1. CVIČENÍ Z ÚVODU DO APROXIMACÍ

poznáváme aproximace, vrcholové pokrytí

D(aproximační algoritmus**):** Máme-li polynomiální algoritmus pro minimalizační problém, tak o něm řekneme, že je to *c-aproximační algoritmus*, pokud pro každou instanci problému I platí, že hodnota výstupu našeho algoritmu $ALG(I)$ je nejvýš c-krát větší než hodnota optima instance $OPT(I)$, čili že $ALG(I) \leq c \cdot OPT(I)$.

Pro maximalizační problémy platí rovnost trochu jiná: tam musí platit $\forall I\colon ALG(I) \geq c \cdot OPT(I)$ a rozsah c bude jiný. To znamená, že občas budeme mluvit o $\frac{1}{2}$-aproximačním algoritmu, a občas o 2-aproximačním algoritmu.

Většinu času se budeme snažit, aby c byla konstanta, ale nepůjde to vždy. Proto můžeme aproximační poměr c vyjádřit i jako funkci velikosti vstupu a tvořit $\frac{1}{n}$-aproximační nebo $O(\log n)$-aproximační algoritmy.

Některé algoritmy jsou snadné a konstantně aproximační:

Příklad první Uvažme problém Maximální rovinný podgraf. Na vstupu dostaneme souvislý nerovinný graf G a naším cílem je najít jeho podgraf $R \subseteq G$, který už rovinný je – a maximalizujeme počet zbylých hran v grafu R.

Navrhněte 1/3-aproximační algoritmus pro tento problém.

Jiné snadné algoritmy zas tak dobré nejsou:

Příklad druhý Mějme následující aproximační algoritmus pro problém Maximální klika v grafu:

„Pro každou hranu $e \in E(G)$ hladově vyzkoušíme, jestli k ní existuje třetí vrchol, se kterým tvoří trojúhelník. Pokud ano, hledáme a vybereme libovolný čtvrtý vrchol, který s těmito třemi tvoří K_4. Pokračujeme, dokud nenajdeme co největší kliku, pak si velikost zapamatujeme a zkusíme hledat pro další hranu $e' \in E(G)$. Na výstupu vrátíme tu největší z nalezených klik."

Jaký je jeho aproximační poměr?

A u některých problémů lze dokázat, že dosáhnout dobrého poměru je těžké:

Příklad třetí Mějme klasický optimalizační problém Pakování do košů. Na vstupu tohoto problému dostaneme n objektů s velikostí mezi $[0,1]$, a naším úkolem je napakovat objekty do co nejméně košů, kde každý koš má nosnost právě 1. Minimalizujeme počet použitých neprázdných košů.

Dokažte, že kdyby pro tento problém existoval 1.499-aproximační algoritmus, tak už $P = NP$.

Hinty:

- Chceme dokazovat těžkost, takže náš argument musí být v kostce takovýto: „Vezměme si všechny vstupní instance nějakého NP-úplného problému, převedeme je na problém Pakování do košů a pokud je onen 1.499-aproximační algoritmus zapakuje, tak převedeme řešení na řešení původního NP-úplného problému."
- Zkuste využít toho, že pokud by optimum bylo k košů, tak 1.499-aproximační algoritmus musí použít nejvýše $\lfloor 1.499k \rfloor$ košů.

Zbytek cvičení se budeme zabývat problémem Minimálního vrcholového pokrytí. Vrcholové pokrytí je množina vrcholů taková, že každá hrana grafu je pokrytá, čili alespoň jeden její konec je v pokrytí. Hledáme vrcholové pokrytí minimální co do počtu vrcholů, občas bývá i vážené.

Příklad čtvrtý Nejprve zanalyzujeme hladový algoritmus:

„Setřídíme si vrcholy podle stupně, vezměme ten (některý) s největším stupněm a přidáme ho do vrcholového pokrytí. Tím pokryjeme všechny hrany sousedící s tímto vrcholem, tak je z grafu odstraníme. Setřídíme vrcholy podle stupně znovu a pokračujeme stejným postupem.“

- Na začátek vyvraťte, že tento algoritmus je 1.999-aproximační,
- … nebo rovnou vyvraťte, že tento algoritmus je 2-aproximační,
- … a nebo rovnou vyvraťte, že tento algoritmus je c-aproximační pro jakékoli $c \geq 1$.

Příklad pátý Nyní zvažme tento jiný, ale stejně hladový algoritmus:

„Zvolme jakoukoli hranu v grafu. Tu musíme pokrýt alespoň jedním jejím koncem, ale my vybereme do výsledného pokrytí oba dva konce. Hranu i její koncové vrcholy (se všemi sousedními hranami) odstraníme a pokračujeme dál stejně.“

Má tento algoritmus konstantní aproximační poměr?

Příklad šestý

- Vzpomeňte si na optimalizační metody a formulujte problém Minimální vrcholové pokrytí jako celočíselný program. Začněte proměnnými x_v pro každý vrchol $v \in V(G)$ …
- V optimalizačních metodách se často převede celočíselný program na lineární program a pak se řeší ten (v polynomiálním čase). Nás ale zajímají jen celočíselná řešení. Vymyslete jednoduchý způsob, jak z řešení lineární relaxace z předchozího bodu dostanete celočíselné *přípustné* řešení, které zároveň bude 2-aproximací.

Příklad sedmý Nakonec proberme Minimální vážené vrcholové pokrytí – opět nám jde o to nalézt vrcholové pokrytí, ale tentokrát má každý vrchol v svou váhu $w_v \geq 0$ a my chceme najít pokrytí minimální celkové váhy, ne velikosti.

Navrhněte 2-aproximační algoritmus pro Minimální vážené vrcholové pokrytí.