Mincost-flow-algoritmit

Tuukka Korhonen

October 31, 2016

Mincost-flow

Olkoon $G = (V, E, U, C, s, t)$ virtausverkko, jossa $U(e) \in \mathbb{Z}_+$ on kaaren e kapasiteetti, $C(e) \in \mathbb{R}$ on kaaren e hinta ja s ja t ovat lähtö- ja kohdesolmut. Lisäksi olkoon k kokonaisluku joka tarkoittaa kuinka paljon virtausta lähetetään. Merkinnällä uv tarkoitetaan kaarta solmusta u solmuun v. Oletetaan tämä merkintä yksikäsitteiseksi eli kahden solmun välillä ei saa olla useaa kaarta. Lisäksi oletetaan että jos verkossa on kaari uv, siinä ei ole kaarta vu. Nämä oletukset ovat vain teoriaa varten ja niitä ei tehdä implementaatiossa. MINCOST-FLOW-ongelmassa tavoitteena on löytää minimihintainen virtaus jota nyt merkitään funktiona $f : E \to \mathbb{Z}$ joka:

Minimoi

$$\sum_{uv \in E} C(uv) f(uv)$$

Toteuttaa ehdot:
Säilyvyysehto

$$\sum_{v \in V} f(uv) = \sum_{v \in V} f(vu) \text{ kaikilla } u \in V \setminus \{s, t\}$$

Kapasiteettiehto

$$0 \leq f(uv) \leq U(uv) \text{ kaikilla } uv \in E$$

Virtauksen määrä

$$\sum_{v \in V} f(sv) = k \text{ ja } \sum_{v \in V} f(vt) = k$$

Lisäksi määritellään että MINCOST-FLOW-ongelmassa verkossa ei saa olla suunnattuja syklejä, joiden hintojen summa on negatiivinen.

Toisin sanoen lähetetään k yksikköä virtausta lähtösolmusta s kohdesolmuun t niin että virtauksen käyttämien kaarien hinta on mahdollisimman pieni. Tämän voi ajatella esimerkiksi ongelmana jossa verkon solmut ovat kaupunkeja. Kaupungissa s on tehdas josta pitää lähettää joka päivä k tonnia tavaraa varastoon joka on kaupungissa t. Lisäksi tiedetään että kaupunkien välillä on junayhteyksiä (eli kaaria). Kaaren uv kapasiteetti kertoo paljonko sillä junayhteydellä voi kuljettaa tavaraa päivässä kaupungista u kaupunkiin v. Kaarien hinnat kertovat paljonko maksaa yhden tonnin tavaraa kuljettaminen sillä junayhteydellä. Nyt halutaan kuljettaa k tonnia tavaraa tehtaasta varastoon päivittäin ja minimoida kuljetuksen hinta.

Syy negatiivisten syklien kieltämiseen on, että saadaan yksinkertaisempia toimivia algoritmeja ja tämä määritelmä sopii ongelman luonteeseen jossa lähetetään tavaraa lähtösolmusta kohdesolmuun. Jos negatiivisia syklejä saisi olla, optimaalinen ratkaisu voisi pyörittää virtausta jossain muualla kuin reiteillä lähtösolmusta kohdesolmuun. Myöhemmin esitetään MINCOST CIRCULATION ongelma, jossa negatiiviset syklit ovat sallittuja.

Shortest-Augmenting-Path-algoritmi

Shortest-Augmenting-Path-algoritmi (lyhyemmin SAP-algoritmi) etsii minimihintaisen virtauksen lähettämällä virtausta lyhintä reittiä kaarien hintojen suhteen lähtösolmusta kohdesolmuun. Tällaista lyhintä reittiä kutsutaan *augmentoivaksi poluksi*. Algoritmi on muunnelma Ford-Fulkerson-algoritmista maksimivirtauksen etsimiseen. Intuitiota saa miettimällä miten minimihintainen virtaus etsittäisiin jos virtausta lähetettäisiin vain yksi yksikkö: se olisi lyhin reitti lähtösolmusta kohdesolmuun.

