Výpočtová zložitosť

Marian.Mach@tuke.sk
http://neuron.tuke.sk/~machm

Marec, 2013

Teória výpočtovej zložitosti

- veľký priestor kandidátov inštancie problému
  - typicky aspoň exponenciálne veľký vzhľadom na veľkosť inštancie
    - SAT: 2^n $|S(2n)| / |S(n)| = 2^n$
    - TSP: $n! / 2n$ $|S(2n)| / |S(n)| = (2n)! / 2n!$
    - GC: k^n $|S(2n)| / |S(n)| = k^n$
- otázky
  - čas potrebný pre vyriešenie inštancie problému ako funkcia veľkosti inštancie
  - možno prehľadávať veľký priestor efektívne?
    - efektívne = polynomiálny čas voči veľkosti inštancie

Zložitosť problémov a algoritmov

- typy zložitosti
  - časová a priestorová zložitosť
  - priemerný prípad a najhorší prípad
- zložitosť algoritmu
  - funkčná závislosť medzi veľkosťou inštancie a časom/priestorom potrebným pre vyriešenie
  - na báze modelu formálneho stroja
- zložitosť inštancie problému
  - meraná zložitosťou najlepšieho algoritmu pre daný problém
  - problémy sú klasifikované do zložitostných tried

Triedy P a NP

- trieda P
  - problém riešiteľný deterministickým strojom v polynomiálnom čase
- trieda NP
  - riešiteľný nedeterministickým strojom v polynomiálnom čase
- vzťah medzi P a NP
  - $P \subseteq NP$ $\quad\quad NP \subseteq P$??
- mnoho aplikačných úloh patrí do triedy NP
  - s rastúcou veľkosťou inštancií sa rýchlo stávajú neriešiteľné

