NP-úplné a NP-těžké úlohy, Cookeova věta, heuristiky na řešení NP-těžkých úloh, pravděpodobnostní algoritmy

1. Polynomiální redukce úloh

- U, V jsou **rozhodovací úlohy**
- U se **redukuje** na úlohu V, jestliže existuje algoritmus (program pro RAM, Turingův stroj) M, který pro každou instanci I úlohy U zkonstruuje instanci I′ úlohy V tak, že I je ANO instance U iff I′ je ANO instance V
- značí se $U \lhd V$ nebo $U \lhd_p V$, pokud je redukční algoritmus polynomiální
- tranzitivita redukce - pokud $U \lhd_p V$ a $V \lhd_p W$, pak $U \lhd_p W$

2. NPC (NP complete) úlohy

- rozhodovací úloha U je NP úplná, jestliže
* U je ve třídě NP
* každá NP úloha se polynomiálně redukuje na U

Tvrzení: U, V jsou NP úlohy, pro které platí $U \lhd_p V$. Pak
* jestliže V je ve třídě P, pak také U je ve třídě P
* jestliže U je NP úplná úloha, pak také V je NP úplná úloha

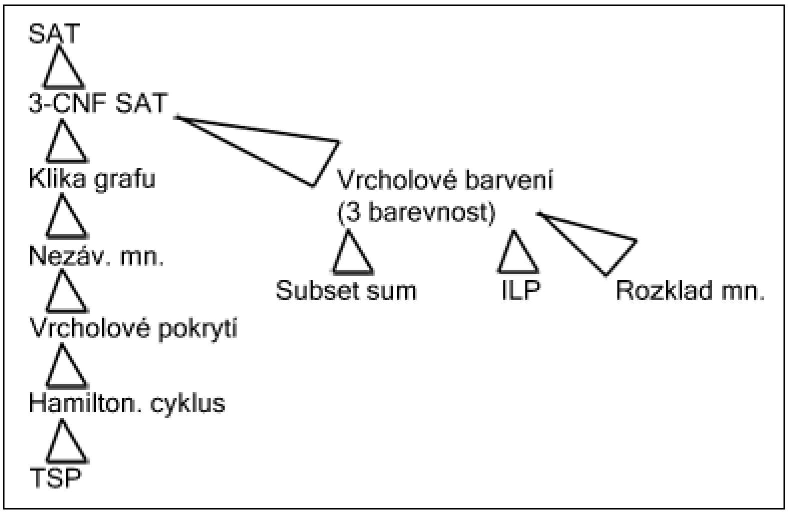
3. NP obtížné úlohy

- jestliže o některé úloze U pouze víme, že se na ní polynomiálně redukuje některá NPC úloha, U je NP obtížná
- jsou alespoň tak těžké jako všechny NP úlohy

4. Cookova věta

- úloha SAT (splňování formulí v konjunktivním normálním tvaru) je NP úplná úloha
- **důkaz**:
1. úloha SAT je ve třídě NP. Nedetermin. alg. vygeneruje ohodnocení logických proměnných. V polynomiálním čase lze ověřit, jestli je formule v daném ohodnocení pravdivá či ne.
2. Druhá část důkazu spočívá v popisu práce Turingova stroje formulí výrokové logiky.

5. Převody NPC úloh



Převod SAT na 3-CNF SAT

Rozhodovací úloha: Je dána formule φ v konjunktivním normálním tvaru, kde každá klausule má 3 literály. Je formule φ splnitelná?

Tvrzení. Dokazuje se zavedením zavedením nových logických proměnných. Platí SAT $\triangleleft_p$ 3-CNF SAT, tedy 3-CNF SAT je NPC.

Převod 3-CNF SAT na problém klik

- klika je úplný podgraf
Rozhodovací úloha: Je dán prostý neorientovaný graf G = (V,E) bez smyček a číslo k. Existuje v grafu G klika o alespoň k vrcholech?

Tvrzení. Dokazuje se přes k-partitní neorientovaný graf. Platí 3-CNF SAT $\triangleleft_p$ problém klik, tedy problém klik je NPC.

Převod problému klik na nezávislé množiny

- **nezávislá mn.** vrcholů nesdílí žádnou společnou hranu
Rozhodovací úloha: Je dán prostý neorientovaný graf G = (V,E) bez smyček a číslo k. Existuje v grafu G nezávislá mn. o k vrcholech?

Tvrzení. Dokazuje se přes opačný graf. Platí problém klik $\triangleleft_p$ nezávislá mn., tedy nezávislá mn. je NPC.

Převod nezávislé množiny na vrcholové pokrytí

- **vrcholové pokrytí** je minimální mn. vrcholů, se kterými incidují všechny hrany
Rozhodovací úloha: Je dán prostý neorientovaný graf G = (V,E) bez smyček a číslo k. Existuje v grafu G vrcholové pokrytí o k vrcholech?

Tvrzení. Pokud N je nezáv.mn., pak V mínus N je vrcholové pokrytí. Platí nezávislá mn. $\triangleleft_p$ vrcholové pokrytí, tedy vrcholové pokrytí je NPC.

Převod vrcholového pokrytí na 3-barevnost

- **3-barevnost** – vrcholy grafu jsou obarveny 3 barvami tak, že dvě stejné barvy spolu nesousedí

Rozhodovací úloha: Je dán prostý neorientovaný graf G = (V,E) bez smyček a číslo k. Lze graf obarvit pomocí 3 barev?