Jäännösverkko

Jotta virtausta voidaan lähettää useampia yksiköitä, täytyy olla tapa korjata aiemmin lähetettyä virtausta. Lähetetään virtausta siis jäännösverkossa. Jäännösverkko toimii kuten Ford-Fulkersonin jäännösverkko: virtausta voidaan peruuttaa kulkemalla jäännösverkossa vastakkaiseen suuntaan menevää kaarta pitkin. Virtausta peruuttaessa kaaren hinta on alkuperäisen kaaren hinnan vastaluku. Tästä seuraa että negatiivisia kaaria syntyy jäännösverkkoon vaikka niitä ei olisi verkossa alunperin.

Sanotaan että G_f on virtauksen f jäännösverkko. Määritellään kapasiteetit jäännösverkossa:

$$U_f(uv) = \begin{cases} U(uv) - f(uv) \text{ jos } uv \in E \\ f(vu) \text{ jos } vu \in E \\ 0 \text{ muuten} \end{cases}$$

Määritellään hinnat jäännösverkossa:

$$C_f(uv) = \begin{cases} C(uv) \text{ jos } uv \in E \\ -C(vu) \text{ jos } vu \in E \\ 0 \text{ muuten} \end{cases}$$

Sanotaan että kaari uv on jäännösverkossa jos $U_f(uv) > 0$.

Jos jäännösverkossa on polku $s \rightsquigarrow t$ jonka pienin kapasiteetti on u, niin sitä polkua pitkin voidaan lähettää maksimissaan u yksikköä virtausta ja yhden lähetetyn virtausyksikön hinnaksi tulee polun kaarien hintojen summa. Tämä kasvattaa virtauksen määrää. Toinen sallittu tapa muuttaa virtausta on lähettää virtausta jotain jäännösverkon sykliä pitkin, maksimissaan syklin pienimmän kapasiteetin verran. Laskemalla voidaan tarkistaa että nämä muutokset säilyttävät virtauksen säilyvyys- ja kapasiteettiehdot.

SAP-algoritmi

SAP-algoritmi lähettää virtausta jäännösverkon lyhintä $s \rightsquigarrow t$-polkua pitkin kunnes virtausta on lähetetty k yksikköä tai maksimivirtaus on löydytty. Tässä lyhin polku tarkoittaa lyhintä polkua jäännösverkon hintojen suhteen. Lähettämällä virtausta aina lyhintä polkua pitkin jäännösverkkoon ei koskaan synny negatiivisia syklejä. (Negatiivinen sykli on sykli jonka kaarien hintojen summa on negatiivinen.) Tästä seuraa myös tapa nähdä että virtaus on optimaalinen: ainoa tapa muuttaa virtauksen hintaa muuttamatta virtaukseen määrää on lähettää virtausta jotain sykliä pitkin. Jos negatiivisia syklejä ei ole, virtauksen hintaa ei voi muuttaa mitenkään pienemmäksi. Seuraavaksi todistetaan nämä väitteet.

Virtauksen optimaalisuus

Lemma: k-virtaus on optimaalinen jos jäännösverkossa ei ole sykliä jonka hinta on negatiivinen.

Todistus: Olkoon f joku k-virtaus niin että G_f:ssä ei ole negatiivista sykliä. Oletetaan että on olemassa virtaus f', joka on k-virtaus ja jonka hinta on pienempi kuin f:n hinta. Määritellään $(f' - f)(uv) = f'(uv) - f(uv)$. Nyt $f' - f$ on virtaus joka toteuttaa säilyvyysehdon kaikissa solmuissa, mukaan lukien lähtö- ja kohdesolmut, koska f' ja f ovat molemmat k-virtauksia. Virtauksen $f' - f$ hinta on negatiivinen. Lisäksi $f' - f$ on sallittu virtaus G_f:ssä. Säilyvyysehdon toteuttavan virtauksen voi hajottaa joukoksi syklejä jotka toteuttavat säilyvyysehdon, koska jos otetaan mikä tahansa sykli virtauksesta pois, jäljelle jää virtaus joka toteuttaa säilyvyysehdon. Siispä kun $f' - f$ hajotetaan joukoksi syklejä, niin ainakin yhden syklin hinta on negatiivinen. Sitä sykliä pitkin voidaan lähettää virtausta G_f:ssä, joten G_f:ssä on negatiivinen sykli. Seuraa ristiriita, joten f on optimaalinen.