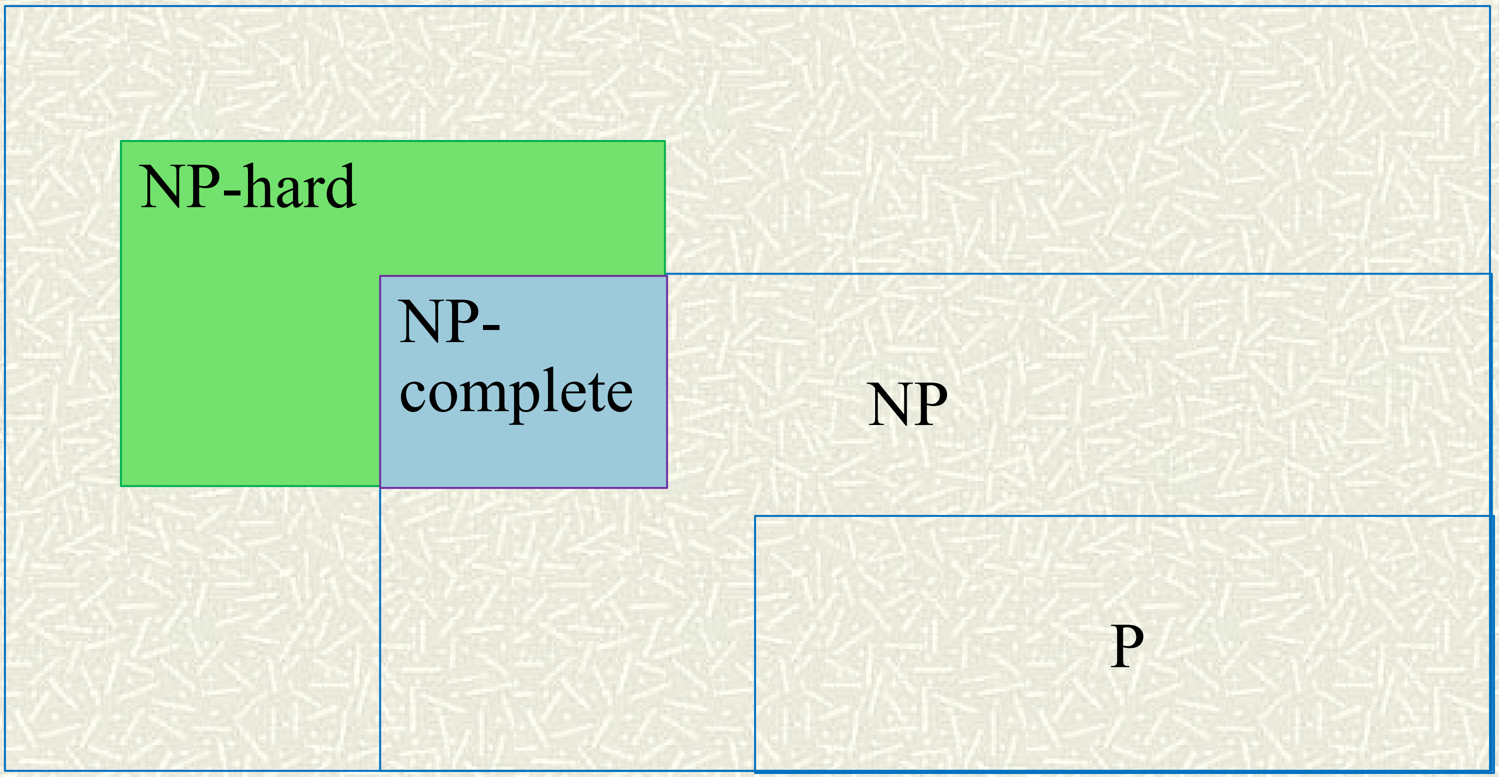
Triedy Np-hard a Np-complete

- trieda NP-hard
  - problém je aspoň tak zložitý ako hocijaký problém v NP
    - problémy z NP sú naň polynomiálne redukovateľné
  - nemusí patriť do NP (môže byť ešte zložitejší)
- trieda NP-complete
  - sú to problémy, ktoré patria do NP-hard a súčasne sú z NP
- príklady
  - SAT – NP-complete (špec. prípady sú P)
  - TSP – NP-hard (špec. prípady sú P)

Kombinatorické úlohy

- NP-complete rozhodovací problém
  - hľadací a rozhodovací variant sú rovnako zložité
    - rozhodovanie zahrnuté v hľadaní
    - iteračná fixácia zložiek a rozhodovanie o existencii riešenia
- NP-hard optimalizačný problém
  - hľadací a ohodnocovací variant sú rovnako zložité
    - ohodnocovanie zahrnuté v hľadaní
    - riešenie pomocou asociovaného rozhodovacieho problému
  - ak optimalizačný problém je možné vyriešiť efektívne, tak vieme riešiť aj rozhodovací problém

