

Paikallisuus hajautetuissa verkkoalgoritmeissa

Juhana Laurinharju

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 28. toukokuuta 2013

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Juhana Laurinharju			
Työn nimi — Arbetets titel — Title			
Paikallisuus hajautetuissa verkkoalgoritmeissa			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma		28. toukokuuta 2013	15
Tiivistelmä — Referat — Abstract			
<p>Olkoon n tietokonetta asetettu sykliin siten, että jokainen voi kommunikoida vain kahden naapurinsa kanssa. Verkon värytyksessä jokaiselle verkon solmulle täytyy antaa väri siten, että kahdella vierekkäisellä solmulla ei saa olla samaa väriä. Linial [2] on todistanut, että tällöin hajautettu algoritmi, joka haluaa värittää syklin solmut kolmella värillä käyttää tähän vähintään $\Omega(\log^* n)$ kommunikaatiokierrosta. Tässä työssä esitellään Linialin alarajatodistus.</p>			
Avainsanat — Nyckelord — Keywords			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	3
2	Määritelmiä	3
2.1	Verkko	3
2.2	Laskennan malli	4
2.3	Verkon väritys	4
2.4	Sykli	4
2.5	Iteroitu logaritmi \log^*	5
2.6	Näkymä	5
2.7	Naapurustoverkot	7
3	Sykliä ei voi 3-värittää alle $\log^* n$ kierroksessa	9
4	Yhteenveto	15
5	Lähteet	15

1 Johdanto

Hajautettu laskenta tarkastelee asetelmaa, jossa verkon solmut ovat tietokoneita jotka voivat kommunikoida keskenään verkon kaaria pitkin. Erityisesti hajautetussa asetelmassa tietokoneet haluavat ratkaista yhteistyössä jonkin kommunikaatioverkkoa koskevan laskennallisen ongelman.

Paikallinen algoritmi on vakioaikainen hajautettu algoritmi. Jokainen solmu saa siis laskettua tällöin oman vastauksensa vakiomäärässä kommunikointikierroksia. Tässä työssä esitellään Linialin [2] klassinen todistus sille, että hajautettu algoritmi käyttää ainakin $\Omega(\log^* n)$ kommunikaatiokierrosta $n:n$ solmun sykli-verkon värittämiseen kolmella värillä. Tämä tulos siis näyttää, ettei ole olemassa paikallista algoritmia joka värittäisi syklin kolmella värillä.

Tämä alarajatulos on myös asymptoottisesti tiukka, sillä Colen ja Vishkinin algoritmi [1] värittää n -solmuisen syklin hajautetusti kolmella värillä $\mathcal{O}(\log^* n)$ kierroksessa. Linialin näyttää alarajan deterministiselle algoritmille, mutta Naor [3] on yleistänyt tuloksen myös satunnaisuutta käyttäville hajautetuille algoritmeille.

2 Määritelmiä

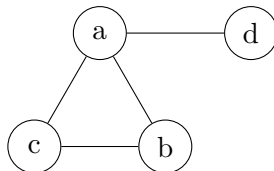
2.1 Verkko

Määritelmä 1. *Suuntaamaton verkko* on pari $G = (V, E)$, missä V on solmujoukko ja E on kaarijoukko. Kaari solmusta $v \in V$ solmuun $u \in V$ on kaksikko $\{v, u\} \in E$. Kaarta voidaan myös merkitä lyhyemmin vu . Jos G on suuntaamaton verkko, niin sen solmujoukkoon voidaan viitata myös merkinnällä $V(G)$ ja kaarijoukkoon merkinnällä $E(G)$.

Esimerkiksi verkko $G = (V, E)$, missä

$$\begin{aligned} V &= \{a, b, c, d\} \text{ ja} \\ E &= \{\{a, b\}, \{b, c\}, \{c, a\}, \{a, d\}\} \\ &= \{ab, bc, ca, ad\} \end{aligned}$$

näyttää seuraavalta



Määritelmä 2. Kaikilla positiivisilla kokonaisluvuilla k merkitään:

$$[k] = \{1, \dots, k\}$$

2.2 Laskennan malli

Olkoon $G = (V, E)$ suuntaamaton verkko. Verkon jokaisessa solmussa $v \in V$ on tietokone. Laskenta koostuu *kommunikaatiokierroksista*. Yhden kommunikaatiokierroksen aikana jokainen solmu voi

1. suorittaa mielivaltaista laskentaa,
2. lähettää viestin jokaiselle naapurilleen ja
3. vastaanottaa naapureiden lähettämät viestit.

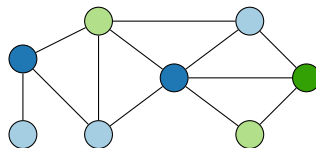
Lisäksi jokaiselle solmulle $v \in V$ on annettu yksikäsitteinen tunniste $ID(v) \in [|V|]$. Laskennan päätyttyä jokaisen solmun tulee tietää oma tulosteensa.