Potentiaalifunktio

Sanotaan että potentiaalifunktio jäännösverkossa G_f on funktio $p : V \to \mathbb{R}$, joka toteuttaa $p(u) + C_f(uv) \geq p(v)$ kaikilla jäännösverkon kaarilla uv. Jos G_f:ssä ei ole negatiivista sykliä, potentiaalifunktio on olemassa, sillä potentiaalifunktioksi voidaan ottaa $p(u)$ on lyhimmän reitin pituus solmusta s solmuun u. Määritellään että kaaren uv *vähennetty hinta* on $C_{fr}(uv) = p(u) + C_f(uv) - p(v)$. Nähdään että syklin hinta vähennetyissä hinnoissa on sama kuin syklin hinta koska potentiaalit kumoavat toisensa. Nähdään myös että kaikki vähennetyt hinnat ovat epänegatiivisia, josta on helppo nähdä että jos on olemassa potentiaalifunktio, niin negatiivisia syklejä ei ole.

Lemma: Jos jäännösverkossa ei ole negatiivista sykliä, niin lähettämällä virtausta lyhintä $s \rightsquigarrow t$ polkua pitkin sinne ei synny negatiivista sykliä.

Todistus: Kun lähetetään virtausta lyhintä $s \rightsquigarrow t$ polkua pitkin, voidaan olemassaolevasta potentiaalifunktiosta p päivittää potentiaalifunktio p' joka on pätevä uudessa jäännösverkossa. Olkoon $sp(u)$ lyhimmän reitin pituus solmusta s solmuun u kun tarkastellaan reitin hintaa vähennetyissä hinnoissa. Jos ei ole polkua solmusta s solmuun u, niin $sp(u)$ on joku riittävän suuri vakio. Uusi potentiaalifunktio on $p'(u) = p(u) + sp(u)$. Tämä toteuttaa potentiaalifunktion ehdot kaikilla kaarilla jotka olivat jo jäännösverkossa koska $sp(u) + C_{fr}(uv) \geq sp(v)$ koska sp on lyhin-reitti-funktio joten $sp(u) + p(u) + C_f(uv) - p(v) \geq sp(v)$ joten $sp(u) + p(u) + C_f(uv) \geq sp(v) + p(v)$ joten $p'(u) + C_f(uv) \geq p'(v)$. (Huomaa että tämä pätee myös jos solmusta s ei ole reittiä solmuun u koska vähennetyt hinnat ovat epänegatiivisia.) Lisäksi verkkoon saattoi ilmestyä uusia kaaria kun lyhintä $s \rightsquigarrow t$ polkua päinvastaiseen suuntaan menevien kaarien kapasiteetti

kasvoi. Kun u ja v ovat lyhimmällä polulla peräkkäin olevat solmut, $sp(u) + C_{fr}(uv) = sp(v)$ joten $p'(u) + C_f(uv) = p'(v)$ joten $p'(v) - C_f(uv) = p'(u)$ eli potentiaalifunktio toimii kaikilla uusilla kaarilla jotka ilmestyivät jäännösverkkoon.

