2	M	M	2
D	3	M	0	2	M

	L	A	B	C	D
L	M	0	2	0	3
A	0	M	0	2	M
B	2	0	M	M	0
C	0	2	M	M	2
D	3	M	0	2	M

	L	A	B	C	D
L	M	0	3	0	3
A	0	M	0	0	M
B	3	0	M	M	0
C	0	0	M	M	0
D	3	M	0	0	M

Viivoja on 4 < 5 => väh/lis L=2





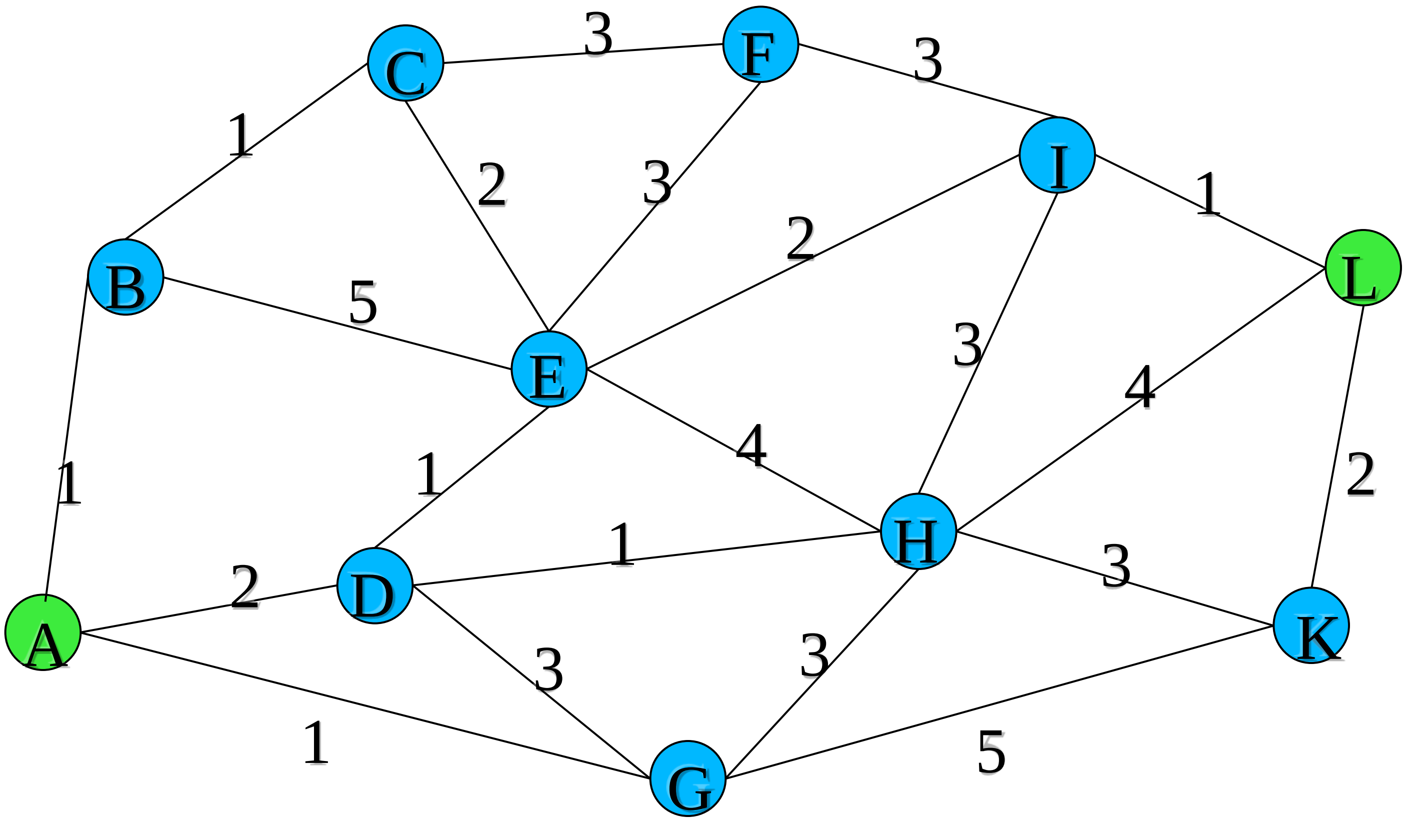
Lyhimmän polun ongelma

Kauppamatkustajan ongelmassa oli tavoitteena löytää verkossa lyhin sellainen polku, joka käy verkon jokaisessa solmussa ainakin kerran ja palaa sitten lähtösolmuun. Tällainen suljettu polku muodostaa siis verkossa silmukan.

