

Enajste vaje APS2: Barvanje grafov

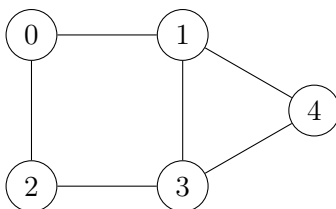
1 Opis problema

Podan je neusmerjen graf. Njegova vozlišča želimo pobarvati tako, da sosednji vozlišči ne bosta iste barve. Najmanjšemu številu barv, ki ga potrebujemo za to opravilo, rečemo *kromatsko število* grafa. Pri problemu barvanja grafov nas za podani graf G torej zanima njegovo kromatsko število, ki ga označimo s $\chi(G)$.

Problem je v splošnem NP-težak, zato zanj po vsej verjetnosti ni učinkovite rešitve. Če se ga lotimo naivno, v najslabšem primeru potrebujemo $O(mn^n)$ časa, kjer je n število vozlišč, m pa število povezav. Število možnih barvanj vozlišč s k barvami znaša k^n , za preverjanje veljavnosti barvanja pa moramo preveriti vse povezave. Kromatsko število je v najslabšem primeru enako številu vozlišč.

2 Ideja

Naj $G[U]$ označuje podgraf grafa G , ki vsebuje vozlišča $U \subseteq V$ in vse povezave $(u, v) \in E$, za katere velja $u \in U$ in $v \in U$. Na primer, za graf na sliki 1 je graf $G[\{1, 2, 4\}]$ sestavljen iz vozlišč 1, 2 in 4 ter povezave (1, 4).



Slika 1: Primer grafa.

Spomnimo se tudi, da je *neodvisna množica* grafa podmnožica množice vozlišč, tako da noben par vozlišč v njej ni med seboj povezan. Na primer, ena od neodvisnih množic grafa na sliki 1 je množica $\{2, 4\}$.

Kromatsko število lahko določimo nekoliko učinkoviteje, če upoštevamo, da lahko vozlišča v neodvisni množici grafa pobarvamo z isto barvo. Če iz grafa G odstranimo neodvisno množico I in ugotovimo, da lahko preostanek grafa $(G[V \setminus I])$ pobarvamo s k barvami, potem vemo, da lahko graf G pobarvamo s $k + 1$ barvami. Seveda moramo izbrati takšno neodvisno množico I , da bo vsota $1 + \chi(G[V \setminus I])$ minimalna:

$$\chi(G) = \begin{cases} 0, & \text{če } G \text{ nima vozlišč;} \\ 1 + \min_{I \in \mathcal{I}(G)} \chi(G[V \setminus I]) & \text{sicer,} \end{cases}$$

pri čemer je $\mathcal{I}(G)$ množica vseh neodvisnih množic grafa G .

Na primer, graf na sliki 1 ima devet neodvisnih množic: $\{0\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{0, 3\}$, $\{0, 4\}$, $\{1, 2\}$, $\{2, 4\}$. Če iz grafa odstranimo množico $\{0\}$, potem lahko preostanek grafa pobarvamo s tremi barvami. Vendar pa to ne pomeni, da je kromatsko število grafa enako $3 + 1 = 4$; če odstranimo, na primer, množico $\{2, 4\}$, lahko preostanek grafa (torej graf z vozlišči 0, 1 in 3 ter povezavama $(0, 1)$ in $(1, 3)$) pobarvamo z dvema barvama. Izkaže se, da je to optimalna izbira. Kromatsko število grafa je torej enako 3.

3 Algoritem

Najprej se moramo dogovoriti, kako učinkovito predstavljamo podmnožice množice vozlišč (zaradi enostavnosti bomo predpostavili, da so vozlišča grafa označena z $0, 1, \dots, n - 1$). Saj že vemo: množico $\{a_1, a_2, \dots, a_k\}$ predstavimo z n -bitnim številom, v katerem so biti na indeksih $a_1, a_2, \dots, a_k$ enaki 1, ostali pa 0. Na primer, če je $n = 5$, bomo množico $\{0, 2, 3\}$ predstavili z dvojiškim številom 01101 (kot ponavadi je bit z indeksom 0 najlažji, torej skrajno desni bit).

