FINANČNI PRAKTIKU

Uravnotežen rdeče-modri povezan podgraf

Matej Rojec, Ana Marija Belingar, Vito Rozman

1 Predstavitev problema

Naj bo $G = (V, E)$ graf. Vsako vozlišče $v \in V$ je obarvano rdeče ali modro. Najti želimo največji povezani podgraf $G' = (V', E')$, ki ima enako število rdečih in modrih vozlišč. Velikost podgrafa je število njegovih vozlišč. Ta problem je v splošnem NP-težek, kar pomeni da ga ne moremo rešiti v polinomskem času. Osredotočili se bomo na optimalen algoritem za reševanje problema na mrežah oblike $1 \times n$ (pot), $2 \times n$, $3 \times n$ in $4 \times n$. Naš algoritem bomo testirali na grafih, kjer bomo vožlišča obarvali rdeče z verjetnostjo $p \in (0, 1)$ in modro z verjetnostjo $1 - p$.

2 Osnovni pojmi

Definicija 1. (Induciran podgraf) Naj bo $G = (V, E)$ graf in naj bo $S \subseteq V$ podmnožica vozlišč grafa G. Graf $G[S]$ je induciran podgraf grafa G, natanko takrat ko $\forall u, v \in S$ velja, da sta u in v sosednja v $G[S]$, natanko takrat ko sta sosednja v G.

V našem primeru bomo iskali največji povezan inducirani podgraf $G' = (V', E')$ grafa $G = (V, E)$, kjer lahko možico volzlišč zapišemo kot:

$$V' = V_R \cup V_B,$$

za katero velja $V_R \cap V_B = \emptyset$ in $|V_R| = |V_B| = \frac{|V'|}{2}$. Tako bomo dobili največji uravnotežen povezan podgraf z enakim številom rdečih in modrih vozlišč.

Definicija 2. (Pot) Graf $P_n = (V, E)$ je pot, kjer je $V = \{v_1, ..., v_n\}$, če je množica povezav oblike: $E = \{(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)\}$.

Definicija 3. (Mreža) Graf $M_{m,n} = (V, E)$ je mreža velikosti $m \times n$, kjer je $V = \{v_{1,1}, ..., v_{1,n}, v_{2,1}, ..., v_{2,n}, ..., v_{m,1}, ..., v_{m,n}\}$, če je množica povezav oblike:

$$\begin{aligned} E = \{&(v_{1,1}, v_{1,2}), (v_{1,2}, v_{1,3}), ..., (v_{1,n-1}, v_{1,n}), \\ &(v_{2,1}, v_{2,2}), (v_{2,2}, v_{2,3}), ..., (v_{2,n-1}, v_{2,n}), \\ &(v_{m,1}, v_{m,2}), (v_{m,2}, v_{m,3}), ..., (v_{m,n-1}, v_{1,n}), \\ &(v_{1,1}, v_{2,1}), (v_{2,1}, v_{3,1}), ..., (v_{m-1,1}, v_{m,1}), \\ &(v_{1,2}, v_{2,2}), (v_{2,2}, v_{3,2}), ..., (v_{m-1,2}, v_{m,2}), \\ &(v_{1,n}, v_{2,n}), (v_{2,n}, v_{3,n}), ..., (v_{m-1,n}, v_{m,n})\}. \end{aligned}$$

3 Ideje

Najprej si bomo ogledali problem na poteh $(1 \times n)$, pri čemer bomo uporabili dinamično programiranje kot je razvidno iz algoritma 1 . Algoritem sprejme kot vhod pot dolžine n, kjer je vsako vozlišče obarvano bodisi modro (B) bodisi rdeče (R). Najprej barvam dodelimo uteži u_i, modrim vozliščem dodelimo utež 1, rdečim pa utež -1. Potem na vsakem koraku izračunamo delno vsoto s_i, ki pove razliko med številom modrih in rdečih vozlišč od prvega do $i-$ tega vozlišča. Nato izračunamo še $t_{j,i}$, ki pove razliko med številom modrih in rdečih vozlišč od $j-$tega do $i-$tega. Kandidati so tisti $t_{j,i}$ za katere je $t_{j,i} = 0$. Med temi vzamemo največje podpoti $P_{l,k}$, pri katerih je razlika $l - k$ maksimalna.

Algoritem 1 BCS za poti .

Vhod: Pot P_n, , barve vozlišč (B/R).
Izhod: Najdaljša utežena (povezana) podpot $P_{l,k}$.

```
function BSCP(P_n)
    for i ← 1 to n do
        if Barva(v_i) = B then
            u_i ← 1
        else
            u_i ← −1
        end if
    end for
    s_0 ← 0
    for i ← 1 to n do
        s_i ← s_{i−1} + u_i
        for j ← 1 to i do
            t_{j,i} = s_i − s_{j−1}
        end for
    end for
    len ← max{i − j | t_{j,i} = 0}
    P_{l,k} ← {{v_l, ..., v_k} | (t_{l,k} ← 0) and k − l = len }
    return P_{l,k}
end function
```

Načrt za naprej:

- Posplošitev algoritma na mreže
- Implementacija algoritma
- Eksprerimenti