Lyhimmän polun ongelmassa ei palata takaisin lähtöpisteeseen vaan tavoitteena on löytää lyhin polku kahden eri solmun välillä. Toinen ero kauppamatkustajan ongelmaan on se että verkon kaikissa solmuissa ei ole tarpeen käydä.

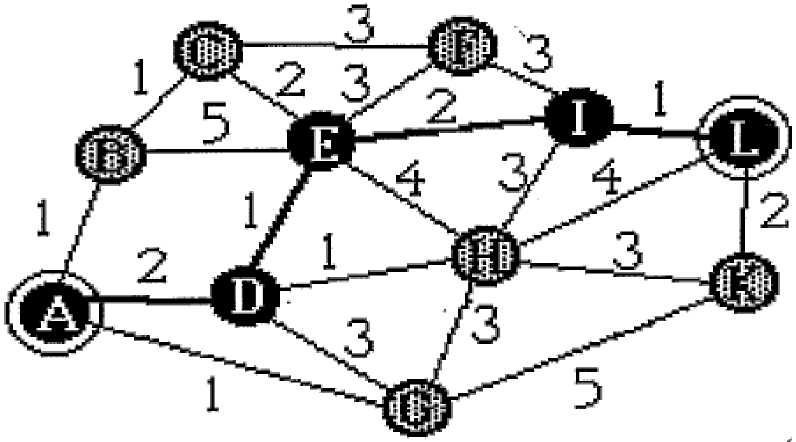
Eräs paljon käytetty tapa hakea verkosta lyhin polku on käyttää **Dijkstran algoritmia**. Ratkaistaan lyhin polku seuraavasta verkosta Dijkstran algoritmin avulla.

Verkko



Aloitetaan solmusta A merkitsemällä sille pysyvä tunnus (alleviivattu luku), joka on 0. Etäisyysmatriisista katsotaan matka naapurisolmuihin.

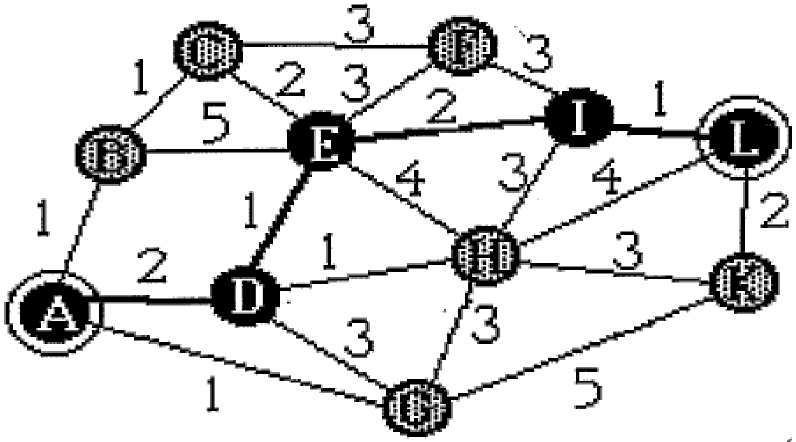
A	B	C	D	E	F	G	H	I	K	L
<u>0</u>	1	∞	2	∞	∞	1	∞	∞	∞	∞