Aikavaativuus

Tässä määritelty SAP-algoritmi ei anna mitään tietoa tarvittavien augmentoivien polkujen määrästä. Nähdään kuitenkin että virtauksen määrä kasvaa ainakin yhdellä, joten augmentoivia polkuja on maksimissaan k. Sanotaan että verkossa on n solmua, m kaarta ja suurin kaaren kapasiteetti on U. Periaatteessa virtausta voi lähettää korkeintaan Um yksikköä. Lyhimmän reitin etsiminen ei ole suoraviivaista koska verkossa on negatiivisia kaaria. Lyhimmän reitin verkossa jossa on negatiivisia kaaria voi etsiä Bellman-Ford-algoritmilla, jonka aikavaativuus on $O(nm)$. Kokonaisaikavaativuudeksi tulee $O(Um^2n)$. Käytännössä käytetään SPFA-algoritmia, joka on käytännössä hyvin nopeasti toimiva versio Bellman-Fordista.

Dijkstran algoritmin käyttö

Dijkstran algoritmia ei voi käyttää ongelmaan suoraan koska verkossa on kaaria joiden hinta on negatiivinen. Kuitenkin aiemmin esitellyn potentiaalifunktion avulla voidaan muuttaa kaikkien kaarien hinnat epänegatiivisiksi. Vaikka jäännösverkko muuttuu, potentiaalifunktion voi päivittää helposti päteväksi käyttämällä lyhimpiä polkuja jotka kuitenkin lasketaan kun lähetetään virtausta. Jos jäännösverkossa olevan $s \rightsquigarrow t$-polun pituus on L, niin sen pituus vähennetyissä hinnoissa on $L + p(s) - p(t)$ koska kaikki muut potentiaalit kumoavat toisensa. Seuraa että lyhimmän $s \rightsquigarrow t$ polun voi yhtä hyvin hakea vähennettyjen hintojen verkossa.
Saadaan nopeampi algoritmi: Etsitään ensin joku potentiaalifunktio Bellman-Ford-algoritmilla. Sen jälkeen käytetään Dijkstran algoritmia vähennettyjen hintojen verkossa augmentoivien polkujen löytämiseen ja potentiaalifunktion päivittämiseen. Algoritmin aikavaativuus on $O(nm + Um^2 \log n)$.

Mincost-circulation

Käytetään nimeä MINCOST-CIRCULATION yleisemmästä versiosta ongelmalle jossa pitää laskea minimihintainen virtaus. Mincost-circulation-ongelmassa sallitaan että verkossa on syklejä joiden hinta on negatiivinen. Tämä tekee lähtö- ja kohdesolmujen ja virtauksen määrän määrittämisestä turhaa, koska jos halutaan laskea aiemmin määritelty Mincost-flow käyttäen Mincost-circulationia, voidaan lisätä kyseiseen verkkoon kaari kohdesolmusta lähtösolmuun jonka kapasiteetti on k ja hinta on itseisarvoltaan tarpeeksi suuri negatiivinen luku. Määritellään Mincost-circulation-ongelma: Olkoon $G = (V, E, U, C)$ virtausverkko, jossa $U(e) \in \mathbb{Z}_+$ on kaaren e kapasiteetti ja $C(e) \in \mathbb{R}$ on kaaren e hinta. Tavoitteena on löytää minimihintainen virtaus f jota merkitään funktiona $f : E \to \mathbb{Z}$ joka:

Minimoi

$$\sum_{uv \in E} C(uv) f(uv)$$

Toteuttaa ehdot:
Säilyvyysehto

$$\sum_{v \in V} f(uv) = \sum_{v \in V} f(vu) \text{ kaikilla } u \in V$$

Kapasiteettiehto

$$0 \leq f(uv) \leq U(uv) \text{ kaikilla } uv \in E$$

Huomaa että optimaalinen hinta ei ole koskaan positiivinen koska $f(uv) = 0$ kaikilla uv toteuttaa ehdot aina ja sen hinta on 0.

