4. CVIČENÍ Z ÚVODU DO APROXIMACÍ

Je hladové rozvrhování do batohu splnitelné?

Příklad první Uvažujme Rozvrhování se závislostmi: rozvrhujeme úlohy různých délek na m počítačů (m součástí vstupu), ale navíc máme závislosti mezi úlohami. Přesněji, na úlohách je definovaný acyklický graf a úlohu je možné zahájit až ve chvíli, kdy jsou dokončeny všechny úlohy jí předcházející v grafu.

1. Dokažte, že následující dolní odhad na optimum je pravdivý:
   „$OPT \geq$ délka libovolného řetězu úloh, kde řetěz chápeme jako orientovanou cestu v grafu závislostí. Jeho délka je pak součet délek úloh na řetězu."
2. Navrhněte hladový algoritmus a dokažte, že je 2-aproximační.

Příklad druhý Mějme klasický NP-těžký Problém batohu, to jest je dáno n objektů, přičemž i-tý objekt má váhu w_i a cenu c_i, a my máme batoh o nosnosti B, do kterého chceme naskládat objekty o co největší celkové ceně.

1. Rozmyslete si, proč „naivně hladový algoritmus", to jest „vložím nejdražší objekt, co se vejde, do batohu, a pokračuji dále", je špatným.
2. OK, tak zkusme tento algoritmus: setřiďme si objekty podle „hustoty" (poměr cena/velikost), procházejme je od nejhustšího a berme jen ty, které se vejdou. Prozradím vám, že i tento algoritmus nebude mít dobrý výsledek. Najděte vstup, kde selže.
3. Navrhněte už funkční 2-aproximační algoritmus pro tento problém. Algoritmus nemusí nutně být hladový.
   Hint: Když procházíte objekty podle hustoty, může se vám stát, že objekt P se vám do batohu nevejde s věcmi, co už tam máte. Co je chytré udělat potom?

Příklad třetí Další ze zajímavých metrických problémů je Problém k center, kde na vstupu je množina $V, |V| = n$ bodů v metrice a my vybíráme k center (k-prvkovou podmnožinu z n bodů) tak, aby body byly co nejblíže centrům – aby ten nejvzdálenější bod od všech středisek byl co nejblíže. Formálně minimalizujeme funkci $u(S) = \max_{p \in V} d(p, S)$, kde $d(p, S)$ je vzdálenost mezi p a nejbližším z S.

Navrhněte a zanalyzujte hladový 2-aproximační algoritmus pro Problém k center.

Tip: Pozor, náš algoritmus i optimum zvolí právě k bodů, aproximační poměr ovlivní jen povolenou chybu vzdálenosti. V analýze se k bodů optima nějak musí projevit – dává tedy smysl začít tak, že uvažujete optimální výběr k bodů a srovnáváte ji s vaším výběrem.

Příklad čtvrtý Pravděpodobnostní algoritmus pro Max Sat, který zvolí $x_i = 1$ s pravděpodobností $1/2$, je jen $1/2$-aproximační v instancích, které obsahují mnoho klauzulí (x_i) nebo $(\neg x_j)$, ale je efektivnější pro klauzule délky 2 a delší.

Pojďme nahlédnout, že můžeme předpokládat trochu hezčí vstup:

1. Dokažte, že máme-li c-aproximační algoritmus pro Max Sat na vstupech, které neobsahují negativní jednoklauzule $(\neg x_i)$, tak ho umíme přetvořit na c-aproximační algoritmus pro Max Sat s všemi přípustnými vstupy.
2. Dokažte, že stejný převod platí i pro Vážený Max Sat, kde každá klauzule má váhu w_i a my maximalizujeme vážený součet splněných klauzulí $\max \sum_i w_i C_i$.

PŘÍKLAD PÁTÝ Připomeňte si celočíselný program pro MAX SAT a jeho lineární relaxaci. Na přednášce jste viděli 3/4-aproximační algoritmus pro MAX SAT, založený na výběru lepšího ze dvou řešení, z čehož jedno řešení bylo založené na zaokrouhlování této lineární relaxace. Lepším zaokrouhlováním lineární relaxace se lze vyhnout výběru ze dvou řešení při zachování aproximačního poměru 3/4.

Najděte instanci, tedy množinu klauzulí, takovou, že pro optimum relaxace OPT_r a optimum instance OPT platí $\mathrm{OPT} = (3/4)\mathrm{OPT}_r$. To ukazuje, že pomocí této lineární relaxace nedostaneme lepší než 3/4-aproximační algoritmus. (Poměru mezi OPT_f a OPT se říká celočíselná mezera, v angličtině integrality gap.)

Hint: stačí 2 proměnné a 4 klauzule. Optimum relaxace je splní všechny, kdežto celočíselné optimum splní jen 3 klauzule.