Automatizované řešení úloh s omezeními

Martin Kot

Katedra informatiky, FEI,
Vysoká škola báňská – Technická universita Ostrava
17. listopadu 15, Ostrava-Poruba 708 33
Česká republika

27. září 2012

Přednášející

Garant předmětu:

Jméno: prof. RNDr. Petr Jančar, CSc.

E-mail: petr.jancar@vsb.cz

Místnost: A1046

Přednášející a cvičící:

Jméno: Ing. Martin Kot, Ph.D.

E-mail: martin.kot@vsb.cz

Místnost: A1024

WWW stránky

Webové stránky k předmětu naleznete na adrese:

`http://www.cs.vsb.cz/kot/aruo`

Na těchto stránkách najdete:

- Informace o předmětu
- Slidy z přednášek
- Aktuální informace

Požadavky

Klasifikovaný zápočet (100 bodů):

- Projekt (60 bodů)
- Zápočtová písemka (40 bodů)

Pro získání zápočtu je potřeba získat alespoň 31 bodů z projektu a alespoň 20 bodů ze zápočtové písemky.

Výukové materiály

Hlavními výukovými texty jsou:

- Slajdy
- Rina Dechter
  Constraint Processing,
  Morgan Kaufmann Publishers, 2003.
- Slajdy autorky knihy

Úvod

Úvod

- Úlohy s omezeními řešíme v běžném životě
- Obvykle je řešíme intuitivně, bez počítače
- Např. hospodaření s penězi, sestavování jídelníčku, ...
- S rostoucí složitostí úloh se obracíme na počítače
- V obecnosti je většina úloh s omezeními "výpočetně nezvládnutelná" (úlohy bývají NP-těžké)
- Nemůžeme počítat s efektifními algoritmy řešícími úlohy v plném rozsahu
- Můžeme navrhnout řešení efektivní pro většinu instancí

Základní koncept

- **Proměnné** (variables) - mohou nabývat různé hodnoty
- **Doména, definiční obor** (domain) - množina možných hodnot pro danou proměnnou
- **Omezení** (constraints) - pravidla kladoucí podmínky na možné ohodnocení proměnných a jejich kombinací
- Často je více možných modelů problému (zasedací pořádek hostů na svatbě - hosté jako proměnné nebo židle jako proměnné)
- **Síť omezení** (constraint network), **problém s omezeními** (constraint problém) - model zahrnující proměnné, jejich domény a omezení
- Pozn. pojem síť omezení vychází z historie, kdy se výzkum zaměřoval na typy problémů, kde závislosti byly popsatelné jednoduchými grafy
- **Řešení** (solution) - přiřazení jedné hodnoty z domény pro každou proměnnou tak, aby nebyla porušena žádná omezení
- Řešení může být jedno nebo více, ale taky žádné
- **Splnitelný** (satisfiable), **konzistentní** (consistent) - má řešení
- **Nesplnitelný** (unsatisfiable), **nekonzistentní** (inconsistent) - nemá řešení

Typické úlohy

U problémů s omezeními nás obvykle zajímají tyto úlohy:

- určit, zda existuje řešení (splňující všechny omezení)
- najít jedno řešení
- zjistit jestli přiřazení několika hodnot proměnných může být rozšířeno na řešení
- najít optimální řešení vzhledem k cenové funkci (cost function)

Všechny tyto úlohy souhrně označujeme jako constraint satisfaction problems (CSP)

Příklad – n dam na šachovnici

- Úkolem je umístit n dam na šachovnici $n \times n$ tak, aby se vzájemně neohrožovaly
- Model např.:
  - proměnné $x_1, \ldots, x_n$ pro sloupce
  - doména D_i každé proměnné je $D_i = \{1, \ldots, n\}$ (číslo řádku)
  - omezení pro každý pár sloupců zakazují sdílení řádku nebo diagonály

Příklad - křížovky

- Úkolem je ze slovníku přiřadit slova buňkám horziontálně a vertikálně
- Každé obsaditelné políčko je proměnná
- Doménou každé proměnné je abeceda
- Omezení jsou dána slovníkem
- Slidy Riny Dechter - 1. kapitola, slide 3

Příklad - barvení map

- Úkolem je obarvit všechny státy politické mapy 4 barvami tak, aby sousední státy (s nenulovou délkou hranice) neměly stejnou barvu
- Od roku 1976 dokázáno, že to vždy jde (Appel, Haken)
- Abstrakce grafem - vrchol pro každou zemi, hrana při společné hranici
- Vrcholy jsou proměnné, domény obsahují barvy, omezením je zákaz stejné barvy pro sousední vrcholy
- Zobecněním je (vrcholové) barvení grafu k barvami
- Mnoho praktických problémů se dá převést na barvení grafu (např. přidělování radiových frekvencí)
- Slidy Riny Dechter - 1. kapitola, slide 4