Porovnanie tried



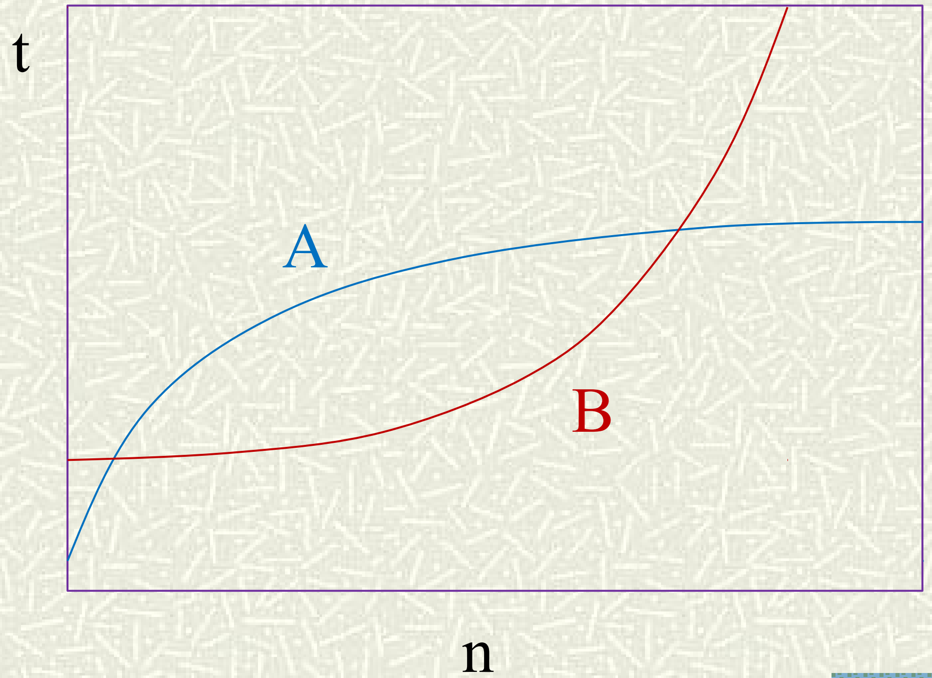
Riešenie NP problémov

- veľa praktických kombinatorických problémov je v NP triede (resp. NP-hard)
  - nevieme takéto úlohy riešiť v praxi ?
- možnosti pre riešenie
  - vlastnosti riešenej inštancie
    - malá rozmernosť problému
    - príslušnosť k podtriede zložitosti P
    - najhorší vs. priemerný prípad
    - úloha konštánt
  - zmena prístupu
    - stochastický prístup
    - aproximačný prístup

