UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET

Algoritmi bojenja čvorova u grafovima

- master rad -

Mentor:
dr Boris Šobot

Student:
Čisar Angela

Novi Sad, 2016.

Predgovor

Za temu master rada uzela sam algoritme bojenja čvorova u grafovima iz oblasti teorije algoritama. Algoritmi danas imaju veliki značaj u matematici i informatici za rešavanje problema u oblastima vezanim za računare, odnosno programiranje kao što su telekomunikacije, računarske mreže, internet i baze podataka. Bojenje čvorova grafova je značajno zbog brojnih primena u praksi u računarstvu, u hemiji, biohemiji, u sociologiji i raznih oblasti modeliranja problema.

Rad se sastoji od 4 poglavlja. Prva glava predstavlja kratak uvod u grafove, osnovne osobine i definicije obojivosti i hromatskog broja. Pomenut je i problem 4-obojivosti planarnih grafova. Druga glava sadrži pregled neophodnog predznanja iz teorije algoritama, uz uvođenje klasa složenosti P i NP, kao i klase NP-kompletnih problema. U trećoj glavi prikazano je svođenje problema k-obojivosti na problem 3-SAT, čime se dokazuje da je problem k-obojivosti NP-kompletan. Četvrta glava sadrži prikaz nekoliko algoritama za nalaženje hromatskog broja datog grafa: svođenje na r-nezavisne skupove i svođenje na set-covering problem sa pomoćnim algoritmima za njihovo nalaženje.

Novi Sad, 2016. Čisar Angela

Sadržaj

1 Uvod u teoriju grafova 4

1.1 Bojenje čvorova 6

1.2 Problem 4-obojivosti planarnih grafova 8

2 Uvod u teoriju algoritama 11

3 NP kompletnost problema k-obojivosti 17

4 Algoritmi za nalaženje hromatskog broja 20

4.1 Algoritam za određivanje 2-obojivosti 20

4.2 Rekurentna relacija 23

4.3 Algoritam za određivanje svih maksimalnih nezavisnih podskupova 24

4.4 Algoritam svođenjem na r-nezavisne skupove 30

4.5 Set-covering problem 33

Literatura 38

Biografija 39

1 Uvod u teoriju grafova

Teorija grafova je samostalan deo matematike, koji se bavi istraživanjem grafova i njihovih osobina. Pomoću grafova možemo modelovati složene probleme veoma jednostavno. Važna je upotreba grafova za opis modela i struktura podataka u računarstvu, kod mobilnih mreža, u biohemiji i za rešavanje brojnih problema.

Definišemo osnovne pojmove:

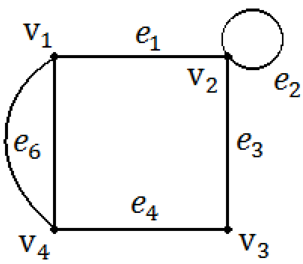
Definicija 1.1 Graf G je uređeni par G=(V,E) koji se sastoji od konačnog nepraznog skupa čvorova V(G) i konačnog skupa grana E(G).

Definicija 1.2 Dva čvora x,y su *susedna* ako su povezana granom e, pišemo e=(x,y). Čvorovi grane se nazivaju *krajevi*. *Skup svih suseda* čvora v u grafu G označavamo sa $N_G(v)$ – skup čvororva grafa G koji su povezani sa v.

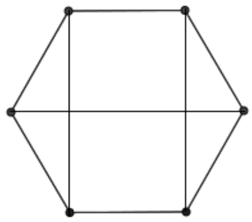
Definicija 1.3 Grana koja spaja čvor sa samim sobom naziva se *petljom*.

Definicija 1.4 *Paralelne grane* povezuju iste čvorove, odnosno imaju istu zajedničku početnu i krajnju tačku.

Definicija 1.5 Graf koji nema petlje ni paralelne grane naziva se *prostim grafom*. *Multigrafom* nazivamo grafove u kojima mogu pojavljivati paralelne grane ili petlje (slika 1.1).



Slika 1.1


Slika 1.2

Definicija 1.6 Stepen čvora v $\in$V(G) je broj suseda čvora, i to zapisujemo sa $d_G(V)=|N_G(v)|$. Minimalan stepen svih čvorova obeležavamo sa δ(G), a maksimalan sa Δ(G).