Převod na 3-barevnost na ILP

- **ILP** je lineární programování, kde proměnné jsou celá čísla

Rozhodovací úloha: Je dán prostý neorientovaný graf G = (V,E) bez smyček a číslo k. Lze graf obarvit pomocí 3 barev?

Dokazuje se soustavou rovnic a nerovnic, které vyjadřují 3-barevnost. Přiřazení $x_v^b = 1$ znamená, že vrchol v má barvu $b \in \{c, m, z\}$. Jedna z rovnic je např. $x_v^c + x_v^m + x_v^z = 1$, což znamená, že vrchol může mít jen jednu barvu.

Převod na 3-barevnost na rozklad množiny

- **rozklad množiny:** je dána konečná mn. X a systém jejích podmn. A je rozklad mn. X, jestliže jsou splněny dvě podmínky:

1. každý prvek x∈ X leží v některé podmnožině B⊆A, $\bigcup B_i = X$
2. žádné dvě různé podmnožiny z A nemají společný prvek, tj. jsou po dvou disjunktní.

Tvrzení: 3-barevnost $\triangleleft_p$ problém rozkladu, tedy problém rozkladu je NPC.

Existence hamiltonovského cyklu

Rozhodovací úloha: Existuje v orientovaném grafu G hamiltonovský cyklus (cyklus procházející všemi vrcholy)?

Tvrzení: vrcholové pokrytí $\triangleleft_p$ existence hamiltonovského cyklu

6. Heuristiky

- NPC úlohy často nelze optimálně vyřešit v rozumném (polynomiálním čase), ale někdy nám stačí **přibližné řešení**, které se tomu optimálnímu blíží (heuristické alg.)
- **heuristické alg.** pracují v polynom. čase
- u **aproximačních alg.** umíme zaručit jak daleko je nalezené řešení od optimálního

Tvrzení: Kdyby existovala konstanta r a polynomiální algoritmus A, který pro každou instanci obchodního cestujícího I najde trasu délky D ≤ r OPT(I), kde OPT(I) je délka optimální trasy instance I, pak P = NP.

Trojúhelníková nerovnost

- pokud TSP splňuje **trojúhelníkovou nerovnost** (pro každá tři města i, j, k platí d(i, j) ≤ d(i, k) + d(k, j)), pak existuje polynomiální algoritmus A, který pro I najde trasu délky D ≤ 2OPT(I)

- **popis algoritmu**: máme úplný graf G s vrcholy V = {1, 2, . . . , n} a ohodnocením d
1. v grafu G najdeme minimální kostru
2. kostru (V,K) prohledáme do hloubky z libovolného vrcholu
3. trasa TSP je dána pořadím první návštěvy uzlů během prohledávání grafu

Christofidesův algoritmus
- instance TSP splňuje trojúhelníkovou nerovnost, pak Christ. algoritmus najde trasu o délce $D \leq \frac{3}{2}$ OPT(I)
- používá min. kostru, DFS, vytvoření úplného grafu, nejlevnější perfektní párování, eulerovský tah

7. Třída co-NP

- jazyk L patří do třídy co−NP, jestliže jeho **doplněk** patří do třídy NP

8. Pravděpodobnostní algoritmy

Randomizovaný Turingův stroj RTM

= TS se dvěma nebo **více páskami**, kde první páska má stejnou roli jako u deterministického TS, ale druhá páska obsahuje **náhodnou posloupnost 0 a 1**

Třída RP

- jazyk L patří do třídy RP právě tehdy, když existuje RTM M takový, že:
1. jestliže $w \in L$, stroj M se ve stavu q_f zastaví s **pravděpodobností** 0%
2. jestliže $w \in L$, stroj M se ve stavu q_f zastaví s pravděpodobností alespoň 50%
3. každý běh M (tj. pro jakýkoli obsah druhé pásky) trvá maximálně **polynomiální** počet kroků

- např. **Millerův test prvočíselnosti** je příklad algoritmu, který splňuje všechny tři podmínky (utvoříme-li k němu odpovídající RTM) a proto jazyk L, který se skládá ze všech složených čísel, patří do třídy RP

Turingův stroj typu Monte-Carlo

- RTM splňující podmínky 1 a 2 třídy RP se nazývá TM typu Monte-Carlo
- nemusí pracovat v polynomiálním čase

Třída ZPP

- jazyk L patří do třídy ZPP, pokud existuje RTM M takový, že:
1. Jestliže $w \in L$, M se úspěšně zastaví ve stavu q_f s pravděpodobností 0
2. Jestliže $w \in L$, stroj M se úspěšně zastaví ve stavu q_f s pravděpodobností 1
3. **Střední hodnota** počtu kroků M v jednom běhu je **polynomiální**

- M neudělá chybu, ale není zaručen polynomiální počet kroků, pouze střední hodnota počtu kroků je polynomiální
- TS typu **Las Vegas** splňuje všechny 3 podmínky

9. Třídy PSPACE a NPSPACE

- jazyk L patří do třídy PSPACE (NPSPACE), pokud když existuje deterministický (nedeterministický) TM, který přijímá jazyk L a pracuje s **polynomiální paměťovou složitostí**
- platí P $\subseteq$ PSPACE

10. Zdroje

[1] Demlová, Marie: Přednášky 5-8 z TAL.
http://math.feld.cvut.cz/demlova/teaching/tal