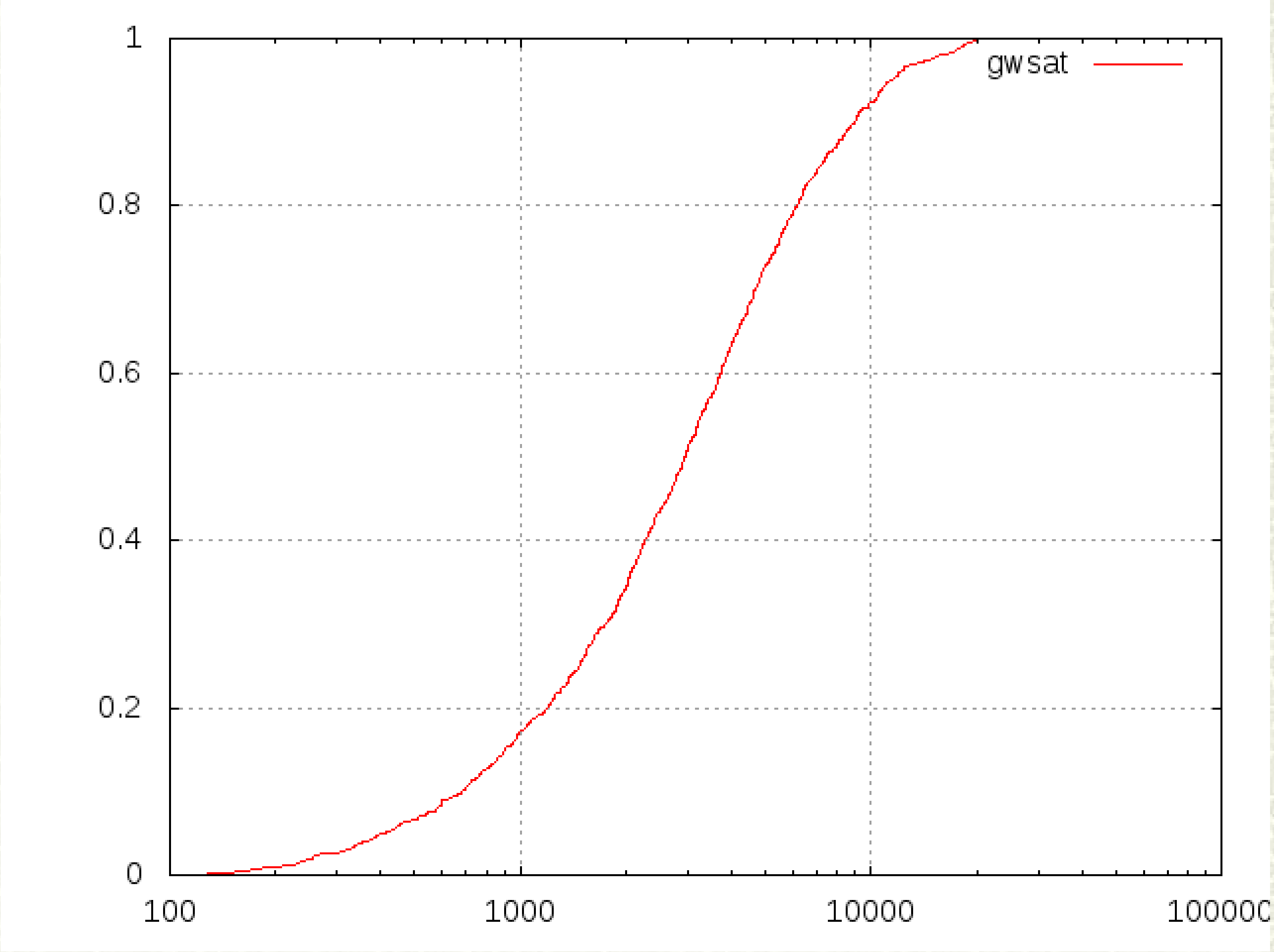
NP jednoduchšie ako P

- výpočtová zložitosť
  - skúma asymptotické chovanie
  - zanedbáva konštanty

- príklad
  - algoritmus A
    - polynomiálny
  - algoritmus B
    - exponenciálny



Stochastický prístup



Aproximačný prístup

- hľadanie suboptimálneho riešenia
- stupeň suboptimality
  - aproximačný pomer: $r = q / q^*$
- asociovaný aproximačný problém
  - minimalizácia: $r \leq \alpha = 1 + \epsilon$ (pre $\epsilon > 0$)
  - maximalizácia: $r \geq \alpha = 1 - \epsilon$ (pre $\epsilon > 0$)
- príklad – TSP
  - všeobecný – neexistuje α
  - trojuholníková nerovnosť – $\alpha = 1.5$ (1976)
  - Euklidovský TSP – α ľubovoľné (1998)

Aproximačné triedy

- α-aproximačný algoritmus
  - problém patrí do nejakej triedy v závislosti na kvalite existujúceho *polynomiálneho* aproximačného algoritmu
- triedy
  - APX nejaké α
  - PTAS ľubovoľné α, polynomiálny voči *n*
  - FPTAS ľubovoľné α, polynomiálny voči *n* a *1/ ϵ*
- vzťahy medzi triedami
  - FPTAS $\subseteq$ PTAS $\subseteq$ APX $\subseteq$ NP-hard
  - rovnosti iba v prípade, ak $P = NP$

Randomizované algoritmy

- algoritmus robí svoje rozhodnutia náhodne
  - hádzanie *mince*
    - uniformná distribúcia pravdepodobnosti
  - dosahované parametre sú náhodné premenné
    - riešenie, aproximačný pomer, doba behu
- vlastnosti
  - výkonnosť
    - často urýchlenie voči deterministickým schémam
    - distribúcia nad algoritmami - reštarty
  - zložitosť vyjadrenia
    - jednoduchá štruktúra algoritmu
    - náhodné generovanie riešenia

Randomizovaný MAX-SAT

- $(1 – 1 / 2^k)$-aproximačný algoritmus vykonávaným v polynomiálnom čase

- input: π (n, m, k)
- n – počet premenných,
- m – počet klauzúl,
- k – minimálna dĺžka klauzúl
- output: *s є S*
- **repeat**
- s **=** random-variable-assignment()
- w **=** count-satisfied-clauses(s)
- **until** $w <= m – m / 2^k$