Hajautetun algoritmin aikavaativuutena tarkastellaan kommunikaatiokierrosten lukumäärää. Solmujen suorittama mielivaltainen laskenta jätetään siis aikavaativuusanalyysissä huomiotta. Hajautettua algoritmia ajatellaan ajettavan jossain tietokoneverkossa, jolloin viestien välittäminen tietokoneiden välillä on huomattavan hidasta verrattuna tietokoneen sisäiseen viestinvälitykseen ja laskentaan.

2.3 Verkon väritys

Määritelmä 3. Verkko on *väritetty*, jos jokaiseen solmuun $v \in V$ on liitetty jokin *väri* $c(v) \in \mathbb{N}$ ja kahdella vierekkäisellä solmulla ei koskaan ole samaa väriä. Tarkemmin, verkon $G = (V, E)$ *solmuväritys* on kuvaus $c: V \rightarrow [k]$ jollain luonnollisella luvulla $k \in \mathbb{N}$. Lisäksi vaaditaan, että jos verkossa on kaari solmusta v solmuun u , eli $vu \in E$, niin $c(v) \neq c(u)$.

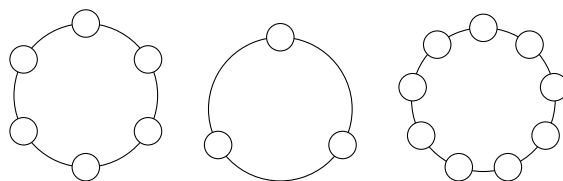
Verkon voi värittää k :lla värillä jos löytyy yllä olevan ehdon täyttävä kuvaus $c: V \rightarrow [k]$. Tällaista väritystä kutsutaan *k-väritykseksi*.



Jos verkkoa väritetään hajautetulla algoritmilla, niin jokaisen solmun tulee tietää oma värinsä laskennan päätyttyä.

2.4 Sykli

Määritelmä 4. Verkko on *sykli*, jos se on yhtenäinen ja sen jokaisella solmulla on tasan kaksi naapuria.

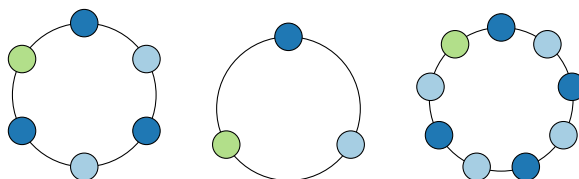


Tarkemmin sanoen, n -sykli, missä $n \geq 3$, on verkko $C_n = (V, E)$ jolla

$$V = \{v_1, v_2, \dots, v_n\}$$

$$E = \{v_i v_{i+1} \mid 1 \leq i < n\} \cup \{v_n v_1\}$$

Syklin voi aina värittää kolmella värillä.



2.5 Iteroitu logaritmi \log^*

Määritelmä 5. *Iteroitu logaritmi* \log^* kertoo kuinka monta kertaa luvusta täytyy ottaa logaritmi, kunnes lopputulos on korkeintaan yksi. Tarkemmin,

$$\log^* x = \begin{cases} 0, & \text{jos } x \leq 1, \\ 1 + \log^*(\log x), & \text{muutoin.} \end{cases}$$

Esimerkiksi

$$\begin{aligned} \log^* 16 &= \log^* 2^{2^2} = 1 + \log^* 2^2 \\ &= 2 + \log^* 2 = 3 + \log^* 1 = 3 \end{aligned}$$

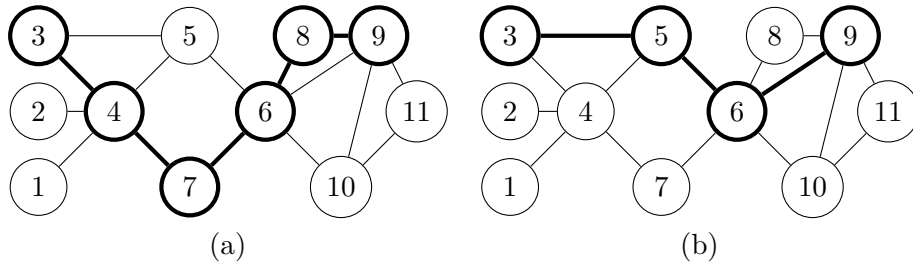
ja

$$\begin{aligned} \log^* 65536 &= \log^* 2^{2^{2^2}} = 1 + \log^* 16 \\ &= 4, \end{aligned}$$

joten $\log^* n$ on arvoltaan pienempi kuin 5 kun $n < 2^{65536}$. Iteroitu logaritmi on siis äärimmäisen hitaasti kasvava funktio.