Definicija 1.7 Graf G nazivamo regularnim grafom ako svaki čvor u grafu ima isti broj suseda. Tada važi: δ(G)= Δ(G). Primer na slici 1.2 je 3-regularan graf sa 6 čvorova.

Definicija 1.8 *Kompletan graf* K_n je graf sa n čvorova u kom su svaka dva čvora povezana granom. *Prazan graf* $\overline{K_n}$ je graf sa n čvorova u kom nikoja dva čvora nisu povezana.

Definicija 1.9 *Šetnja* je niz čvorova koje su međusobno povezane:
$W = v_0\, v_1\, v_2 \ldots v_k,\ v_i \in V(G),\ e_i = v_{i-1}\, v_i \in E(G)$

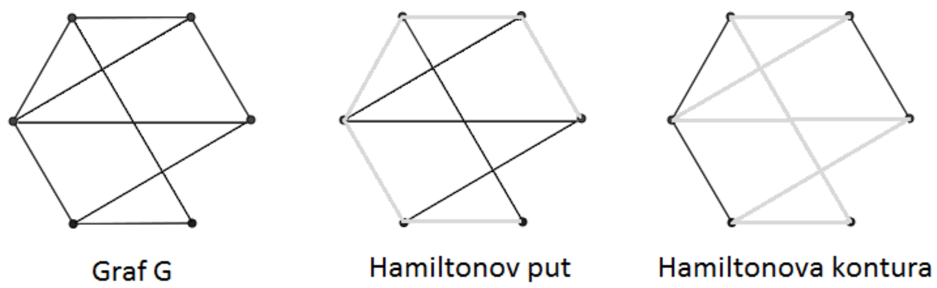
Definicija 1.10 *Staza* je šetnja kod kojeg kroz svake grane pređemo samo jedanput:
$W = v_0\, e_1\, v_1\, e_2\, v_2\, e_2 \ldots e_2\, v_k = v_0\, v_1\, v_2 \ldots v_k$
$v_i \in V(G),\ e_i = v_{i-1}\, v_i \in E(G),\ e_i \neq e_j,\ i \neq j$

Definicija 1.11 *Put* P je staza kod kojeg kroz svakog čvora pređemo samo jedanput:
$P_{k+1} = v_0\, e_1\, v_1\, e_2\, v_2\, e_2 \ldots e_2\, v_k = v_0\, v_1\, v_2 \ldots v_k$
$v_i \in V(G),\ e_i = v_{i-1}\, v_i \in E(G),\ v_i \neq v_j,\ i \neq j$

Definicija 1.12 *Kontura* C je zatvoreni put:
$C_{k+1} = v_0\, e_1\, v_1\, e_2\, v_2\, e_2 \ldots e_2\, v_k\, e_{k+1}\, v_0 = v_0\, v_1\, v_2 \ldots v_k\, v_0$
$v_i \in V(G),\ e_i = v_{i-1}\, v_i \in E(G),\ e_i \neq e_j,\ i \neq j$.

Definicija 1.13 *Ojlerov put* je staza W za koji važi: E(W)=E(G).
Ojlerova kontura je zatvorena staza W za koji važi: E(W)=E(G).

Definicija 1.14 *Hamiltonov put* P je put za koji važi: V(P)=V(G).
Hamiltonova kontura C je kontura za koji važi: V(C)=V(G). (Slika 1.3)



Slika 1.3

Definicija 1.15 Čvorovi u i v su *povezani* akko postoji (u-v)-put u G.

Definicija 1.16 *K-partitan graf* $G(X_1, X_2, \ldots, X_k)$ je graf gde su čvorovi podeljeni u k klasa $X_1, X_2, \ldots, X_k$, a grane postoje isključivo između čvorova iz različite klase.
2-partitni grafovi se zovu bipartitni grafovi.

Definicija 1.17 *Komponenta povezanosti grafa G* je klasa ekvivalencije u odnosu na povezanost. Graf je *povezan* ako su svaka dva čvora povezana, tada graf ima jednu komponentu povezanosti.