	A	B	C	D	E	F	G	H	I	K	L
A	∞	1	∞	2	∞	∞	1	∞	∞	∞	∞
B	1	∞	1	∞	5	∞	∞	∞	∞	∞	∞
C	∞	1	∞	∞	2	3	∞	∞	∞	∞	∞
D	2	∞	∞	∞	1	∞	3	1	∞	∞	∞
E	∞	5	2	1	∞	3	∞	4	2	∞	∞
F	∞	∞	3	∞	3	∞	∞	∞	3	∞	∞
G	1	∞	∞	3	∞	∞	∞	3	∞	5	∞
H	∞	∞	∞	1	4	∞	3	∞	3	3	4
I	∞	∞	∞	∞	2	3	∞	3	∞	∞	1
K	∞	∞	∞	∞	∞	∞	5	3	∞	∞	2
L	∞	∞	∞	∞	∞	∞	∞	4	1	2	∞

Vaihtuvista tunnuksista valitaan pienin pysyväksi ja katsotaan jälleen matka etäisyysmatriisista. Jos summa pysyvän tunnuksen arvo ja matka on pienempi kuin vastaavalla solmulla aiemmin ollut vaihtuva tunnus niin summa sijoitetaan vaihtuvaksi tunnukseksi.

A	B	C	D	E	F	G	H	I	K	L
<u>0</u>	1	∞	2	∞	∞	1	∞	∞	∞	∞
	<u>1</u>	2	2	6	∞	1	∞	∞	∞	∞



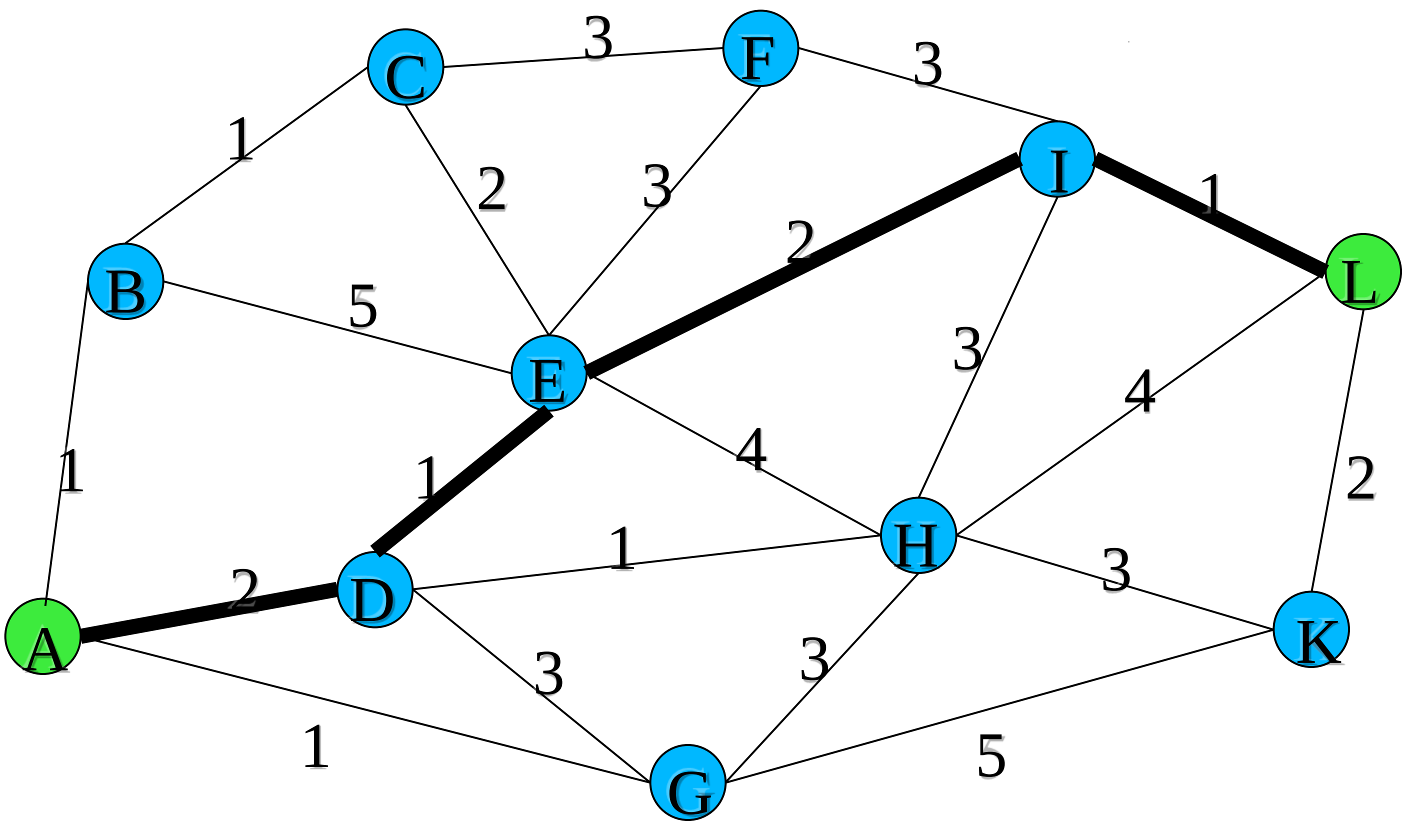
	A	B	C	D	E	F	G	H	I	K	L
A	∞	1	∞	2	∞	∞	1	∞	∞	∞	∞
B	1	∞	1	∞	5	∞	∞	∞	∞	∞	∞
C	∞	1	∞	∞	2	3	∞	∞	∞	∞	∞
D	2	∞	∞	∞	1	∞	3	1	∞	∞	∞
E	∞	5	2	1	∞	3	∞	4	2	∞	∞
F	∞	∞	3	∞	3	∞	∞	∞	3	∞	∞
G	1	∞	∞	3	∞	∞	∞	3	∞	5	∞
H	∞	∞	∞	1	4	∞	3	∞	3	3	4
I	∞	∞	∞	∞	2	3	∞	3	∞	∞	1
K	∞	∞	∞	∞	∞	∞	5	3	∞	∞	2
L	∞	∞	∞	∞	∞	∞	∞	4	1	2	∞

