FOCS: Annual IEEE Symposium on Foundations of Computer Science 2023

Stare, Matic   Starič, Martin

April 1, 2024

Contents

Abstract

abstract

1 Introduction

2 A Randomized Algorithm for Single-Source Shortest Path on Undirected Real-Weighted Graphs

Problem najkrajše poti je eden izmed najbolj znanih problemov v teoriji grafov. Cilj je, da najdemo najkrajšo pot med dvema vozliščema na podlagi uteženih grafov. Za reševanje tovrstnih problemov se je v preteklosti najbolj uveljavil Dijkstrin algoritem, ki deluje s časovno zahtevnostjo $O(V^2)$, kjer je V število vozlišč. To mejo so kasneje izboljšali s Fibonaccijevo kopico na $O(V\log V + E)$, kjer je E število povezav. V tem članku so se avtorji osredotočili na problem najkrajše poti iz enega vira (SSSP) na uteženih grafih. Zasnovali so nov algoritem, ki deluje s časovno zahtevnostjo $O(E\sqrt{\log V * \log\log V})$, vendar je pogojen z verjetnostjo. To pomeni, da je časovna zahtevnost algoritma pričakovana in ne zahtevnost v najslabšem primeru.

Do nedavnega je bilo ozko grlo pri iskanju SSSP vrsta s prednostjo. Ideja izboljšave je v tem, da v vrsto vnesemo manjše število vozlišč. To dosežemo z tehniko *Bundle Construction*, ki deluje na sledeč način:

- Iz množice vozlišč $V \setminus \{s\}$ naključno izberemo $R \subseteq V \setminus \{s\}$ z verjetnostjo $\frac{1}{k}$. Na koncu množici R dodamo vozlišče s.
- Na vsakem vozlišču $v \notin R$ poženemo Dijkstrin algoritem z začetkom v v dokler iz kopice ne izločimo prvega vozlišča iz R, ki je označen z $b(v)$. Ker sta v in $b(v)$ v neposredni bližini, pravimo, da je v *bundled* v $b(v)$.
- Definirajmo $u \in R$. $Bundle(u)$ je množica vseh vozlišč, ki so bundled v u. Na ta način unija vseh $Bundle(u_i)$ tvori particijo V.
- skip

Časovna zahtevnost algoritma $O(mk\log k)$. Z njegovo pomočjo smo pridobili 3 nove množice s pričakovanimi velikostmi:

- $E[|R|] = O(\frac{m}{k})$.
- $E[|S_v|] = \Theta(k)$.
- $E[|Ball(v)|] = \Omega(k)$.

Bundle Construction so avtorji uporabili kot osnovo za nov algoritem. Ta deluje na sledeč način: Bundle Dijkstra - Ustvarimo tabelo, kjer bomo hranili razdalje vozlišč v od vozlišča s, in v Fibonaccijevo kopico vstavimo vsa vozlišča iz R.

1. Ko iz kopice izločimo vozlišče u, posodobimo tabelo tako da za vsako vozlišče, ki je zapakirano u-ju, najdemo točno razdaljo. To storimo tako, da izberemo minimum izrazov $d(v)$, $d(u) + dist(u, v)$, $d(y) + dist(y, v)$ in $d(z_1) + w_{z1z2} + dist(z_2, v)$
2. Ko je vsako vozlišče x iz Bundle(u) posodobljeno pa posodobimo še sosednja vozlišča $y \in N(x)$ in vozlišča, ki so znotraj Ball(y).
3. Ko posodobimo vozlišče, ki ni v množici R, posodobimo še njegovo zapakirano vozlišče b(v) z d(v) + dist(v,b(v)).