Kapasiteettiskaalaava algoritmi Mincost-circulation-ongelmalle

Aiemmin esitellyt algoritmit mincost-flow-ongelmalle eivät toimi polynomisessa ajassa, sillä niiden aikavaativuuksissa on kerroin U, joka voi olla eksponentiaalinen suhteessa syötteen pituuteen. Nyt esitetään polynomisessa ajassa toimiva algoritmi Mincost-circulation-ongelmalle, jonka aikavaativuudessa on kerroin $\log U$ kertoimen U sijasta. Algoritmissa tullaan käyttämään samaa määritelmää jäännösverkolle, potentiaalifunktiolle ja vähennettyjen hintojen verkolle kuin aiemmin esitellyssä SAP-algoritmissa.

Skaalaus

Skaalaavat algoritmit ratkaisevat ongelman ottamalla ensin huomioon vain syötteen eniten merkitsevät bitit ja lisäämällä ratkaisun tarkkuutta bitti kerrallaan. Käytetään tätä tekniikkaa Mincost-circulation-ongelmaan. Oletetaan että $U(uv) < 2^k$ kaikilla $uv \in E$. Skaalaava algoritmi käyttää $k + 1$ vaihetta jotka numeroidaan $0 \dots k$. Vaiheessa i verkko on muuten sama kuin alkuperäinen verkko, mutta kapasiteetit ovat $U_i(uv) = \lfloor \frac{U(uv)}{2^{k-i}} \rfloor$. Jokaisessa vaiheessa ratkaistaan ongelma optimaalisesti sen vaiheen kapasiteeteille joten vaiheen k ratkaisu on ongleman ratkaisu. Toisin sanoen vaiheessa i ratkaistaan ongelma niin että katsotaan kapasiteettien binääriesityksien i:tä eniten merkitsevää bittiä. Vaiheessa 0 kaikki kapasiteetit ovat 0 joten ratkaisu on triviaali. Ongelmaksi jää siirtyminen vaiheesta i vaiheeseen $i + 1$. Sanotaan että siirtyminen tehdään kahdessa vaiheessa, skaalausvaiheessa ja augmentointivaiheessa.

Skaalausvaihe

Huomataan että $U_{i+1}(uv) = U_i(uv) \cdot 2$ tai $U_{i+1}(uv) = U_i(uv) \cdot 2 + 1$. Voidaan siis ensin kertoa kaikkien kaarien kapasiteetit ja niissä kulkeva virtaus kahdella. Virtaus toteuttaa edelleen kaikki ehdot ja on optimaalinen.

Augmentointivaihe

Skaalausvaiheen jälkeen oltaisiin valmiita jos $U_{i+1}(uv) = U_i(uv) \cdot 2$ toteutuisi kaikilla kaarilla. Kuitenkin kaarilla joilla $U_{i+1}(uv) = U_i(uv) \cdot 2 + 1$ pitää lisätä kapasiteettia vielä yhdellä. Kun kasvatetaan tällaisten kaarien kapasiteettia yhdellä yksi kaari kerrallaan ongelma palautuu ongelmaan: Annetaan virtausverkko ja minimihintainen virtaus siellä. Löydä minimihintainen virtaus kun yhden kaaren kapasiteettia kasvatetaan yhdellä.

Augmentointi

Muistetaan että virtaus on optimaalinen jos sen jäännösverkossa ei ole negatiivista sykliä. Tästä seuraa että kaaren kapasiteetin kasvattamisen jälkeen virtaus on edelleen optimaalinen jos se ei luo uutta negatiivista sykliä jäännösverkkoon. Toisaalta jos kapasiteetin kasvattaminen luo negatiivisen syklin jäännösverkkoon niin sitä sykliä pitkin voidaan augmentoida virtausta ja saadaan pienempi hinta. Siis kapasiteetin kasvattaminen muuttaa optimaalista virtausta jos ja vain jos se luo negatiivisen syklin jäännösverkkoon.

Otetaan jälleen avuksi potentiaalifunktio. Pidetään pätevää potentiaalifunktiota yllä koko algoritmin ajan. Aluksi potentiaalifunktio voi olla $p(u) = 0$ kaikilla u. Skaalausvaiheessa potentiaalifunktiota ei tarvitse muuttaa koska skaalaus ei luo uusia kaaria jäännösverkkoon eivätkä kaarien hinnat muutu.

