5. CVIČENÍ Z ÚVODU DO APROXIMACÍ

zaokrouhlování lineárních programů

PŘÍKLAD PRVNÍ Následující problémy formulujte jako celočíselné nebo lineární programy:

1. MINIMÁLNÍ VRCHOLOVÉ POKRYTÍ: Na vstupu máme neorientovaný graf G. Cílem je nalézt co nejmenší množinu VC takovou, že pro každou hranu $uv \in E(G)$ platí, že buď $u \in VC$, nebo $v \in VC$ (mohou být oba).
2. MINIMÁLNÍ VÁŽENÁ DOMINUJÍCI MNOŽINA: Na vstupu máme neorientovaný graf G s váhami na vrcholech $w_v, w_v \geq 0$. Cílem je naleznout co nejlehčí (vůči váze) množinu vrcholů D takovou, že každý vrchol je buď uvnitř D, nebo alespoň jeden jeho soused je v D.
3. MAXIMÁLNÍ TOK: Máme dán orientovaný graf $\vec{G}$ s kapacitami hran $c_{\vec{e}}$ a dvěma význačnými vrcholy s a t. Máme najít co největší tok z s do t, který respektuje kapacity.
4. MAXIMÁLNÍ VRCHOLOVÝ ŘEZ: Námi už zkoumaný problém. Máme dán neorientovaný graf G s váhami na hranách w_e. Cílem je najít rozdělení vrcholů na skupiny V a W tak, aby váha všech hran mezi skupinami byla co největší.
5. 3-BAREVNOST: Na vstupu máme neorientovaný graf G. Vytvořte celočíselný program, který nalezne ohodnocení vrcholů grafu třemi barvami tak, že žádná hrana není jednobarevná (a selže, pokud to nejde).

PŘÍKLAD DRUHÝ Zkuste si procvičit zaokrouhlování lineárních programů při tvoření 2-aproximačního algoritmu pro ROZLOŽENÍ ZÁTĚŽE V SÍTI SONET.

V tomto problému se síť skládá z kružnice na n vrcholech. K síti máme seznam požadavků, kde každý požadavek má určen zdrojový vrchol a cílový vrchol. Ke každému požadavku musí být přiřazena právě jedna z dvou možných cest mezi zdrojem a cílem (po směru nebo proti směru hodinových ručiček). Cílem je minimalizovat zátěž (počet použití) na nejvíce zatížené hraně.

PŘÍKLAD TŘETÍ Nyní uvažme problém MAX ORIENTOVANÝ ŘEZ – na vstupu dostaneme orientovaný graf $G = (V, \vec{E})$ s nezápornými vahami na hranách, a cílem je najít podmnožinu vrcholů S takovou, že $\vec{E}(S, V \setminus S)$ (hrany vedoucí z S do zbytku, ale ne opačné) mají co největší váhu.

Navrhněte pravděpodobnostní $\frac{1}{4}$-aproximační algoritmus pro MAX ORIENTOVANÝ ŘEZ. (Tento algoritmus nepoužívá lineární programy.)

PŘÍKLAD ČTVRTÝ Zkusme vymyslet lepší algoritmus pro MAX ORIENTOVANÝ ŘEZ:

1. Navrhněte $\{0, 1\}-$celočíselný program, který řeší MAX ORIENTOVANÝ ŘEZ. Použijte proměnné y_{ij} pro hrany a proměnné x_i pro vrcholy – a správně je propojte.
2. Vyberme každý vrchol v_i s pravděpodobností $1/4 + x_i^*/2$, kde x_i^* je optimální řešení lineární relaxace celočíselného programu z podbodu 1. Dostaneme s toutou volbou 1/2-aproximaci?

PŘÍKLAD PÁTÝ Uvažujme problém POKRYJ CO NEJVÍC. Na vstupu dostaneme n – velikost univerza $U = \{1, 2, \ldots, n\}$, a také seznam m podmnožin univerza, čili vypsané množiny $S_1, S_2, \ldots, S_m$, kde každá S_i je podmnožinou U. Na vstupu také dostaneme cislo $k \leq m$.

Úkolem algoritmu je vybrat k podmnožin S_i tak, aby vybraných k podmnožin pokrývalo co nejvíc prvků univerza.

1. Zformulujte problém POKRYJ CO NEJVÍC jako celočíselný lineární program.
2. Navrhněte randomizovaný aproximační algoritmus pro POKRYJ CO NEJVÍC na základě zaokrouhlování lineárního programu. Proveďte jeho analýzu a odvoďte aproximační poměr. Tentokrát nevíte, jaký přesný poměr má vyjít – což většinou dopředu není známo.