BIPARTITNÍ PÁROVÁNÍ

Petr Ryšavý

14. září 2020

Katedra počítačů, FEL, ČVUT

ÚVOD

Rumpelstiltskin principle (něm. Rumpelstilzchen)

Credits: Patrick Winston (MIT), Zdeněk Hanzálek,
`https://www.pohadky.org/index.php?co=pohadka&pohadka=322`,
`https://commons.wikimedia.org/wiki/File:Rumpelstiltskin-Crane1886.jpg`

Párování a základní algoritmus

Bipartitní graf

Definice 2.1 (Bipartitní graf) *Graf* G *je bipartitní, pokud lze jeho vrcholy rozdělit do dvou skupin* X *a* Y *tak, že každá hrana má jeden koncový vrchol v* X *a druhý v* Y*.*

Párování

Definice 2.2 (Párování) *Nechť G je neorientovaný graf se smyčkami. $P \subseteq E(G)$ se nazývá párování (anglicky matching) pokud žádné dvě různé hrany nesdílejí společný vrchol.*

Proč by nás párování mělo zajímat?

- Přiřazení lidí k pracem, strojů k úkolům, ...
- Problém tanečních
- Aproximace problému obchodního cestujícího (Christofidesův algoritmus)

Terminologie

Definice 2.3 (Nasycený vrchol) *Je-li P párování, pak v je* ***nasycený*** *(anglicky saturated) vrchol pokud existuje $e \in P$, že v je koncový vrchol e. Jinak je v* ***volný*** *(anglicky free) vrchol.*

Definice 2.4 (Dokonalé párování) ***Dokonalé párování*** *(anglicky perfect matching) je párování, kde jsou všechny vrcholy nasycené.*

Definice 2.5 (Maximální párování) ***Maximální párování*** *(anglicky maximal matching) P je takové s největším možným $|P|$.*

Střídavá a zlepšující cesta

Definice 2.6 (Střídavá cesta) *Nechť P je párování v G.* ***Střídavá*** *(anglicky alternating) cesta C vzhledem k P je cesta v G taková, že*

1. *obsahuje na střídačku hrany, které jsou z P a nejsou z P;*
2. *pokud koncový vrchol C je saturovaný v P, pak v C leží hrana, která ho saturuje.*

Pokud oba koncové vrcholy C jsou volné, pak C nazýváme zlepšující cestou vzhledem k P.

K čemu je dobrá zlepšující cesta

Tvrzení 2.7 *Pokud C je zlepšující cesta vzhledem k párování P, pak párování $P' = P \oplus C$ je také párováním.*[1]

[1] Symetrický rozdíl je definovaný jako $P \oplus C = (P \setminus C) \cup (C \setminus P)$.

Algoritmus (Edmonds's Blossom algorithm, Hopcroft-Karp)

$P \leftarrow$ prázdné párování
while existuje zlepšující cesta C **do**
    $P = P \oplus C$
end while

Existuje zlepšující cesta vždy?

Věta 2.8 (Berge) *Nechť P je párování v G a nechť*

$$|P| = |P_{max}| - k.$$

Pak existuje k zlepšujících cest vzhledem k P, jejichž množiny vrcholů jsou disjunktní. Alespoň jedna z nich má délku $\leq \frac{n}{k}$.

Důsledek Nechť G je neorientovaný graf a P párování v G. Pak P je maximální párování právě tehdy, když neexistuje žádná zlepšující cesta vzhledem k P.

HLEDÁNÍ ZLEPŠUJÍCÍCH CEST

Jak na hledání zlepšujících cest

- Zlepšující cesta začíná a končí ve volném vrcholu
- Začneme ve všech volných vrcholech
- Vpřed jdeme hranami z $E \setminus P$
- Zpět jdeme hranami z P
- Když skončíme ve volném vrcholu máme zlepšující cestu

Formulace jako BFS

- Přidáme nový vrchol s a vedeme hrany do všech volných vrcholů levé strany
- Nový vrchol t a hrany do něj z volných vrcholů pravé strany
- Hrany z párování orientujeme doleva, zbylé doprava
- Hledáme cestu z s do t
- V pomocném grafu existuje cesta z s do t právě tehdy, když existuje zlepšující cesta
- Pomocí BFS sestrojíme *alternating level graph*
- Skončíme při nálezu první cesty do t pomocí DFS nalezneme vrcholově disjunktní cesty

Příklad

Celý algoritmus

$P \leftarrow$ prázdné párování
while existuje zlepšující cesta **do**
    pomocí BFS sestroj *alternating level graph*
    pomocí DFS nalezni množinu disjunktních zlepšujících cest
    pro každou zlepšující cestu C nahraď $P \leftarrow P \oplus C$
end while

Čas běhu

Lemma *Po k iteracích musí být délka zlepšující cesty alespoň k.*

Společně s Bergeho větou to znamená, že čas běhu je $\mathcal{O}((m+n) \cdot \sqrt{n})$.

Souvislost s dalšími problémy

- Párování v obecných grafech
- Toky v sítích

Programování

- 11045 - My T-shirt suits me
- 10092 - The Problem with the Problem Setter
- 10080 - Gopher II

Reference

- Předmět Teorie grafů
  http://math.feld.cvut.cz/demlova/teaching/grafy/grafy_dom.html
- Robert Sedgewick and Kevin Wayne, Algorithms,
  http://algs4.cs.princeton.edu/home/, namely
  http://algs4.cs.princeton.edu/44sp/
- Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013). Competitive Programming 3. Lulu Independent Publish.
- https://www.youtube.com/watch?v=lM5eIpF0xjA
- https://www.youtube.com/watch?v=JpapV5DrBek

Děkuji za pozornost.

Čas na otázky!