2.6 Näkymä

Määritelmä 6. Verkon G *polku* on jono $P = (p_1, \dots, p_n)$, missä jokainen $p_i \in V(G)$ on verkon G solmu, kahden jonon perättäisen solmun välillä täytyy aina olla kaari ja lisäksi sama solmu ei saa esiintyä jonossa kahdesti. Siis kaikilla $i \in [n-1]$ täytyy olla voimassa ehto $p_i p_{i+1} \in E(G)$. Lisäksi kaikilla $i, j \in [n], i \neq j$ täytyy olla voimassa ehto $p_i \neq p_j$. Polku P on polku solmusta p_1 solmuun p_n .



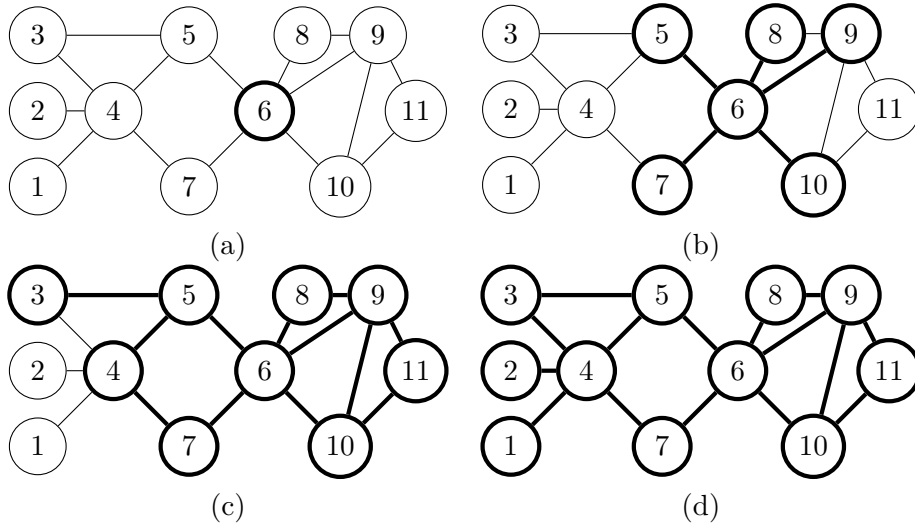
Kuva 1: (a) Polku $(3, 4, 7, 6, 8, 9)$ solmusta 3 solmuun 9. (b) Lyhin polku solmusta 3 solmuun 9.

Määritelmä 7. Polun $P = (p_1, \dots, p_n)$ pituus on sen kaarten lukumäärä. Siis polun P pituus on $n - 1$.

Kuvan 1 (a) polun pituus on 5.

Määritelmä 8. Kahden verkon G solmun $u, v \in V(G)$ välinen *etäisyys* verkossa G , on lyhimmän solmusta u solmuun v kulkevan polun pituus.

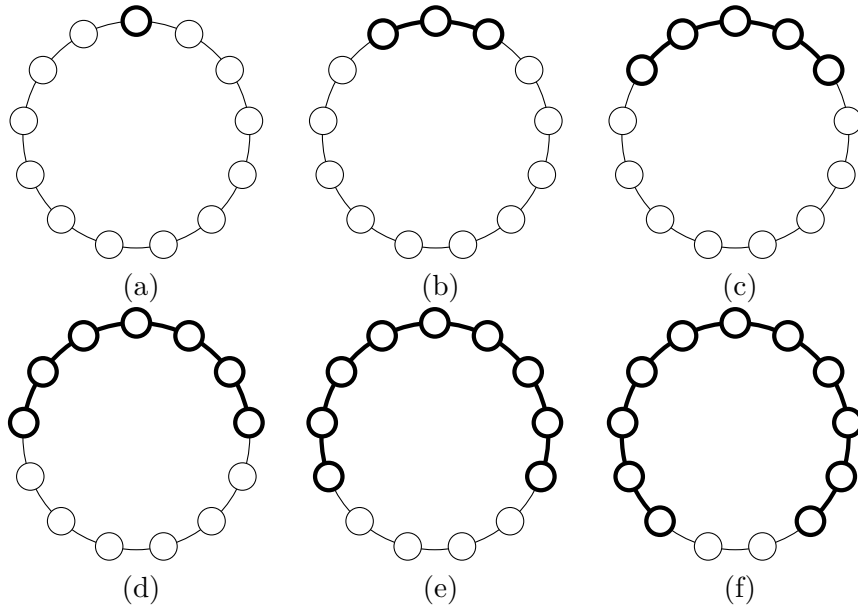
Kuvassa 1 (b) on havainnollistettu lyhintä polkua solmusta 3 solmuun 9. Solmujen 3 ja 9 välinen etäisyys on 3.



Kuva 2: (a) Solmun 6 näkemä alue verkossa ennen ensimmäistä kierrosta. (b)-(d) Solmun 6 näkemä alue verkossa kierrosten 1-3 jälkeen.

Hajautetussa algoritmossa solmu $v \in V$ saa k :ssa kierroksessa selville oman k -ympäristönsä. Toisaalta solmu ei pysty tässä ajassa saamaan mitään selville solmuista, joiden etäisyys v :stä on yli k . Kuvassa 2 havainnollistetaan solmun näkymää.