Druga lastnost, ki jo opazimo, je tale: pri računanju kromatskega števila nekega grafa bomo uporabljali kromatska števila njegovih podgrafov, pri čemer lahko isti podgraf nastopa kot sestavni del več različnih večjih grafov. To pomeni, da imamo prekrivajoče se podprobleme, zato se problema lotimo z dinamičnim programiranjem. Kot vemo, pri dinamičnem programiranju potrebujemo shrambo že izračunanih rezultatov, ki jo lahko polnimo bodisi »od zgoraj navzdol« (z rekurzijo in memoizacijo) ali »od spodaj navzgor« (iterativno). Omenjena shramba bo v našem primeru kar tabela velikosti 2^n (število podmnožic množice $\{0, \dots, n - 1\}$), v kateri bo element na indeksu i hranil že izračunano kromatsko število grafa, določenega z množico vozlišč, ki ga predstavlja število i . V primeru na sliki 1 bo kromatsko število grafa na množici vozlišč $\{0, 2, 3\}$ shranjeno na indeksu $01101_{(2)} = 13_{(10)}$.

Pri iterativnem polnjenju shrambe že izračunanih rezultatov moramo paziti na pravi vrstni red računanja: ko računamo kromatsko število grafa, določenega z množico vozlišč $U \subseteq V$, morajo biti kromatska števila za vse njegove podgrafe (torej za vse podmnožice množice U) že izračunana. To lahko dosežemo preprosto tako, da tabelo polnimo po naraščajočih indeksih, saj imajo vse podmnožice množice U manjši indeks kot množica U .

4 Primer

Slika 2 prikazuje delovanje algoritma za določanje kromatskega števila na grafu s slike 1. Pripravimo si tabelo T z 2^5 elementi in jo polnimo po naraščajočih indeksih; spomnimo se, da v element na indeksu i vpišemo kromatsko število grafa na množici vozlišč, ki jo predstavlja indeks i . Nekaj izračunov si ogledimo podrobneje:

- Element $T[0]$ se nanaša na graf brez vozlišč. Njegovo kromatsko število je enako 0.
- Element $T[1]$ se nanaša na graf na vozlišču $\{0\}$. Ta ima eno samo neprazno neodvisno množico — množico $\{0\}$. Kromatsko število grafa potemtakem izračunamo takole:

$$\begin{aligned} T[1] &= \chi(G[\{0\}]) \\ &= 1 + \min\{\chi(G[\{0\} \setminus \{0\}])\} \\ &= 1 + \min\{\chi(G[\emptyset])\} \\ &= 1 + \min\{T[0]\} \\ &= 1. \end{aligned}$$

- Element $T[3]$ se nanaša na graf na vozliščih $\{0, 1\}$, ki ima dve neprazni neodvisni množici: $\{0\}$ in $\{1\}$.

$$\begin{aligned}
T[3] &= \chi(G[\{0, 1\}]) \\
&= 1 + \min\{\chi(G[\{0, 1\} \setminus \{0\}]), \chi(G[\{0, 1\} \setminus \{1\}])\} \\
&= 1 + \min\{\chi(G[\{1\}]), \chi(G[\{0\}])\} \\
&= 1 + \min\{T[2], T[1]\} \\
&= 2.
\end{aligned}$$

- Graf na množici $\{0, 3\}$ ima tri neprazne neodvisne množice: $\{0\}$, $\{3\}$ in $\{0, 3\}$.

$$\begin{aligned}
T[9] &= \chi(G[\{0, 3\}]) \\
&= 1 + \min\{\chi(G[\{0, 3\} \setminus \{0\}]), \chi(G[\{0, 3\} \setminus \{3\}]), \chi(G[\{0, 3\} \setminus \{0, 3\}])\} \\
&= 1 + \min\{\chi(G[\{3\}]), \chi(G[\{0\}]), \chi(G[\emptyset])\} \\
&= 1 + \min\{T[8], T[1], T[0]\} \\
&= 1.
\end{aligned}$$