Tapaukset

Tarkastellaan tapauksia joita voi käydä kun kaaren uv kapasitettia kasvatetaan yhdellä.

1. Kaari toteuttaa vanhan potentiaalifunktion

Jos kaari toteuttaa potentiaalifunktion niin negatiivisia syklejä ei ole eikä potentiaalifunktiota tarvitse päivittää. Huomaa että tämä tapaus tapahtuu aina jos kaari uv on jo jäännösverkossa eli $U_f(uv) > 0$ ennen kapasiteetin lisäämistä.

2. Kaari rikkoo vanhaa potentiaalifunktiota

Tässä tapauksessa potentiaalifunktiota täytyy päivittää ja negatiivisia syklejä voi syntyä. Ennen kuin kasvatetaan kaaren uv kapasiteettia lasketaan lyhin reitti solmusta v kaikkiin solmuihin vähennettyjen hintojen verkossa. Sanotaan että lyhimmän reitin pituus solmusta v solmuun w vähennettyjen hintojen verkossa on $sp(w)$. Jos solmusta v ei ole reittiä solmuun w niin $sp(w)$ on joku riittävän suuri vakio.

2.1. Negatiivista sykliä ei syntynyt

Jos syntyy negatiivinen sykli, niin se kulkee ensin uutta kaarta uv pitkin ja sitten lyhintä reittiä solmusta v solmuun u. Tarkistamalla lyhimmän reitin pituuden solmusta v solmuun u voidaan selvittää syntyikö negatiivista sykliä. Toisin sanoen negatiivista sykliä ei syntynyt jos $sp(u) + p(u) + C(uv) - p(v) \geq 0$. Potentiaalifunktio pitää kuitenkin korjata vaikka negatiivista sykliä ei syntynyt. Asetetaan uudeksi potentiaalifunktioksi $p'(w) = p(w) + sp(w)$. Tämä toteuttaa potentiaalifunktion ehdot kaikilla jäännösverkossa jo olevilla kaarilla wz koska $sp(w) + C_{fr}(wz) \geq sp(z)$ koska sp on lyhin-reitti-funktio joten $p(w) + sp(w) + C_f(wz) \geq p(z) + sp(z)$ joten $p'(w) + C_f(wz) \geq p'(z)$. (Huomaa että tämä pätee myös jos solmusta v ei ole reittiä solmuun w koska vähennetyt hinnat ovat epänegatiivisia.) Uusi potentiaalifunktio toteuttaa potentiaalifunktion ehdot myös uudella kaarella uv koska $sp(u) + p(u) + C(uv) - p(v) \geq 0$ joten $p(u) + sp(u) + C_f(uv) \geq p(v) + sp(v)$ koska $sp(v) = 0$ ja $C_f(uv) = C(uv)$ joten $p'(u) + C_f(uv) \geq p'(v)$.

2.2. Negatiivinen sykli syntyi

Negatiivinen sykli syntyy jos $sp(u) + p(u) + C(uv) - p(v) < 0$. Tässä tapauksessa virtausta lähetetään yksi yksikkö sykliä joka koostuu kaaresta uv ja lyhimmästä $v \rightsquigarrow u$-polusta pitkin. Kun virtausta on lähetetty tätä sykliä pitkin, verkossa ei ole enää negatiivisia syklejä. Todistetaan se näyttämällä pätevä potentiaalifunktio. Jälleen kerran uusi potentiaalifunktio on $p'(w) = p(w) + sp(w)$. Edellisen kohdan analyysin perusteella tämä potentiaalifunktio on pätevä kaikille jäännösverkossa jo olleille kaarille. Jäännösverkkoon kuitenkin saattoi ilmestyä uusia kaaria jotka menevät vastakkaiseen suuntaan lyhintä $v \rightsquigarrow u$-polkua. Otetaan lyhimmällä $v \rightsquigarrow u$-polulla peräkkäin olevat solmut w ja z. Nyt $sp(z) = sp(w) + C_{fr}(wz)$ joten $sp(z) = sp(w) + p(w) + C_f(wz) - p(z)$ joten $p(z) + sp(z) = p(w) + sp(w) + C_f(wz)$ joten $p'(z) = p'(w) + C_f(wz)$ joten $p'(z) + C_f(zw) = p'(w)$ eli potentiaalifunktio pätee kaarilla jotka menevät vastakkaiseen suuntaan lyhintä $v \rightsquigarrow u$-polkua. Lisäksi jäännösverkkoon tulee kaari vu. $sp(u)+p(u)+C(uv)-p(v) <$