Valitaan jälleen pienin pysyväksi tunnukseksi ja jatketaan näin loppuun saakka. Täten saadaan lyhimmän polun pituus. Mitä kautta lyhin polku kulkee selviää takaperin laskemalla. Verrataan ensin loppusolmun etäisyyksiä vastaavien pysyvien tunnuksien erotuksiin. Jos ne ovat samat niin solmu kuuluu lyhimpään polkuun. **Esimerkiksi loppusolmun L etäisyydet naapurisolmujen joukkoon {H,I,K} ovat {4,1,2} ja vastaavat pysyvien tunnusten erotukset ovat 6-{3,5,6}= {3,1,0}.** Vertaamalla joukkoja nähdään että erotus = etäisyys solmun I kohdalla (=1). Tästä seuraa että solmu I kuuluu lyhimpään polkuun. Seuraavaksi tarkastellaan aivan samoin I:n naapurisolmuja ja näin jatketaan solmu solmulta takaisin alkuun. On syytä muistaa että lyhyimpiä polkuja voi olla myös useampia.

A	B	C	D	E	F	G	H	I	K	L
0	1	∞	2	∞	∞	1	∞	∞	∞	∞
	1	2	2	6	∞	1	∞	∞	∞	∞
		2	2	6	∞	**1**	4	∞	6	∞
		2	**2**	3	∞		3	∞	6	∞
		2		3	5		3	∞	6	∞
				3	5		**3**	6	6	7
				3	5			5	6	7
					5			5	6	7
								5	6	6
									6	6
										6

