Mincost-flow-algoritmit

Tuukka Korhonen

October 24, 2016

Mincost-flow

Annetaan virtausverkko $G = (V, E, U, C, s, t)$, jossa $U(e) \in \mathbb{Z}_+$ on kaaren e kapasiteetti, $C(e) \in \mathbb{R}$ on kaaren e hinta ja s ja t ovat lähde- ja viemärisolmut. Annetaan lisäksi kokonaisluku k 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 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 lähteestä s viemäriin 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 niin saadaan yksinkertaisempia toimivia algoritmeja ja tämä määritelmä sopii ongelman luonteeseen jossa lähetetään tavaraa lähteestä viemäriin. Jos negatiivisia syklejä saisi olla, niin optimaalinen ratkaisu voisi pyörittää virtausta jossain muualla kuin reiteillä lähteestä viemäriin. Myöhemmin esitetään MINCOST CIRCULATION ongelma, jossa negatiiviset syklit ovat sallittuja.

Shortest-Augmenting-Path-algoritmi

Shortest-Augmenting-Path-algoritmi (lyhyemmin SAP-algoritmi) etsii mincost-flown lähettämällä virtausta lyhyintä reittiä kaarien hintojen suhteen lähteestä viemäriin. Tällaista lyhyintä reittiä kutsutaan *augmentoivaksi poluksi.* Algoritmi on muunnelma Ford-Fulkerson-algoritmista maksimivirtauksen etsimiseen. Intuitiota saa miettimällä miten mincost-flow etsittäisiin jos virtausta lähetettäisiin vain yksi yksikkö: se olisi lyhyin reitti lähdesolmusta viemärisolmuun.

Jälkiverkko

Sanotaan että G_f on virtauksen f jälkiverkko. Määritellään kapasiteetit jälkiverkossa:

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

Määritellään hinnat jälkiverkossa:

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

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

Nyt jos jälkiverkossa 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 s:stä t:hen. Toinen sallittu tapa muuttaa virtausta on lähettää virtausta jotain jälkiverkon sykliä pitkin, maksimissaam syklin pienimmän kapasiteetin verran. Laskemalla voidaan tarkistaa että nämä muutokset säilyttävät virtauksen säilyvyys- ja kapasiteettiehdon.

Virtauksen optimaalisuus

Lemma 1: k-virtaus on optimaalinen jos jälkiverkossa ei ole sykliä jonka hinta on negatiivinen.

Todistus 1: Olkoon f joku k-virtaus niin että G_f:ssä ei ole negatiivista sykliä. Olkoon f' joku k-virtaus 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 lähde- ja viemärisolmut, 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, niin jäljelle jää virtaus joka toteuttaa säilyvyysehdon. Siispä kun $f' - f$ hajotetaan sykleiksi, niin ainakin yhden syklin hinta on negatiivinen. Sitä sykliä pitkin voidaan lähettää virtausta G_f:ssä, joten G_f:ssä on negatiivinen sykli.

Lemma 2: Jos jälkiverkossa ei ole negatiivista sykliä, niin lähettämällä virtausta halvinta $s \rightsquigarrow t$ polkua pitkin sinne ei synny negatiivista sykliä.

Todistus 2: Sanotaan että solmujen potentiaalifunktio on funktio $p : V \to \mathbb{R}$, joka toteuttaa $p(u) + C_f(uv) \geq p(v)$ kaikilla jälkiverkon kaarilla uv. Jos G_f:ssä ei ole negatiivista sykliä niin tällainen funktio on olemassa, sillä esim. lyhyin reitti jostain solmusta kelpaa täksi funktioksi. Nyt määritellään kaaren uv vähennetty hinta $C_{fr}(uv) = p(u) + C_f(uv) - p(v)$. Nähdään että syklin hinta vähennetyissä hinnoissa on sama kuin syklin hinta. Nähdään myös että kaikki vähennetyt hinnat ovat epänegatiivisia, josta on helppo nähdä että jos on olemassa validi potentiaalifunktio, niin negatiivisia syklejä ei ole.
Kun lähetetään virtausta lyhyintä $s \rightsquigarrow t$ polkua pitkin, voidaan potentiaalifunktiota päivittää seuraavalla tavalla: Olkoon $sp(u)$ lyhyimmän reitin pituus solmusta s solmuun u kun tarkastellaan reitin pituutta 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 verkossa koska $sp(u) + C_{fr}(uv) \geq sp(v)$ koska sp on lyhyin 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 $p'(u) + C_f(uv) \geq p'(v)$. Lisäksi verkkoon saattoi ilmestyä uusia kaaria kun lyhyintä $s \rightsquigarrow t$ polkua päinvastaiseen suuntaan menevien kaarien kapasiteetti kasvoi. Kun u ja v ovat lyhyimmällä polulla peräkkäin olevat solmut, niin $sp(u) + C_{fr}(uv) = sp(v)$ joten $p'(u) + C_f(uv) = p'(v)$ joten $p'(v) - C_f(uv) = p'(u)$ eli potentiaalifunktio toimii kaikilla uusilla kaarilla jotka ilmestyivät jälkiverkkoon.

SAP-algoritmi siis lähettää virtausta jälkiverkon lyhintä $s \rightsquigarrow t$ polkua pitkin niin kauan kunnes löytää k-virtauksen tai maksimivirtauksen. Yllä todistettiin että algoritmi pitää joka vaiheessa yllä optimaalista virtausta.

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. Periaatteessa virtausta voi lähettää korkeintaan Um yksikköä, jossa U on verkon suurin kapasiteetti. Lyhyimmän reitin etsiminen ei ole ihan suoraviivaista koska verkossa on negatiivisa kaaria. Lyhyimmän 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.

Todistuksessa 2 käytettiin solmujen potentiaalifunktiota. Nähtiin miten potentiaalifunktion avulla saadaan muutettua kaikkien kaarien pituudet epänegatiivisiksi ja miten potentiaalifunktiota voi päivittää helposti lyhyimpien reittien perusteella. Minkä tahansa $s \rightsquigarrow t$ polun pituus vähennetyissä hinnoissa on polun pituus $+p(s) - p(t)$, eli lyhyimmän $s \rightsquigarrow t$ polun voi yhtä hyvin hakea vähennettyjen hintojen verkossa. Saadaan algoritmi: etsitään ensin joku potentiaalifunktio Bellman-Ford-algoritmilla. Sen jälkeen käytetään Dijkstraa vähennettyjen hintojen verkossa augmentoivien polkujen löytämiseen ja päivitetään potentiaalifunktiota. Algoritmin aikavaativuss on $O(nm + Um^2 logn)$.