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

17. října 2013

Šíření omezení

Šíření omezení (constraint propagation)

- Můžeme dokázat nekonzistentnost sítě
- Můžeme vyloučit některé kombinace přiřazení hodnot proměnným a tím urychlit výpočet
- Můžeme i najít řešení - nová omezení způsobí, že každé řešení se najde prostým prohledáváním do hloubky, kde nenastane nikdy dead-end
- Většinou ale řešení problému vyvozováním (inference) je výpočetně náročné, vyžaduje přidání exponenciálního množství nových omezení
- Obvykle se přidávaji jen nějaká omezení, prohledávání pak může narazit na dead-end, ale končí relativně rychle
- Algoritmy přidávající omezený počet omezení označujeme vynucení lokální konzistentnosti (local consistency-enforcing), omezené vyvozování konzistence (bounded consistency inference) nebo šíření omezení (constraint propagation)

Hranová konzistentnost

- V minimální síti platí, že přiřazení jakékoliv hodnoty z domény proměnné je rozšiřitelné na jakoukoliv další proměnnou
- Takovou vlastnost nazýváme **hranová konzistentnost** (arc-consistency)
- Může být splněna také v neminimálních sítích
- Může být vynucena na každé síti efektivním výpočtem, často nazývaným **propagace** (propagation)
- Definice: Pro danou síť $\mathcal{R} = \{X, D, C\}$ s $R_{ij} \in C$ je proměnná x_i **hranově konzistentní** (arc-consistent) vzhledem k x_j právě tehdy, když pro každou hodnotu $a_i \in D_i$ existuje hodnota $a_j \in D_j$ tž. $(a_i, a_j) \in R_{ij}$.
- Definice: Podsíť (nebo hrana) definovaná $\{x_i, x_j\}$ je hranově konzistentní, právě když x_i je hranově konzistentní vzhledem k x_j a x_j je hranově konzistentní vzhledem k x_i
- Definice: Síť omezení je hranově konzistentní, právě když všechny její hrany (podsítě velikosti 2) jsou hranově konzistentní

Revidující procedura

- Slidy R. Dechter, chapter 3, slide 7
- Převádí síť na hranově konzistentní redukcí domén proměnných v rozsahu omezení
- Aplikována na x_i, x_j vrací největší doménu D_i pro kterou x_i je hranově konzistentní vzhledem k x_j
- Složitost v nejhorším případě je $O(k^2)$, kde k je mez velikosti domén

Algoritmus ARC-CONSISTENCY-1 (AC-1)

- Slidy R. Dechter, chapter 3, slide 9
- Brute-force algoritmus zajišťující hranovou konzistentnost sítě
- Aplikuje revidující proceduru na všechny dvojice proměnných, které se vyskytují v omezeních dokud se nějaká doména mění
- Může zjistit nekonzistentnost sítě
- Složitost v nejhorším případě je $O(enk^3)$, kde n je počet proměnných, k je mez velikostí domén, e je počet omezení

Algoritmus ARC-CONSISTENCY-3 (AC-3)

- Slidy R. Dechter, chapter 3, slide 10
- Algoritmus AC-1 je možné vylepšit
- Když se změní jen pár domén, není potřeba procházet všechna omezení znovu
- Stačí udržovat frontu dvojic proměnných, které ještě zpracovány nebyly nebo v doméně jedné z nich došlo ke změně
- Algoritmus označujeme ARC-CONSISTENCY-3 (AC-3), AC-2 byl mezikrok
- Složitost je $O(ek^3)$

Algoritmus ARC-CONSISTENCY-4 (AC-4)

- Slidy R. Dechter, chapter 3, slide 12
- Algoritmus AC-3 stále není optimální (z hlediska časové složitosti)
- Pouhé ověření konzistentnosti sítě potřebuje ek^2 operací, zajištění konzistentnosti asi nemůže jít rychleji
- Algoritmus AC-4 dosahuje tuto mez
- Nevyužívá revidující proceduru, zkoumá strukturu relace omezení
- Každé hodnotě a_i z domény x_i přiřadí počet konzistentních hodnot z x_j
- Hodnota a_i je odebrána z domény, pokud nemá v některé ze sousedních proměnných odpovídající protějšek
- Složitost je $O(ek^2)$

Složitost

- Místo k^2 vzniklé z univerzální relace můžeme uvažovat t pro maximální počet dvojic v relacích
- Revidující procedura potom má $O(t)$
- AC-1: $O(nket)$
- AC-3: $O(ekt)$
- AC-4: $O(et)$
- Složitost v nejlepším případě pro AC-1 a AC-3 je ek, když už problém je hranově konzistentní. Pro AC-4 je to ek^2, protože tolik trvá už inicializace
- Při slabých omezeních v praxi často AC-1 a AC-3 jsou rychlejší než AC-4