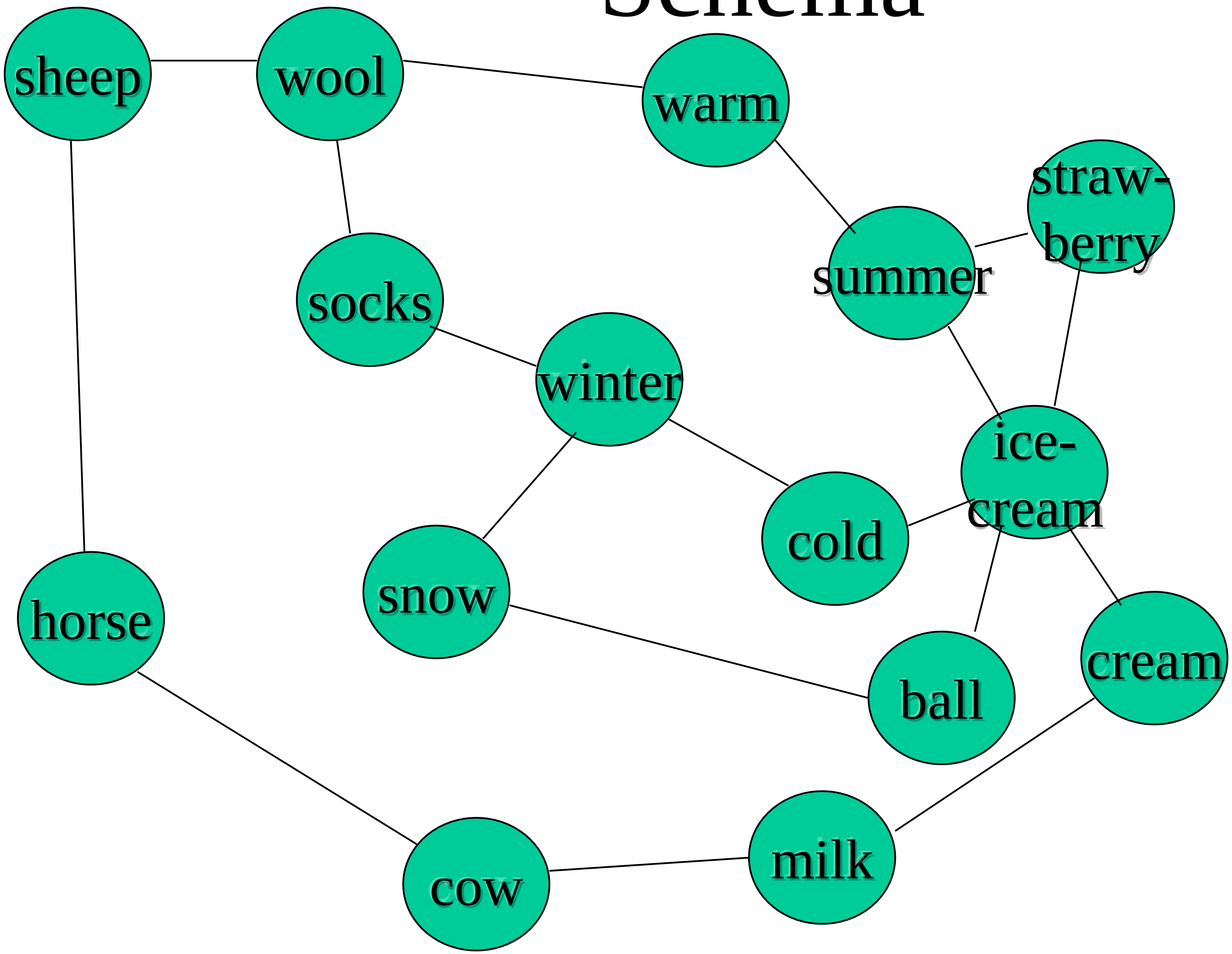
ratkaisu

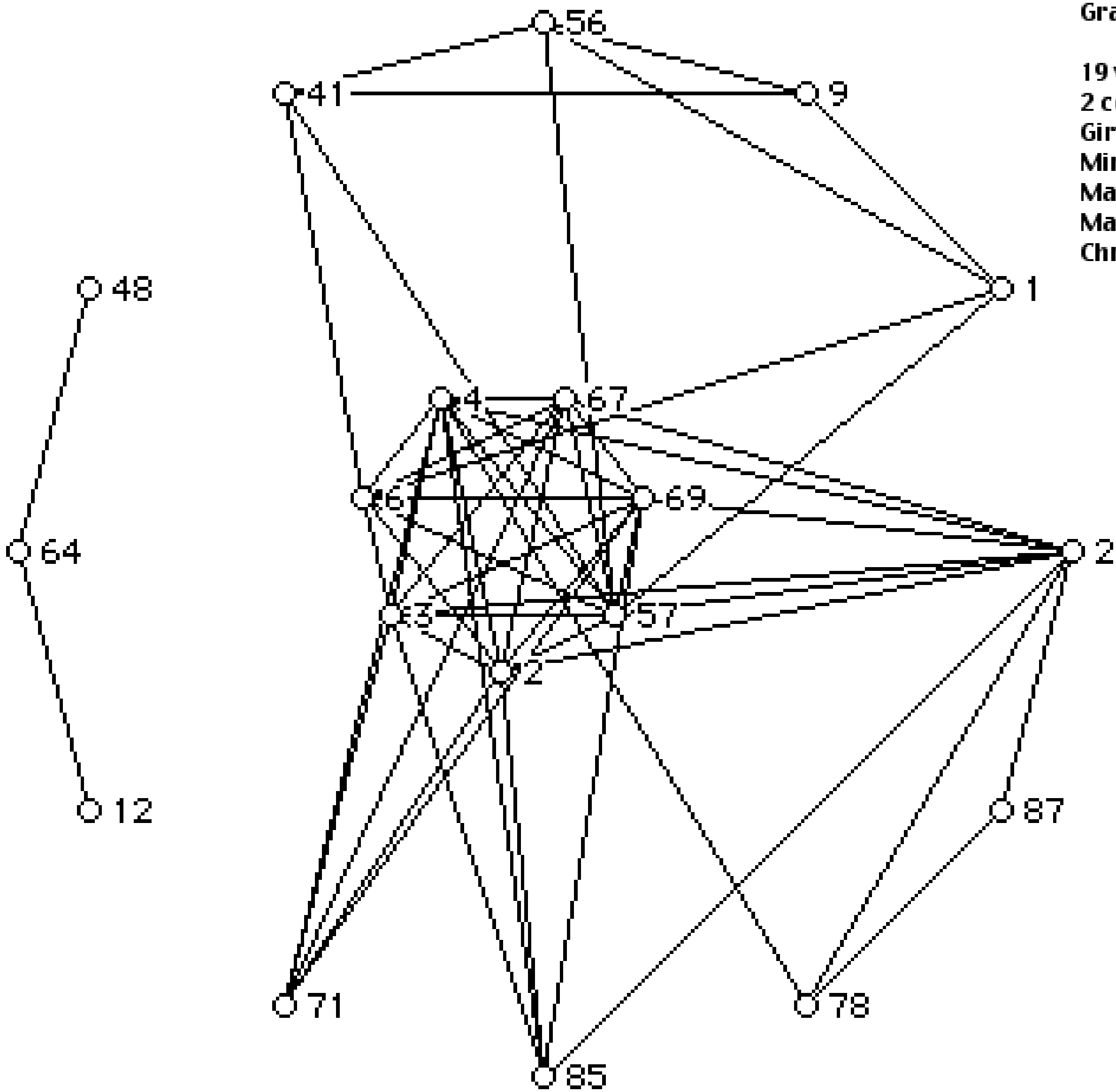
A	B	C	D	E	F	G	H	I	K	L
0	1	∞	2	∞	∞	1	∞	∞	∞	∞
	1	2	2	6	∞	1	∞	∞	∞	∞
		2	2	6	∞	1	4	∞	6	∞
		2	2	3	∞		3	∞	6	∞
		2		3	5		3	∞	6	∞
				3	5		3	6	6	7
				3	5			5	6	7
					5			5	6	7
								5	6	6
									6	6
										6



Extra-asiana käytiin läpi muistiverkkojen rakennetta ja toimintaa sekä sosiometrisiä verkkoja

Schema


sheep
wool
warm
straw-
berry
summer
socks
winter
ice-
cream
cold
snow
horse
cream
ball
milk
cow



PROPERTIES

Graph "sosio04c" :

19 vertices and 53 edges
2 connected components
Girth 3
Minimum degree 1 and Maximum degree 10
Maximum Independant Set : alpha = 7
Maximum Clique : omega = 7
Chromatic Number : khi = 7