Hajautettu algoritmi, jonka ajoaika on k kierrosta on siis funktio, jonka lähtöjoukkona on solmujen mahdolliset k -ympäristöt.



Kuva 3: Solmun näkemä alue syklissä: (a) Ennen ensimmäistä kommunikaatiokierrosta solmu näkee vain itsensä. (b)-(f) Solmun näkemä alue 1-5 kommunikaatiokierroksen jälkeen.

Erityisesti syklissä algoritmi, jonka ajoaika on k kierrosta, tekee päätöksensä k :n edeltäjän, k :n seuraajan ja oman tunnisteensa perusteella. Toisin sanoen solmun $v_l \in V(C_n)$ tuloste on funktio arvoilta

$$(\text{ID}(v_{l-k}), \text{ID}(v_{l-k+1}), \dots, \text{ID}(v_{l-1}), \text{ID}(v_l), \text{ID}(v_{l+1}), \dots, \text{ID}(v_{l+k})),$$

Missä yhteen- ja vähennyslaskut suoritetaan modulo n . Kuvassa 3 on havainnollistettu solmun näkemää aluetta syklissä eri kommunikaatiokierrosten jälkeen.

Erityisesti jos algoritmi tuottaa 3-väriyksen n -syklissä k :ssa kierroksessa, niin täytyy olla olemassa sellainen funktio $f : [n]^{2k+1} \rightarrow [3]$, joka tuottaa laillisen 3-väriyksen riippumatta siitä miten solmuille on annettu tunnisteet.

2.7 Naapurustoverkot

Määritelmä 9. Naapurustoverkko $B_{t,n} = (V, E)$ on verkko, jonka solmujoukon V muodostaa vektoreiden (x_1, \dots, x_{2t+1}) joukko, missä x_i :t ovat keskenään erisuuria kokonaislukuja joukosta $[n]$. Verkossa $B_{t,n}$ solmut muotoa

$$(x_1, \dots, x_{2t+1}) \text{ ja } (y, x_1, x_2, \dots, x_{2t})$$

ovat naapureita, kun $y \neq x_{2t+1}$.

Naapurustoverkon $B_{t,n}$ solmu on siis syklissä ajatun hajautetun algoritmin näkymä t kierroksen jälkeen. Kahden näkymän välillä on naapurustoverkossa

kaari, jos ne ovat vierekkäisten solmujen näkymät jossain yksikäsitteisillä tunnisteilla varustetussa n -syklissä.

Määritelmä 10. Solmun $v \in V(G)$ *asteluku* on sen naapureiden lukumäärä verkossa G . Tarkemmin, solmun $v \in V(G)$ asteluku on

$$|\{e \in E(G) \mid v \in e\}|.$$

Verkossa $B_{t,n}$ on

$$n(n-1)(n-2) \cdots (n-2t)$$

solmua ja sen kaikkien solmujen asteluku on $2(n-2t-1)$.

Hajautettu algoritmi, joka 3-värittää syklin t kierroksessa antaa jokaiselle solmulle värin tarkastelemalla vain sen t -ympäristön tunnisteita. Väritysalgoritmi on siis todellisuudessa funktio $c : V(B_{t,n}) \rightarrow [3]$, sillä naapurustoverkossa $B_{t,n}$ on solmuina kaikki mahdolliset n -syklin t -naapurustot.

Nyt c on myös laillinen 3-väritys verkolle $B_{t,n}$, sillä jos c antaa solmuille

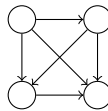
$$(x_1, \dots, x_{2t}, x_{2t+1}) \text{ ja } (y, x_1, \dots, x_{2t})$$

saman värin, niin se antaa myös syklissä kahdelle vierekkäiselle solmulle saman värin kun syklissä esiintyy pätkä

$$y, x_1, x_2, \dots, x_{2t+1}.$$

Siis jos näytetään, että verkkoa $B_{t,n}$ ei voi 3-värittää, niin ei voi myöskään olla hajautettua algoritmia joka värittäisi n -syklin kolmella värillä t kierroksessa.

Määritelmä 11. *Suunnatto verkko* $G = (V, E)$ on verkko, jossa kaarilla on suunta. Suunnatussa verkossa on kaari solmusta $u \in V$ solmuun $v \in V$ jos $(u, v) \in E$.

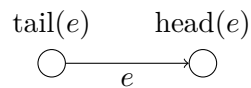


Määritelmä 12. Suunnatulla kaarella $e = (u, v) \in E$ on *kärki*

$$\text{head}(e) = v$$

ja *häntä*

$$\text{tail}(e) = u.$$



Määritelmä 13. Verkon G *väritysluku* $\chi(G)$ on pienin määrä värejä, jolla sen voi värittää.