- Celoten graf (tj. graf na množici $\{0, 1, 2, 3, 4\}$) ima devet nepraznih neodvisnih množic: $\{0\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{0, 3\}$, $\{0, 4\}$, $\{1, 2\}$ in $\{2, 4\}$.

$$\begin{aligned}
T[31] &= \chi(G[\{0, 1, 2, 3, 4\}]) \\
&= 1 + \min\{\chi(G[\{0, 1, 2, 3, 4\} \setminus \{0\}]), \chi(G[\{0, 1, 2, 3, 4\} \setminus \{1\}]), \\
&\quad \chi(G[\{0, 1, 2, 3, 4\} \setminus \{2\}]), \chi(G[\{0, 1, 2, 3, 4\} \setminus \{3\}]), \\
&\quad \chi(G[\{0, 1, 2, 3, 4\} \setminus \{4\}]), \chi(G[\{0, 1, 2, 3, 4\} \setminus \{0, 3\}]), \\
&\quad \chi(G[\{0, 1, 2, 3, 4\} \setminus \{0, 4\}]), \chi(G[\{0, 1, 2, 3, 4\} \setminus \{1, 2\}]), \\
&\quad \chi(G[\{0, 1, 2, 3, 4\} \setminus \{2, 4\}])\} \\
&= 1 + \min\{\chi(G[\{1, 2, 3, 4\}]), \chi(G[\{0, 2, 3, 4\}]), \chi(G[\{0, 1, 3, 4\}]), \\
&\quad \chi(G[\{0, 1, 2, 4\}]), \chi(G[\{0, 1, 2, 3\}]), \chi(G[\{1, 2, 4\}]), \\
&\quad \chi(G[\{1, 2, 3\}]), \chi(G[\{0, 3, 4\}]), \chi(G[\{0, 1, 3\}])\} \\
&= 1 + \min\{T[30], T[29], T[27], T[23], T[15], T[22], T[14], T[25], T[11]\} \\
&= 3
\end{aligned}$$

5 Časovna zahtevnost

Za vsako od 2^n podmnožic U množice V in za vsako neodvisno množico $I \subseteq U$ moramo v izračunati množico $U \setminus I$ in jo uporabiti kot indeks v tabelo. Če neodvisne množice grafa poiščemo vnaprej (to lahko storimo z naivnim postopkom) in če za delo z množicami uporabljamo učinkovite operacije, lahko časovno zahtevnost algoritma ocenimo kot $O(2^n 2^n) = O(4^n)$. Obstajajo tudi učinkovitejši algoritmi, a zaenkrat vsi tečejo v času $\Omega(2^n)$.

6 2-obarvljivost grafov

Če nas zanima, ali lahko graf pobarvamo z dvema barvama, lahko dobimo odgovor v času $O(m)$, in sicer s preprostim iskanjem v globino ali širino. Premislite sami!

0	1	2	3	4	5	6	7
0	1	1	2	1	2	1	2
{}	{0}	{1}	{0, 1}	{2}	{0, 2}	{1, 2}	{0, 1, 2}
8	9	10	11	12	13	14	15
1	1	2	2	2	2	2	2
{3}	{0, 3}	{1, 3}	{0, 1, 3}	{2, 3}	{0, 2, 3}	{1, 2, 3}	{0, 1, 2, 3}
16	17	18	19	20	21	22	23
1	1	2	2	1	2	2	2
{4}	{0, 4}	{1, 4}	{0, 1, 4}	{2, 4}	{0, 2, 4}	{1, 2, 4}	{0, 1, 2, 4}
24	25	26	27	28	29	30	31
2	2	3	3	2	2	3	3
{3, 4}	{0, 3, 4}	{1, 3, 4}	{0, 1, 3, 4}	{2, 3, 4}	{0, 2, 3, 4}	{1, 2, 3, 4}	{0, 1, 2, 3, 4}

Slika 2: Delovanje algoritma za določanje kromatskega števila na grafu s slike 1.