Definicija 1.18 Neka je dat graf G=(V,E). *Nezavisan skup čvorova* je skup čvorova grafa G gde ne postoje dva čvora koji su povezani, odnosno podgraf od G generisan čvorovima nezavisnog skupa je prazan graf. Nezavisan skup je *maksimalan*, ako ne postoji nezavisan skup koji ga sadrži. *Najveći* nezavisan skup je *maksimalan* nezavisan skup koji sadrži najveći broj čvorova, odnosno najveći po kardinalnosti.

Teorema 1.19 Ako graf nije povezan onda se on može razbiti na povezane komponente, disjunktne po čvorovima.

Dokaz: U [2], strana 10.

Definicija 1.20 *Orijentisani graf* ili *digraf* je graf G=(V,E) gde su grane usmereni od jednog do drugog čvora. Tada je $E \subset V \times V$, i grana $e = (v_1,v_2)$, gde je v_1 početni čvor, a v_2 krajnji.

1.1 Bojenje čvorova

Bojenje čvorova predstavlja početni problem izučavanja ove oblasti i svi ostali problemi bojenja se mogu svesti na njega. Priroda problema zavisi isključivo od broja boja, a ne od konkretne boje.

Definicija 1.21 *Bojenje* čvorova je pridruživanje po jedne boje svakom čvoru grafa G, odnosno to je funkcija f: V(G) → C, gde su elementi skupa C boja: $C=\{c_1, \dots, c_k\}$.

Definicija 1.22 Bojenje se zove *pravilno bojenje* ako ne postoje dva susedna čvora koja su jednako obojena. Dakle, pravilno bojenje je funkcija f: V(G) → C takva da

$$uv \in E(G) \Longrightarrow f(u) \neq f(v).$$

Definicija 1.23 *K-bojenje* je bojenje čvorova sa najviše *k* boja, odnosno kada je $|C|=k$.

Kažemo da je graf G *k-obojiv* ako postoji bojenje grafa G sa k ili manje boja, odnosno postoji r-bojenje grafa G, za $r \leq k$.

Definicija 1.24 *Hromatski broj* grafa G je najmanji prirodan broj *k* tako da je G k-obojiv u oznaci $\chi(G)$. Kažemo da je graf G *k-hromatski* ako je $\chi(G)=k$.

Skup čvorova kojima je dodeljena ista boja se naziva *klasa* te boje, svaka takva klasa formira nezavisni skup. Prema tome, k-bojenje je isto što i podela čvorova grafa na nezavisne skupove čvorova iste boje, i stoga termini k-partitan i k-obojiv imaju isto značenje.

Pored bojenja čvorova postoji i bojenje grana. U ovom radu to nećemo detaljnije razmatrati, samo navodimo definicije.

Definicija 1.25 Bojenje grana pridruživanje po jedne boje svakoj grani grafa G, odnosno to je funkcija f: E(G) → C, gde su elementi skupa C boja. $C=\{c_1, \dots, c_k\}$.

Slično kao kod bojenja čvorova, bojenje grana se zove *pravilno bojenje* ako ne postoje dve susedne grane koje su jednako obojena. Neka su $e_1,e_2 \in E(G)$. Označimo sa $e_1 \sim e_2$ ako e_1, e_2 imaju zajednički čvor. Tada je pravilno bojenje grana funkcija f:E(G) → C, za koju važi:

$$e_1 \sim e_2 \in E(G) \Longrightarrow f(e_1) \neq f(e_2).$$

Definicija 1.26 *K-bojenje* je bojenje grana sa najviše *k* boja, odnosno kada je $|C|$=k.

Slično kao kod bojenje čvorova, kažemo da je graf G *k-obojiv* ako postoji bojenje grafa G sa k ili manje boja, odnosno postoji r-bojenje grafa G, za r ≤ k.

Definicija 1.27 *Hromatski indeks* grafa G je najmanji prirodan broj *k* tako da je G granski k-obojiv u oznaci $\chi_1(G)$. Kažemo da je graf G *granski k-hromatski* ako je $\chi_1(G)$=k.

U daljem radu bavimo se samo sa bojenjem čvorova, i sledeće teoreme su vezane za bojenje čvorova grafa.