3 Sykliä ei voi 3-värittää alle $\log^* n$ kierroksessa

Väritysluvun $\chi(B_{t,n})$ alaraja todistetaan käyttäen suunnattujen verkkojen $D_{s,n}$ perhettä. Suunnatut verkot $D_{s,n}$ liittyvät läheisesti naapurustoverkkoihin $B_{t,n}$. Verkon $D_{s,n}$ solmujoukon $V(D_{s,n})$ muodostavat kaikki vektorit muotoa

$$(a_1, a_2, \dots, a_s),$$

joilla pätee

$$1 \leq a_1 < a_2 < \dots < a_s \leq n.$$

Solmusta (a_1, \dots, a_s) lähtee kaari muotoa

$$(a_2, \dots, a_s, b)$$

oleviin solmuihin, joilla $a_s < b \leq n$.

Lemma 14. *Naapurustoverkko $B_{t,n}$ pitää sisällään aliverkkona suunnatun verkon $D_{2t+1,n}$.*

Todistus. Olkoon

$$\bar{x} = (x_1, \dots, x_{2t+1}) \in V(D_{2t+1,n})$$

verkon $D_{2t+1,n}$ solmu. Koska verkossa $D_{2t+1,n}$ on solmuvektoreilla suuruusjärjestysehto

$$x_1 < x_2 < \dots < x_{2t+1},$$

niin erityisesti nämä vektorin alkiot ovat keskenään erisuuria, joten vektori \bar{x} on myös verkon $B_{t,n}$ solmu.

Lisäksi jos solmulla \bar{x} on verkossa $D_{2t+1,n}$ naapuri

$$\bar{y} = (x_2, \dots, x_{2t+1}, y) \in V(D_{2t+1,n}),$$

eli verkossa $D_{2t+1,n}$ on suunnattu kaari $(\bar{x}, \bar{y}) \in E(D_{2t+1,n})$, niin tällöin myös verkossa $B_{t,n}$ on näiden solmujen välillä kaari. Tämä johtuu siitä, että suunnatun verkon $D_{2t+1,n}$ kaarien suuruusjärjestysehdon nojalla $y > x_1$, joten erityisesti $y \neq x_1$. \square

Tästä erityisesti seuraa, että jokainen verkon $B_{t,n}$ väritys on myös verkon $D_{2t+1,n}$ väritys.

Lemma 15. *Verkkojen $B_{t,n}$ ja $D_{2t+1,n}$ värityslukuille pätee*

$$\chi(B_{t,n}) \geq \chi(D_{2t+1,n}).$$

Todistus. Olkoon $c: V(B_{t,n}) \rightarrow [k]$ verkon $B_{t,n}$ väritys. Nyt jos

$$(x, y) \in E(D_{2t+1,n})$$

on verkon $D_{2t+1,n}$ kaari, niin äskeisen lemmän nojalla solmujen x ja y välillä on kaari myös verkossa $B_{t,n}$. Koska c on verkon $B_{t,n}$ väritys, niin $c(x) \neq c(y)$. Siis c on väritys myös verkolle $D_{2t+1,n}$ ja erityisesti verkon $D_{2t+1,n}$ voi siis myös värittää k :lla värillä. \square

Määritelmä 16. Suunnatun verkon G *kaariverkko* $DL(G)$ on verkko, jonka solmuja ovat alkuperäisen verkon G kaaret ja kahden kaariverkon solmun

$$u, v \in V(DL(G)) = E(G)$$

välillä on kaari, jos $\text{head}(u) = \text{tail}(v)$.

Tarkemmin ilmaistuna,

$$V(DL(G)) = E(G)$$

$$E(DL(G)) = \{(v, u) \in E(G) \times E(G) \mid \text{head}(v) = \text{tail}(u)\}.$$

Lemma 17. $D_{1,n}$ on $n:n$ solmun täydellinen verkko, jossa kaaret on suunnattu pienemmästä solmusta suurempaan.

Todistus. Verkon $D_{s,n}$ määritelmä s :n arvolla 1 antaa seuraavan verkon:

$$\begin{aligned} V(D_{1,n}) &= \{(k) \mid 1 \leq k \leq n\} \\ E(D_{1,n}) &= \{((k), (l)) \mid k < l\}. \end{aligned}$$

Tässä verkossa jokaisen kahden solmun välillä on kaari tasan yhteen suuntaan. \square

Määritelmä 18. Kaksi suunnattua verkkoa G ja H ovat *isomorfiset* jos niiden solmujoukkojen välillä on olemassa kuvaus $\varphi: V(G) \rightarrow V(H)$ joka toteuttaa seuraavat ehdot

1. φ on bijektio
2. $(u, v) \in E(G)$ jos ja vain jos $(\varphi(u), \varphi(v)) \in E(H)$.

Nämä ehdot toteuttavaa kuvausta kutsutaan *isomorfismiksi*.

