Algoritmuselmélet és bonyolultságelmélet MSc hallgatók számára

Folyam algoritmusok

2017. Előadó: Hajnal Péter

1. Ford—Fulkerson-algoritmus

A folyamok alaptételéből adódik a következő séma:
`Ford--Fulkerson-algoritmus`

(I) **Inicializálás:** Vegyünk egy kiinduló f_0 folyamot, például a $f_0 \equiv 0$ folyamot (minden élen 0 anyagmennyiség folyik).

(K) **Keresés:** Keressünk javító utat f-re. Ha nincs, akkor (S), ha találunk, akkor (J).

(J) **Javítás:** A múlt heti lemma alapján „javítsuk" f-et. $f \leftarrow \widetilde{f}$. Folytassuk a (K) lépénél.

(S) **Stop:** Álljunk le, az aktuális folyam optimális (maximális értékű).

A Keresés lépés az egyetlen algoritmikusan problémás lépés. Ez sem okoz nehézséget. A következő fogalom természetesen adódik:

Definíció. Legyen $\mathcal{H}$ egy hálózat és benne egy f folyam. Definiálunk egy maradék (reziduális) $\mathcal{R}$ hálózatot. $\mathcal{R}$ forrás és nyelője ugyanaz mint $\mathcal{H}$-ban. A $\mathcal{R}$ gráfja legyen $\overrightarrow{G}_r$, míg $\overrightarrow{G}$ jelölje $\mathcal{H}$ gráfját. $\overrightarrow{G}_r$ éleit és kapacitásait a kövtekezőképpen kapjuk: Minden olyan $e = \overrightarrow{uv} \in E(\overrightarrow{G})$ élre a következőt végezzük el:

(i) Ha $0 < f(e) < c(e)$, akkor felveszünk egy $e_r^+ = \overrightarrow{uv}$ élet $c(e) - f(e)$ kapacitással, továbbá felveszünk egy $e_r^- = \overrightarrow{vu}$ élet $f(e)$ kapacitással.

(ii) Ha $0 = f(e) < c(e)$, akkor felveszünk egy $e_r^+ = \overrightarrow{uv}$ élet $c(e) - f(e)$ kapacitással.

(iii) Ha $0 < f(e) = c(e)$, akkor felveszünk egy $e_r^- = \overrightarrow{vu}$ élet $f(e)$ kapacitással.

Észrevétel. Az f folyam $\overrightarrow{G}$-beli javító utai és a $\overrightarrow{G}_r$-beli irányított st utak között egy nyilvánvaló bijekció van.

Valóban, egy $\overrightarrow{G}$-beli J javító út esetén az $e \in E_{elre}(J)$ élekre az e_r^+ éleket, míg a $e \in E_{htra}(J)$ élekre az e_r^- éleket véve egy irányított utat kapunk $\overrightarrow{G}_r$-ben. Fordítva: Legyen $\overrightarrow{P}$ egy irányított st út $\overrightarrow{G}_r$-ben. Ekkor minden e_r^ε élére véve ($e \in E(\overrightarrow{G})$, $\varepsilon \in \{+, -\}$) élre véve az e „ős élt" egy javító út élhalmazát kapjuk.

Azaz a keresés $\overrightarrow{G}_r$ konstrukciójával és ebben egy irányított út keresésével megoldható. Az átalakított probléma „mindkét komponense" egyszerű.

1. Tétel. *Legyen $\mathcal{H}$ egy hálózat, amelyre $c, f_0 : E(\overrightarrow{G}) \to \mathbb{Q}_+$. Ekkor a Ford—Fulkerson-algoritmus véges lépésben megtalálja az optimális folyamot.*

Bizonyítás. c skálázása (skalárral való szorzása) a problémát és az algoritmus futását nem befolyásolja. Így feltehetjük, hogy $c, f_0 : E(\overrightarrow{G}) \to \mathbb{Z}_+$.

Az algoritmust futtatva az aritmetika sosem lép ki az egészek köréből. (Bele kell gondolnunk: a kezdeti számokból az eljárásunk csupán, különbségeket, összegeket képez, minimumot vesz.) Így speciálisan minden javítás is legalább 1-gyel növeli a folyamértéket. Ez garantálja az algoritmus leállását. ■

Érdemes megfogalmazni a tétel egyszerű következményét is.

2. Tétel. *Legyen $\mathcal{H}$ egy hálózat, amelyre $c : E(\overrightarrow{G}) \to \mathbb{Z}_{++}$. Ekkor $\mathcal{H}$-ban van egy $f_{opt} : E(\overrightarrow{G}) \to \mathbb{Z}_+$ optimális folyam.*

A tétel csak egy olyan optimális folyam létezését állítja, melyben minden élen egész anyagmennyiség folyik. Elképzelhető, hogy vannak olyan maximális értékű folyamok, amikben egyes éleken nem egész anyagmennyiség folyik.

