

Najdaljša pot

1 Uvod

Za projektno nalogo pri predmetu finančni praktikum bova iskala najdaljše poti v grafih. Za razliko od problema iskanja najkrajše poti v grafih brez negativnih ciklov, ki je rešljiv v polinomskem času, pa je problem iskanja najdaljše poti NP-težak za splošne grafe. Za usmerjene aciklične grafe poznamo linearni algoritem, ki reši problem, kar je predvsem uporabno pri iskanju kritičnih poti.

Iskanje Hamiltonove poti v grafu G z n vozlišči lahko reduciramo na iskanje najdaljše poti, saj graf vsebuje Hamiltonovo pot, če je dolžina najdaljše poti grafa G enaka $n - 1$. Ker je iskanje Hamiltonove poti v grafu NP-težak problem, je posledično tudi iskanje najdaljše poti v grafu NP-težak problem. Če bi iskanje najdaljše poti bil polinomski problem, bi lahko poiskali najdaljšo pot v grafu in jo primerjali s številom vozlišč. Potemtakem bi bilo iskanje Hamiltonove poti polinomski problem, kar pa vemo da ni.

2 Usmerjeni aciklični grafi

V primeru usmerjenih acikličnih grafov poznamo več načinov, kako lahko v linearnem času poiščemo najdaljše poti. Eden izmed njih je, da za uteži povezav grafa G damo nasprotno vrednost cen povezav istega grafa. Tako dobimo novi graf $-G$, katerega najkrajša pot je najdaljša pot v grafu G . Ker je graf G usmerjen in acikličen, s tem nismo ustvarili negativnih ciklov in zato lahko uporabimo algoritme za iskanje najkrajših poti v grafih.

Drugi način je, da graf topološko uredimo, nato pa poiščemo najdaljše poti do vsakega vozlišča v grafu. To naredimo tako, da vozliščem dodamo maksimalno vrednost uteži povezav do njih oziroma najdaljše število povezav do njih, v primeru ko je graf neutežen. Če vozlišče nima vstopnih povezav mu dodelimo vrednost 0. Nato se vračamo od vozlišča z največjo vrednostjo proti začetku, pri čemer na vsakem koraku odštejemo ceno povezave oziroma 1 od predhodnega vozlišča. Pot po kateri pridemo do izhodišča topološko urejenega grafa, tako, da je končna vrednost enaka 0, je najdaljša pot v danem grafu.

3 Splošni grafi

Kot sva že omenila, je problem iskanja najdaljše poti v splošnem grafu NP-težak oziroma ni polinomski problem. Za drevesa oziroma gozdove pa je ta problem seveda enostaven. Zato nas za dani splošni graf G najprej zanima, kako podoben je drevesu (tree-like). V teoriji grafov temu pravimo **drevesno razkrajanje** (tree decomposition). To so torej vozlišča grafa G , ki so razporejena v strukturi, ki je podobna drevesu (namesto vozlišč ima vreče), za katero velja

1. če sta vozlišča u in v soseda, potem mora obstajati vreča (bag), ki ju vsebuje in

2. za vsako vozlišče v velja, da vreče katere ga vsebujejo tvorijo pod-drevo.

Posamezen graf ima lahko več drevesnih razkrojev. **Širina razkroja** je število, ki je enako velikosti največje vreče (številu vozlišč v vreči z največ vozlišči) minus 1.

Drevesna širina (treewidth) danega grafa G pa je število, ki je enako najmanjši širini razkroja po vseh razkrojih grafa G . Manjša kot je drevesna širina grafa, bolj je graf podoben drevesu. Najmanjša drevesna širina je 1 in pripada zgolj drevesom in gozdovom (da drevesa ne bi imela drevesne širine 2, v definiciji odštejemo 1).

Določevanje točne drevesne širine danemu grafu G je NP-težak problem; obstajajo pa algoritmi, ki lahko določijo ali je drevesna širina grafa G navzgor omejena z danim številom k v času $O(n^k)$. Poznamo tudi nekaj družin grafov, za katere vemo, da imajo omejeno drevesno širino. To so na primer kaktusi (cactus graphs), psevdogozdovi (pseudoforests), izven-planarni grafi (outerplanar graphs) in Halinovi grafi (Halin graphs). Na grafih z omejeno drevesno širino pa za številčne, sicer NP-težke, probleme poznamo učinkovite algoritme. Avtorja upava, da bova na grafih z omejeno drevesno širino lahko našla učinkovit algoritem za iskanje najdaljše poti. Najbrž si bova pomagala tudi z barvanjem vozlišč grafa.

4 Eksperiment

Za konec bova naredila simulacijo na različnih grafih. Sprva bova simulacijo naredila na usmerjenih acikličnih grafih. Na njih bova preizkušala različne algoritme (za iskanje kritične poti, za splošne grafe itd.) in primerjala čase, ki ga posamezen algoritem potrebuje za izračun najdaljše poti. Nato pa bova še izvedla simulacijo na splošnih grafih in primerjala čase v odvisnosti od števila vozlišč in povezav.