Konzistentnost cesty

- Definice: Pro danou síť $\mathcal{R} = \{X, D, C\}$ je množina dvou proměnných $\{x_i, x_j\}$ **konzistentní po cestě** (path-consistent) vzhledem k x_k právě tehdy, když pro každé přiřazení $(\langle x_i, a_i\rangle, \langle x_j, a_j\rangle)$ existuje hodnota $a_k \in D_k$ tž. přiřazení $(\langle x_i, a_i\rangle, \langle x_k, a_k\rangle)$ a $(\langle x_k, a_k\rangle, \langle x_j, a_j\rangle)$ jsou konzistentní.
- Definice (alternativa): Pro danou síť $\mathcal{R} = \{X, D, C\}$ je binární omezení R_{ij} je konzistentní cestou vzhledem k x_k x_j právě tehdy, když pro každý pár $(a_i, a_j) \in R_{ij}$, kde a_i, a_j jsou z příslušných domén, existuje hodnota $a_k \in D_k$ tž. $(a_i, a_k) \in R_{ik}$ a $(a_k, a_j) \in R_{kj}$.
- Definice: Podsíť nad třemi proměnnými $\{x_i, x_j, x_k\}$ je konzistentní po cestách jestliže pro každou permutaci (i, j, k) je R_{ij} konzistentní po cestě vzhledem k x_k
- Definice: Síť omezení je konzistentní po cestách, právě když pro všechny R_{ij} a každé $k \neq i, j$ je R_{ij} konzistentní po cestě vzhledem k x_k

Revidující procedura REVISE-3

- Slidy R. Dechter, chapter 3, slide 17
- Bere dvojice proměnných (x, y) a jejich omezení R_{xy} a třetí proměnnou z a vrací nejslabší omezení R'_{xy} splňující konzistentnost po cestě
- Testuje se každá dvojice konzistentních hodnot R_{xy}, jestli je možné ji rozšířit na hodnotu z. Když ne, dvojici smaže.
- Složitost je $O(k^3)$ nebo $O(tk)$

Algoritmus PATH-CONSISTENCY-1 (PC-1)

- Slidy R. Dechter, chapter 3, slide 18
- Vynutí konzistentnost po cestách pro všechny podsítě velikosti 3, výsledkem je síť konzistentní po cestách
- Je analogií AC-1
- Složitost je $O(n^5 k^5)$ nebo $O(n^5 t^2 k)$

Algoritmus PATH-CONSISTENCY-2 (PC-2)

- Slidy R. Dechter, chapter 3, slide 19
- Vylepšuje PC-1 udržováním fronty uspořádáných trojic ke zpracování
- Když je omezení $R_i j$ změněno vymazáním některé dvojice hodnot, všechny trojice zahrnující x_i a x_j a s nimi nějakou třetí hodnotu x_k jsou znovu zpracovány
- Je analogií AC-3
- Složitost je $O(n^3k^5)$ nebo $O(n^3t^2k)$
- Také není optimální z hlediska časové složitosti, existuje optimální algoritmus PC-4 se složitostí $O(n^3k^3)$ nebo $O(n^3tk)$

i-konzistentnost

- Definice: Pro síť $\mathcal{R} = (X, D, C)$ je relace $R_S \in C$, kde $|S| = i - 1$, **i-konzistentní** (i-consistent) vzhledem k proměnné $y \notin S$ právě tehdy, když pro každé $t \in R_S$ existuje hodnota $a \in D_y$ tž. (t, a).
- Definice: Síť je **i-konzistentní** (i-consistent) když pro jakoukoliv instanciaci $i - 1$ různých proměnných existuje instanciace každé $i$té proměnné tž. získaných i hodnot dohromady plňuje všechna omezení na těchto proměnných
- Revidující procedura REVISE-i – slidy R. Dechter, chapter 3, slide 23
- Algoritmus I-CONSISTENCY – slidy R. Dechter, chapter 3, slide 23
- Algoritmus je exponenciální vzhledem k i, tedy při zajišťování globální konzistentnosti (i je počet všech proměnných v síti) exponenciální vzhledem k velikosti sítě
- Zavádí do sítě omezení na $i - 1$ proměnných, z binární sítě tak může udělat nebinární

Zobecněná hranová konzistentnost

- Jedna ze dvou možností rozšíření hranové konzistentnosi na nebinární sítě
- Definice: Pro síť $\mathcal{R} = (X, D, C)$ s relací $R_S \in C$, je proměnná x **hranově konzistentní** vzhledem k R_S právě tehdy, když pro každé $a \in D_x$ existuje n-tice $t \in R_S$ tž. $t[x] = a$
- Definice: Omezení R_S je hranově konzistentní právě tehdy, když je hranově konzistentní vzhledem ke každé proměnné ve svém rozsahu
- Zajištění konzistentnosti redukuje doménu proměnné
- Druhý přístup je **relační hranová konzistentnost**, zaznamenává odvozené omezení ve formě změnou některé z relací reprezentujících omezení

Globální omezení

- Modelování reálných aplikací ve formě problému s omezeními vyžaduje specializované algoritmy šíření omezení pro často využívaná omezení
- alldifferent
  - Požaduje se, aby všechny proměnné měly přiřazeny vzájemně různé hodnoty
  - Je možné popsat binárními omezeními
  - Aplikování hranové konzistentnosti většinou nic nepřinese, protože síť už je konzistentní (pokud nejsou jednoprvkové domény)
- Typicky dávají smysl nad libovolným rozsahem
- Další příklady: omezení součtem (jedna proměnná je součtem jiných), kumulativní omezení (kumulativní spotřeba zdrojů v čase nepřekračuje specifikovanou kapacitu)

Konzistentnost hranic

- Na velkých doménách je "drahé" zajistit hranovou konzistentnost
- Levnější je zajistit **konzistentnost hranic** (bounds-consistency)
- Idea je omezit domény intervalem a zajistit, že krajní body těchto intervalů splňují hranovou konzistentnost
- Slidy R. Dechter, chapter 3, slide 31
- Zajištění konzistentnosti hranic při omezení typu alldifferent je možné v čase $O(n \log n)$