Lemma 19. Verkko $D_{s+1,n}$ on verkon $D_{s,n}$ kaariverkko. Tarkemmin,

$$D_{s+1,n} = DL(D_{s,n}).$$

Todistus. Ideana on samaistaa kaariverkon $DL(D_{s,n})$ kaari

$$((x_1, \dots, x_s), (x_2, \dots, x_s, y))$$

verkon $D_{s+1,n}$ solmun (x_1, \dots, x_s, y) kanssa. Määritellään siis verkkojen välille kuvaus

$$\varphi: V(DL(D_{s,n})) \rightarrow V(D_{s+1,n})$$

asettamalla

$$\varphi((x_1, \dots, x_s), (x_2, \dots, x_s, y)) = (x_1, \dots, x_s, y).$$

Näytetään, että φ on isomorfismi. Ensinnäkin φ on bijektio, sillä sille löytyy seuraava käänteiskuvaus $\psi: D_{s+1,n} \rightarrow DL(D_{s,n})$:

$$\psi(x_1, \dots, x_n, x_{n+1}) = ((x_1, \dots, x_n), (x_2, \dots, x_{n+1})).$$

Kuvaus ψ on kuvauksen φ käänteiskuvaus, sillä

$$\begin{aligned} (\psi \circ \varphi)((x_1, \dots, x_n), (x_2, \dots, x_n, y)) &= \psi(\varphi((x_1, \dots, x_n), (x_2, \dots, x_n, y))) \\ &= \psi(x_1, \dots, x_n, y) \\ &= ((x_1, \dots, x_n), (x_2, \dots, x_n, y)) \end{aligned}$$

ja

$$\begin{aligned} (\varphi \circ \psi)(x_1, \dots, x_n, x_{n+1}) &= \varphi(\psi(x_1, \dots, x_n, x_{n+1})) \\ &= \varphi((x_1, \dots, x_n), (x_2, \dots, x_n, x_{n+1})) \\ &= (x_1, \dots, x_n, x_{n+1}). \end{aligned}$$

Vielä täytyy näyttää, että kuvaus toteuttaa isomorfiaehdon. Olkoon

$$\begin{aligned} \bar{x}_1 &= (x_1, \dots, x_n) \in V(DL(D_{s,n})), \\ \bar{x}_2 &= (x_2, \dots, x_{n+1}) \in V(DL(D_{s,n})), \\ \bar{y}_1 &= (y_1, \dots, y_n) \in V(DL(D_{s,n})) \text{ ja} \\ \bar{y}_2 &= (y_2, \dots, y_{n+1}) \in V(DL(D_{s,n})) \end{aligned}$$

verkon $D_{s,n}$ solmuja. Tällöin erityisesti

$$\begin{aligned} (\bar{x}_1, \bar{x}_2) &\in V(DL(D_{s,n})) \text{ ja} \\ (\bar{y}_1, \bar{y}_2) &\in V(DL(D_{s,n})) \end{aligned}$$

ovat kaariverkon $DL(D_{s,n})$ solmuja. Jos $((\bar{x}_1, \bar{x}_2), (\bar{y}_1, \bar{y}_2)) \in E(DL(D_{s,n}))$ on kaariverkon $DL(D_{s,n})$ kaari, niin kaariverkon määritelmän nojalla

$$\begin{aligned} \text{head}((\bar{x}_1, \bar{x}_2)) &= \text{tail}((\bar{y}_1, \bar{y}_2)) && \implies \\ \bar{x}_2 &= \bar{y}_1 && \implies \\ (x_2, \dots, x_{n+1}) &= (y_1, \dots, y_n) && \implies \\ y_1 &= x_2, \dots \text{ ja } y_n &= x_{n+1}. \end{aligned}$$

Koska $\bar{x}_1 \in V(D_{s,n})$ ja $\bar{y}_2 \in V(D_{s,n})$, niin $x_1 < y_1 = x_2$ ja $y_n < y_{n+1}$. Tällöin

$$\begin{aligned} (\varphi(\bar{x}_1, \bar{x}_2), \varphi(\bar{y}_1, \bar{y}_2)) &= ((x_1, \dots, x_{n+1}), (y_1, \dots, y_{n+1})) \\ &= ((x_1, y_1, y_2, \dots, y_n), (y_1, \dots, y_{n+1})) \in E(D_{s+1,n}). \end{aligned}$$

Ollaan siis näytetty ensimmäinen kahdesta implikaatiosta:

$$((\bar{x}_1, \bar{x}_2), (\bar{y}_1, \bar{y}_2)) \in E(\text{DL}(D_{s,n})) \implies (\varphi(\bar{x}_1, \bar{x}_2), \varphi(\bar{y}_1, \bar{y}_2)) \in E(D_{s+1,n}).$$