⋆

Azt is megjegyezzük, hogy a tétel nem kielégítő:

Nem mond semmit arról hány javítás szükséges, azon túl hogy a javítások száma legfeljebb az optimális folyam értéke. Ez azonban hatalmas szám lehet. Ha a folyam kapacitásai k számjegyű számokkal adott, akkor az optimális folyam értéke lehet exponenciális k-ban.

Arról se mond semmit az algoritmus mi történik, ha elméletben futtatjuk az algoritmust pontos valós aritmetikával. A bizonyítás ekkor nem működik, a véges leállás sem látható. Ez valóságos lehetőség. Előfordulhat, hogy mindig „rossz" javító utakat találunk meg, egyre kisebb növeléseket érünk el és a növelt folyamok értékei nem is konvergálnak az optimális értékhez.

Ennek ellenére a Ford—Fulkerson-algoritmus a legtöbb folyam algoritmus keretét adja.

2. Edmonds—Karp-algoritmus

Edmonds és Karp hozzájárulása az volt, hogy az irányított st út keresését pontosították
`Ford--Fulkerson-algoritmus`

(I) **Inicializálás:** Vegyünk egy kiinduló f_0 folyamot, például a $f_0 \equiv 0$ folyamot (minden élen 0 anyagmennyiség folyik).

(R) **A reziduális gráf:** Építsük fel az $\overrightarrow{G}_r$ reziduális gráfot.

(K) **Keresés:** Szélességi kereséssel keressünk irányított utat $\overrightarrow{G}_r$-ben. Ha nincs út, akkor (S), ha találunk, akkor keressük meg a benne szereplő élek őseit és hajtsuk végre az így kapott ősélek által alkotott javító úttal (J)-t.

(J) **Javítás:** A múlt heti lemma alapján „javítsuk" f-et. $f \leftarrow \widetilde{f}$. Folytassuk a (K) lépénél.

(S) **Stop:** Álljunk le, az aktuális folyam optimális (maximális értékű).

Megjegyzés. A szélességi keresés egy fontos tulajdonsága ha találunk utat, akkor az egyik legrövidebb st utat találja meg. Azaz az Edmonds—Karp-algoritmus mindig a legrövidebb javító út mentén javít.

Az algoritmus lényege nem az, hogy a javító út keresést pontosan leírta. Elméleti jelentősége van. Belátjuk, hogy (akár pontos valós aritmetikát végezve is) az algoritmus véges sok javítás után leáll. Sőt a javítások száma becsülhető az alapgráfunk méretének polinomiális függvényében. Ezt nem tudtuk elvégezni a Ford—Fulkerson algoritmus esetén.

Az Edmonds—Karp-algoritmus futását fázisokra osztjuk. Analízisünk a fázisokra bontott algoritmus struktúráját használja.

Definíció. A futás első javítása megnyitja az első fázist. Minden további javitás esetén megnézzük, hogy megtalált legrövidebb javító út hossza megegyezik-e az előző javítás során talált legrövidebb javító út hosszával. Ha igen, akkor a fázist folytatjuk. Ha a hosszban változás áll be, akkor az aktuális fázis az előző javítással véget ért, az új javítás egy új fázist kezd.

Az analízishez szükségünk lesz még egy fogalomra.

Definíció. $\overrightarrow{G}_r$-ben L_i alkossa azon csúcsok halmazát, amelyek s-ből elérhetők i hosszú irányított úttal, de rövidebbel nem. Speciálisan $L_0 = \{s\}$.

$\overrightarrow{G}_r^0$ legyen az a gráf, amely csúcsai $L_0 \cup L_1 \cup L_2 \cup \ldots$ és élei az s-ből induló legrövidebb utak élei. Speciálisan $\overrightarrow{G}_r^0$ összes éle valamelyik L_i-ből indul és L_{i+1}-be vezet.

Megjegyezzük, hogy ezt a gráfot a szélességi keresés „kiszámolja”. Igazából ennek a gráfnak a kiszámolása a fontos, ha ez adott, akkor a legrövidebb $\overrightarrow{st}$ út $\mathcal{O}(|V|)$ lépésben adódik: t-ből indulve folyamatosan keresünk egy ebbe vezető élt, amin visszalépünk. Ezt ismételgetve s-be jutunk, megtalal'juk a keresett utat. Megjegyezzük, hogy a teljes sz'elességi keresés költsége $\mathcal{O}(|E| + |V|)$.

Most már kimondhatjuk Edmonds és Karp tételét.

3. Tétel. *(i) Az egymást követő fázisok folyamán a legrövidebb javító út hossza nő.*

(ii) Egy fázison belül $\overrightarrow{G}_r^0$ élhalmaza csökken.

Először kimondunk egy nyilvánvaló következményt.