Příklad - konfigurace, návrh

- Celá řada problémů hledající optimální návrh nebo konfiguraci při splnění nějakých podmínek
- Např. návrh automobilů, konfigurace počítačů, rozložení místností na patře budovy, ...
- Slidy Riny Dechter - 1. kapitola, slidy 5-7

Příklad - Huffmanovo-Clowesovo ohodnocení scény

- Jeden z prvních problémů s omezeními (definován 1975)
- Cílem je rozpoznání 3D objektů ve scéně prostřednictvím interpretace čar v 2D kresbě
- Úsečkám přiřazujeme ohodnocení ($+$ konvexní, $-$ konkávní, $<$ okrajové) u krajních bodů
- Je jen omezený počet možných přiřazení více úsečkám se společným krajním bodem
- Omezení jsou dána tím, že úsečka musí mít stejné ohodnocení na obou stranách
- Slidy Riny Dechter - 1. kapitola, slidy 8-9

Příklad - plánování

- Plánování (scheduling) zahrnuje mnoho různých problémů
- Příkladem je např. plánování výroby v dílně - omezený počet strojů a lidí, snaha vyrobit co nejvíce, stroje a lidé by neměli zahálet
- Patří sem i tvorba rozvrhů - učitelé, studenti, místnosti, předměty, ...
- Často jsou úlohy voleny jako proměnné a domény obsahují časy (začátku vykonávání úkolu)

Inference

- **Inference** - snaha vytvořit jednodušší problém díky jeho přeformulování
- Někdy inferenční metody najdou řešení nebo zjistí nekonzistentnost
- **Lokální konzistentnost** (local consistency) - díváme se na podčásti problému a požadujeme, aby tam nebyly sporné části
- Např. při $x_1, \ldots, x_n$, $D_i = \{1, \ldots, 10\}$ a omezení, že x_1 je ostře větší, než ostatní, nepotřebujeme v D_1 nechat 1.
- Algoritmy pro lokální konzistentnost se také nazývají algoritmy **šíření omezení** (constraint propagation)
- Většina algoritmů šíření omezení je polynomiální a převádějí síť na jinou, ekvivalentní ale explicitnější, odvozením nových omezení a jejich přidáním k problému

Inference

- **Hranová konzistentnost** (arc-consistency) - každé možné přiřazení hodnoty proměnné (z její domény) má možné přiřazení z domény jiné vybrané proměnné
- **Konzistentnost cesty** (path-consistency) - každé přiřazení hodnoty 2 proměnným (z jejich domén) je rozšiřitelné na třetí proměnnou
- **i-konzistentnost** (i-consistency) - každé přiřazení hodnoty $i - 1$ proměnným (z jejich domén) je rozšiřitelné na $i$tou proměnnou
- Vynucení i-konzistentnosti je obvykle výpočetně náročné - exponenciální časově i prostorově vzhledem k i
- Často může dojít k dokázání nekonzistentnosti sítě vyprázdněním některé z domén při zajišťování např. i-konzistence

Prohledávání

- Nejčastěji se používá backtracking
- Prochází se prostor řešení do hloubky
- Každý krok je přiřazení hodnoty jedné proměnné z její domény
- Když není možné (konzistentně) přiřadit žádnou hodnotu, nastává tzv. **dead-end**
- Dead-end se řeší backtrackingem - vrátíme se k předchozí proměnné a nastavíme ji jinou hodnotu

Možná vylepšení backtrackingu

- Vylepšení dělíme na ta pro dopředný pohyb (look-ahead schemes) a pro zpětný pohyb (look-back schemes)
- Dopředné:
  - Snaha přiřazovat hodnoty proměnným tak, aby byla co největší šance na úspěch nebo rychlé zjištění nekonzistentnosti sítě
  - Nejprve přiřazujeme proměnným, kterých se týká hodně omezení
  - Při výběru hodnoty pro proměnnou vybereme nejméně omezenou hodnotu
- Zpětné:
  - Řeší např. jak daleko se vrátit (backjumping)
  - Analýzují se příčiny vzniklého dead-endu
  - Zaznamenávají se příčiny dead-endu ve formě nových omezení, aby nenastal ze stejného důvodu znovu (constraint learning, no-good recording)

Stochastické lokální hledání

- Algoritmus vylepšuje instanciaci změnou hodnot proměnných tak, aby maximalizoval počet splněných omezení
- Neúplné algoritmy - můžou se zaseknout na lokálním minimu cenové funkce, nedokáží nekonzistentnost sítě
- Spolu s heuristikami se ukazují užitečné v praxi pro velké a těžké problémy