Toisaalta jos

$$\begin{aligned} \bar{x} &= (x_1, \dots, \bar{x}_{n+1}) \in V(D_{s+1,n}) \text{ ja} \\ \bar{y} &= (y_1, \dots, \bar{y}_{n+1}) \in V(D_{s+1,n}). \end{aligned}$$

ovat verkon $D_{s+1,n}$ solmuja joiden välillä on kaari, eli $(\bar{x}, \bar{y}) \in E(D_{s+1,n})$, niin tällöin

$$x_2 = y_1, \dots \text{ ja } x_{n+1} = y_n.$$

Nyt

$$\begin{aligned} \bar{x}_1 &= (x_1, \dots, x_n), \\ \bar{x}_2 &= (x_2, \dots, x_{n+1}), \\ \bar{y}_1 &= (y_1, \dots, y_n) \text{ ja} \\ \bar{y}_2 &= (y_2, \dots, y_{n+1}) \end{aligned}$$

ovat verkon $D_{s,n}$ solmuja joilla pätee

$$\begin{aligned} (\bar{x}_1, \bar{x}_2) \in E(D_{s,n}) &\implies (\bar{x}_1, \bar{x}_2) \in V(\text{DL}(D_{s,n})) \text{ ja} \\ (\bar{y}_1, \bar{y}_2) \in E(D_{s,n}) &\implies (\bar{y}_1, \bar{y}_2) \in V(\text{DL}(D_{s,n})). \end{aligned}$$

Koska $\bar{x}_2 = \bar{y}_1$, niin kaariverkossa $\text{DL}(D_{s,n})$ on kaari

$$((\bar{x}_1, \bar{x}_2), (\bar{y}_1, \bar{y}_2)) \in E(\text{DL}(D_{s,n})).$$

Koska

$$\begin{aligned} (\bar{x}_1, \bar{x}_2) &= \varphi^{-1}(\bar{x}) \text{ ja} \\ (\bar{y}_1, \bar{y}_2) &= \varphi^{-1}(\bar{y}) \end{aligned}$$

niin väite on todistettu. Ollaan siis näytetty, että jos $(\bar{x}, \bar{y}) \in E(D_{s+1,n})$ on verkon $D_{s+1,n}$ kaari, niin sen solmujen alkukuvien välillä on myös kaari. Tarkemmin sanottuna

$$(\bar{x}, \bar{y}) \in E(D_{s+1,n}) \implies (\varphi^{-1}(\bar{x}), \varphi^{-1}(\bar{y})) \in E(\text{DL}(D_{s,n})).$$

Ollaan siis näytetty, että löytyy bijektio $\varphi: V(\text{DL}(D_{s,n})) \rightarrow D_{s+1,n}$ joka toteuttaa isomorfiaehdon, joten verkot $\text{DL}(D_{s,n})$ ja $D_{s+1,n}$ ovat isomorfisina olennaisesti sama verkko. \square

Verkkojen $D_{s,n}$ välillä on nyt siis tarkkaan tunnettu yhteys, kun s :n arvot vaihtelevat. Lisäksi verkko $D_{1,n}$ on rakenteeltaan yksinkertainen täydellinen verkko.

Lemma 20.

$$\chi(D_{1,n}) = n.$$

Todistus. Koska täydellisessä verkossa on kaari verkon jokaisen solmuparin välillä, täytyy sen jokaisella solmulla olla eri väri kuin millään muulla solmulla. Siispä verkon $D_{1,n}$ värittämiseen tarvitaan n väriä. \square

Seuraavan lemmän avulla saadaan yhteys verkkojen $D_{s,n}$ ja $D_{s+1,n}$ värityslukujen välille.

Lemma 21. *Olkoon G kaariverkko. Tällöin*

$$\chi(\text{DL}(G)) \geq \log \chi(G).$$

Todistus. Olkoon $\Psi: \text{DL}(G) \rightarrow [k]$ kaariverkon $\text{DL}(G)$ k -väritys. Koska verkko $\text{DL}(G)$ on verkon G kaariverkko, niin kuvaus Ψ antaa jokaiselle verkon G kaarelle värin. Jos $u, v \in E(G)$ ovat verkon G perättäisiä kaaria, eli $\text{head}(u) = \text{tail}(v)$, niin Ψ antaa niille eri värit, $\Psi(u) \neq \Psi(v)$.

Muodostetaan verkolle G väritys $c: G \rightarrow \mathcal{P}([k])$ joka värittää G :n 2^k värillä. Olkoon $x \in V(G)$ verkon G solmu. Määritellään x :n väri seuraavasti:

$$c(x) = \{\Psi(u) \mid \text{tail}(u) = x\}.$$

Solmun x väri on siis joukko, jossa on kaikkien solmusta x lähtevien kaarten värit värityksessä Ψ . Jotta c olisi laillinen väritys, se ei saa antaa samaa väriä kahdelle naapurisolmulle. Olkoon $u = (x, y) \in E(G)$ verkon G kaari ja siis $y \in V(G)$ solmun x naapuri. Nyt $\text{tail}(u) = x$, joten u :n väri kuuluu x :n värijoukkoon. Tarkemmin ilmaistuna, $\Psi(u) \in c(x)$.