Sledeća tvrđenja su direktna posledica definicije hromatskog broja ([2], strana 120.):

Teorema 1.28 $\chi(K_n) = n$.

Teorema 1.29 *Ako je* $H \subset G$, *tada je* $\chi(G) \geq \chi(H)$.

Teorema 1.30 *Ako je* $K_n \subset G$, *tada je* $\chi(G) \geq n$.

Teorema 1.31 *Ako su* $G_1, \ldots, G_s$ *komponente grafa G, tada je* $\chi(G) = \max_{1\leq i\leq s}\chi(G_i)$.

Teorema 1.32 $\chi(G) = 1$ *ako i samo ako je G prazan graf.*

Teorema 1.33 $\chi(G) = 2$ *ako i samo ako je G netrivijalan bipartitan graf.*

Dokaz ($\Longrightarrow$) Neka važi $\chi(G) = 2$, i neka je C skup boja: C= $\{1, 2\}$. Definišemo skup A koji sadrži čvorove obojene bojom 1, i slično skup B koji sadrži čvorove obojene bojom 2:
A=$\{v \in V(G), f(v) = 1\}$
B=$\{v \in V(G), f(v) = 2\}$.
Ako grana xy ∈ E(G), tada važi: x ∈ A, y ∈ B ili x ∈ B, y ∈ A. Dakle, graf je bipartitan: G=G(A,B).

($\Longleftarrow$) Neka je G=(A,B) i neka $E(G) \neq \emptyset$.
Tada važi $\chi(G) > 1$. Definišemo preslikavanje:

$$f(v)=\begin{cases} 1, & \text{za } v\in A \\ 2, & \text{za } v\in B. \end{cases}$$

Iz toga sledi $\chi(G) = 2$. ∎

Lema 1.34 *Graf je bipartitan ako i samo ako nema neparnih kontura.*

Dokaz U [2], strana 11.

Teorema 1.35 $\chi(G) \geq 3$ *ako i samo ako G sadrži neparnu konturu.*

<u>Dokaz</u> ($\Longrightarrow$) Neka važi $\chi(G) \geq 3$. Na osnovu teoreme 1.31 i 1.32 znamo da G nije prazan i nije bipartitan. Dalje iz leme 1.33 sledi da G sadrži neparnu konturu.

($\Longleftarrow$) Pretpostavimo da G sadrži neparnu konturu C. Iz teoreme 1.28 sledi

$\chi(G) \geq \chi(H)$, a pošto $\chi(C)=3$ imamo da je $\chi(G) \geq 3$. ■

Teorema 1.36 *Za svaki graf G važi nejednakost:* $\chi(G) \leq \Delta(G) + 1$.

<u>Dokaz</u> U [2], strana 121.

Teorema 1.37 (Mycielski, 1955) *Za svaki prirodan broj k* ($k \geq 1$) *postoji k-hromatski graf koji ne sadrži* K_3 *kao podgraf.*

<u>Dokaz</u> U [2], strana 127.

1.2 Problem 4-obojivosti planarnih grafova

Definicija 1.38 Graf G=(V,E) je *planaran* akko se može smestiti u ravan tako da grane nemaju zajedničkih tačaka osim čvorova.

Definicija 1.39 Neka je G planaran graf. Tada sa *f* označimo *broj povezanih oblasti* $F_1, F_2, \ldots, F_f$ na koje G deli ravan.

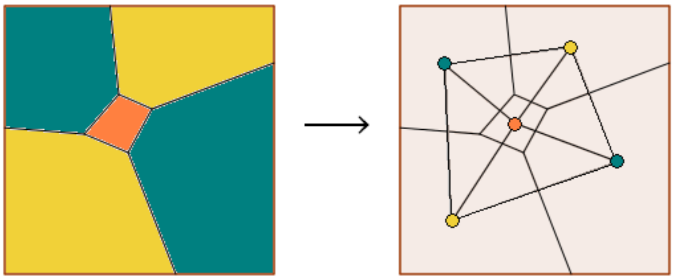
Definicija 1.40 Stepen oblasti F se označava sa d(F). To je broj grana koje ograničavaju oblast F. U svakom planarnom grafu sa n čvorova važi: $3 \leq d(F) \leq n$.