0 joten $sp(u)+p(u)+C(uv) < p(v)$ joten $p(v)+C(vu) > p(u)+sp(u)$ joten $p'(v)+C(vu) > p'(u)$ koska $sp(v) = 0$. Huomaa että jäännösverkkoon ei tule kaarta uv.

Nyt kaikkiin yhden kaaren augmentoinnissa tapahtuviin tapauksiin on esitetty ja analysoitu algoritmi joten augmentointivaihe on valmis. Täten koko algoritmi on valmis, koska siirtymä vaiheesta i vaiheeseen $i+1$ on esitetty kokonaan ja on todettu että vaiheen 0 ratkaisu on triviaali.

Aikavaativuus

Algoritmissa on $\log_2 U$ vaihetta. Jokaisessa vaiheessa augmentointi dominoi aikavaativuutta. Jokaisessa vaiheessa augmentoidaan korkeintaan m kaarta. Kaaren augmentoinnissa haetaan lyhimmät reitit vähennettyjen hintojen verkossa jossa siihen voi käyttää Dijkstran algoritmia, eli lyhimpien reittien haun aikavaativuudeksi tulee $O(m \log n)$. Yhteensä aikavaativuudeksi tulee $O(\log U m^2 \log n)$, joka on polynominen aikavaativuus ja teoriassa parempi kuin aiemmin käsitellyt algoritmit mincost-flow-ongelmaan. Vertailun vuoksi maksimivirtauksen laskemiseen käytetyn Edmonds-Karp algoritmin aikavaativuus on $O(m^2 n)$.

Lisähuomio potentiaaleista

Monessa kohdassa solmujen potentiaaleihin lisätään jotain lukuja jotka riippuvat potentiaaleista ja joiden suuruutta algoritmin aikana ei ole analysoitu. Potentiaalien kasvaminen räjähdysmäisesti osoittautuu myös käytännön ongelmaksi. Potentiaalifunktion kuitenkin pystyy korjaamaan $O(m \log n)$ ajassa sellaiseksi että kaikki potentiaalit ovat rajoitettuja suhteessa syötteen kokoon. Korjaaminen tapahtuu seuraavalla tavalla: Otetaan joku solmu v jonka potentiaalia ei ole vielä korjattu. Lasketaan siitä Dijkstran ja potentiaalifunktion avulla lyhimmät polut solmuihin joihin pääsee siitä ja joita ei ole vielä käsitelty. Asetetaan näiden solmujen potentiaaliksi lyhin reitti solmusta v + suurin korjattu potentiaali ennen tätä vaihetta. Nyt saadaan pätevä potentiaalifunktio, jonka itseisarvo on kaikissa solmuissa korkeintaan nC, jossa C on suurin hintojen itseisarvo. Korjaaminen voidaan tehdä vaikka jokaisen potentiaalien päivityksen jälkeen eikä se vaikuta aikavaativuuteen koska Dijkstra ajetaan kuitenkin aina kun potentiaaleja päivitetään. Kokeilin monia tapoja korjata potentiaaleja implementaatiossa, ja tämä oli ainoa jonka sain toimimaan.