Toisaalta jos $\Psi(u) \in c(y)$, niin tällöin löytyy kaari $v \in E(G)$, joka lähtee solmusta y , eli $\text{tail}(v) = y$ ja lisäksi jolle Ψ antaa saman värin kuin u :lle, eli $\Psi(x) = \Psi(y)$. Tämä ei kuitenkaan ole mahdollista, sillä $\text{head}(u) = y = \text{tail}(v)$, jolloin Ψ ei olisikaan laillinen verkon $\text{DL}(G)$ väritys. Siis c antaa vierekkäisille solmuille aina toisistaan eroavan värin, joten se täyttää väritysehdon.

TODO: Tämä on vähän wall of text, voisi keventää jotenkin

Väritys c antaa verkon G solmuille väriksi jonkin osajoukon värityksen Ψ väreistä. Koska k :n alkion joukolla on yhteensä 2^k osajoukkoa, niin c värittää verkon G korkeintaan 2^k värillä.

Ollaan siis näytetty, että jos verkon G kaariverkon $\text{DL}(G)$ voi värittää k :lla värillä, niin G :n voi värittää 2^k värillä. Toisin sanoen

$$\begin{aligned} \chi(G) \leq 2^{\chi(\text{DL}(G))} &\implies \\ \log \chi(G) &\leq \chi(\text{DL}(G)) \end{aligned}$$

\square

Määritelmä 22. Luonnollisella luvulla $k \in \mathbb{N}$ merkitään

$$\log^{(k)} n = \overbrace{\log \log \dots \log}^k n.$$

Luku $\log^{(k)} n$ on siis se lopputulos joka saadaan, kun luvusta n otetaan logaritmi k kertaa.

Aikaisemmista lemmoista saadaan nyt lopulta verkkojen $D_{s,n}$ väritysluvulle $\chi(D_{s,n})$ alaraja.

Korollaaari 23.

$$\chi(D_{s+1,n}) \geq \log \chi(D_{s,n}).$$

Erityisesti, koska $\chi(D_{1,n}) = n$, niin edellisestä korollaarista seuraa, että

$$\begin{aligned} \chi(D_{s,n}) &\geq \log^{(s)} \chi(D_{1,n}) \implies \\ \chi(D_{s,n}) &\geq \log^{(s)} n \end{aligned}$$

Lause 24. Hajautettu algoritmi ei voi värittää n :n solmun sykliä kolmella värillä alle $\log^* n$ kierroksessa.

Todistus. Hajautettu algoritmi voi värittää n -syklin kolmella värillä t kierroksessa jos verkon $B_{t,n}$ voi värittää kolmella värillä. Yhdistämällä edellisten lemmaa tulokset saadaan seuraava yhtälö:

$$3 \geq \chi(B_{t,n}) \geq D_{2t+1,n} \geq \log^{(2t)} n.$$

Josta voidaan edelleen päätellä seuraavaa:

$$\begin{aligned} \log^{(2t)} n &\leq 3 \implies \\ \log^{(2t+1)} n &\leq 2 \implies \\ \log^{(2t+2)} n &\leq 1. \end{aligned}$$

Luku $\log^* n$ on pienin määrä toistokertoja, jolla luvusta n saadaan logaritmeja ottamalla korkeintaan 1. Äskeisen nojalla $2(t+1)$ toistokertaa varmasti riittää tähän, joten

$$\begin{aligned} \log^* n &\leq 2(t+1) \implies \\ \frac{1}{2} \log^* n - 1 &\leq t. \end{aligned}$$

Ollaan siis näytetty, että hajautettu algoritmi joka värittää n -syklin kolmella värillä käyttää siihen aikaa vähintään $\frac{1}{2} \log^* n - 1$ kierrosta. □

4 Yhteenveto

Hajautettu algoritmi vaatii siis vähintään $\Omega(\log^* n)$ kierrosta n -syklin värittämiseen kolmella värillä. Tämä alaraja on myös asymptoottisesti tiukka, sillä Colen ja Vishkinin algoritmi värittää n -syklin kolmella värillä ajassa $\mathcal{O}(\log^* n)$.

Tässä esitetty todistus ei ota kantaa satunnaisalgoritmeihin, mutta Naor [3] on näyttänyt, että vastaava alaraja pätee myös satunnaisalgoritmeille.

5 Lähteet

- [1] Cole, Richard ja Vishkin, Uzi: *Deterministic coin tossing with applications to optimal parallel list ranking*. Information and Control, 70(1):32–53, 1986.
- [2] Linial, Nathan: *Locality in Distributed Graph Algorithms*. SIAM Journal on Computing, 21(1):193–201, 1992.
- [3] Naor, Moni: *A Lower Bound on Probabilistic Algorithms for Distributive Ring Coloring*. SIAM Journal on Discrete Math, 4(3):409–412, 1991.