4. Következmény. *Az Emonds—Karp-algoritmus futása legfeljebb $|V|$ fázisból áll. Minden fázis legfeljebb $|E|$ javítást tartalmaz. Azaz az algoritmus legfeljebb $|V| \cdot |E|$ javítás után megtalál egy optimális algoritmust.*

Speciálisan az Edmonds—Karp-algoritmus futási ideje $\mathcal{O}(|V||E|^2)$.

A következmény nyilvánvaló hiszen $|V| \cdot |E|$-szer kell felépíteni $\overrightarrow{G}_r$-t és benne egy szélességi keresést végezni. Ez utóbbi feladat-pár $\mathcal{O}(|V| + |E|) = \mathcal{O}(|E|)$ lépésben elvégezhető (feltesszük, hogy alaphálózatunk összefüggő).

A tétel bizonyítása: (i): Egy javítás alatt találunk egy javító utat. Ez egy irányított út lesz $\overrightarrow{G}_r^0$-ban. Kiszámoljuk δ-t és az út minden élén megváltoztatjuk az ott folyó anyagmennyiséget. δ-t úgy választottuk meg, hogy valamelyik élen a

folyó anyagmennyiség eléri a kapacitást, vagy lenullázódik. Az ilyen él(ek) az új $\overrightarrow{G_r}$ reziduális gráfhoz való hozzájárulása biztos csökken. Más változás még az lehet, hogy egy a javító úton szerepelt él (amely $\overrightarrow{G_r^0}$-hez hozzájárult) mellé ellentétes irányban megjelenik egy új él.

Nézzük az eredeti $\overrightarrow{G_r}$ gráf $L_0, L_1, \ldots$ halmazait. A változás él eltűnés és L_i-ből L_{i+1}-be vezető élek mellé egy ellenirányú él megjelenése. Ez a legrövidebb út hosszát nem növelheti. Így (i) adódik.

(ii): A reziduális gráfban megjelenő ellentétes él a fentiek miatt egy L_{i+1}-ből L_i-be vezető él lehet csak, ahol $i+1 \leq \ell$ (ahol $t \in L_\ell$). Ez garantálja, hogy az új $\overrightarrow{G_r^0}$ gráf nem tartalmazhat más élt mint a korábbi. Sőt az élfogyás is szükségszerű δ választása miatt. ■

3. Dinic-algoritmus

Láttuk, hogy a szélességi keresés kiszámol egy $\overrightarrow{G_r^0}$ gráfot. Ebben ott van a legrövidebb st út, sőt ez $\mathcal{O}(|V|)$ időben ki is számolható. Igazából az út már a reziduális gráf egy kisebb részében is ott van:

Definíció. Legyen $\overrightarrow{G_r^*}$ gráf, amelyet azon élek adnak ki, amelyek egy legrövidebb $\overrightarrow{st}$ úton szerepelnek.

Ezt is könnyen kiszámolhatjuk, ha egy szélességi kereséssel meghatározzuk a $\overrightarrow{G_r^*}$ gráfot. Majd ebben t-ből indulva visszafelé végzünk egy szélességi keresést.

Ebben a gráfban a legrövidebb $\overrightarrow{st}$ út még egyszerűbben kiszámolható: s-ből elindulva sétálunk előre. Szükségszerűen t-be jutunk egy legrövidebb úton. Ennek költsége $\mathcal{O}(|V|)$.

Dinic ötlete a következő: Tartsuk meg Edmonds—Karp algoritmusának alapstruktúráját. Egy fázison belül azonban ne építsük fel mindent javítás után a $\overrightarrow{G_r^0}$ gráfot a semmiből. A fázis elején számoljuk ki $\overrightarrow{G_r^*}$ gráfot. A fázis során ez a gráf csökken. A csökkenést tartsuk nyilván.

5. Tétel (Dinic). *Egy fázison belül a $\overrightarrow{G_r^*}$ gráf változása $\mathcal{O}(|E|)$ költséggel nyilvántartható.*

Ezt a tételt az amortizációs analízis témakörben fogjuk tárgyalni.

Előnye nyilvánvaló: A $|V|\cdot|E|$ javító út keresése az aktuális $\overrightarrow{G_r^*}$ gráfokban $|V||E|\cdot\mathcal{O}(|V|)$ lépéssel megoldható. A $\overrightarrow{G_r^*}$ gráf minden fázis elején $\mathcal{O}(|E|)$ lépéssel felépíthető és az egész fázis alatt $\mathcal{O}(|E|)$ lépéssel update-elhető. A teljes folyam algoritmus költsége:

$$|V||E|\cdot\mathcal{O}(|V|) + |V|\cdot\mathcal{O}(|E|) = \mathcal{O}(|V|^2|E|).$$

6. Tétel. *Adott hálózatban a Dinic-algoritmus $\mathcal{O}(|V|^2|E|)$ lépésben megtalálja az optimális folyamot.*