Problem četiri boje je pitanje *da li se može svaka geografska karta u ravni ili na sferi obojiti sa najviše četiri boje tako da zemlje sa zajendničkom granicom ne budu obojene istom bojom.* Pri čemu, pretpostavlja se da su zemlje kompaktne regije. Odnosno smatramo da su države susedne ako imaju zajedničku granicu, a ne samo zajedničku tačku.

Problem 4-obojivosti potiče od matematičara Fransis Gatri[1] ko je 1852. godine primetio da za bojenje karte nije potrebno više od četiri boje. U pismu pitao je profesora De Morgana[2] za uzrok te činjenice. On nije znao odgovor, pa je pismo poslao dalje kolegi Hemiltonu[3] u Dublin. Ali Hemilton nije bio zainteresovan. Engleski matematičar Kejli[4] je

[1] Francis Guthrie, (1831.– 1899.) Južno Afrički matematičar, botaničar
[2] Augustus De Morgan, (1806. – 1871.) engleski matematičar
[3] Sir William Rowan Hamilton, (1805. – 1865.) irski matematičar, fizičar i astronom
[4] Arthur Cayley, (1821. – 1895.) engleski matematičar

1878. godine predložio isti problem Londonskom matematičkom društvu i godinu dana kasnije Kempe[5] je tvrdio da je dokazao pretpostavku, ali posle su našli grešku u dokazu.



Slika 1.4

Pošto se svaka karta može nacrtati kao planaran graf, problem bojenja karte se može svesti na problem bojenja čvorova planarnog grafa tako što svaki čvor u grafu označi jednu regiju na karti, a dva čvora su povezana u grafu akko su te dve regije susedne kao na slici 1.4.
Nakon deset godina Kempeovog dokaza problema četiri boje Hivud[6] je pronašao grešku u dokazu. On je koristeći Kempeovu metodu dokazao problem 5-obojivosti.

Lema 1.41 U svakom povezanom planarnom multigrafu sa čvorovima stepena najmanje 3, postoji oblast stepena najviše 5.

<u>Dokaz</u> U [4], strana 27.

Lema 1.42 Neka je M mapa koja je 5-obojiva. Ako se pet oblasti susretaju u jednoj tački, tada se mapa može obojiti sa pet boja na takav način da se koriste samo četiri boje za tih pet oblasti.

<u>Dokaz</u>: U [4], strana 34.

Teorema 1.43 (Heawood, 1890) Svaka mapa je obojiva sa najviše pet boja.

<u>Dokaz</u> Indukcijom po broju oblasti mape dokazujemo tvđenje.

1. Baza indukcije će biti mapa sa najviše pet oblasti. Tada trivijalno važi da je pravilno bojenje moguće sa pet boja.

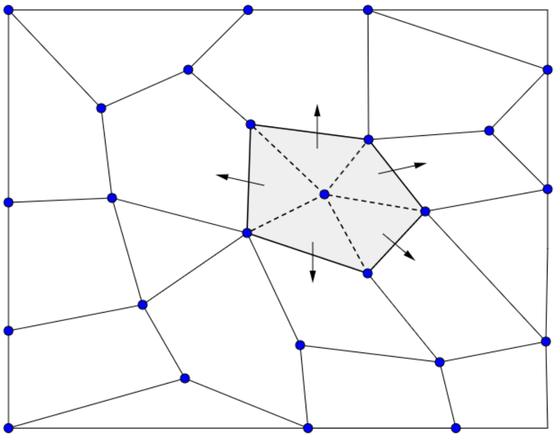
2. Indukcijska hipoteza: za svaku mapu sa najviše *n* oblasti važi da je 5-obojiva.

3. Indukcijski korak: dokažimo da je mapa sa *n+1* oblasti 5-obojiva.

Na osnovu leme 1.41 možemo uzeti oblast mape stepena pet ili manje. Izabranu oblast podelimo na toliko delova koliko ima suseda, tako da se linije podele susretaju u jednoj tački (slika 1.5). Bez umanjenja opštosti možemo pretpostaviti da je izabrana oblast stepena 5.

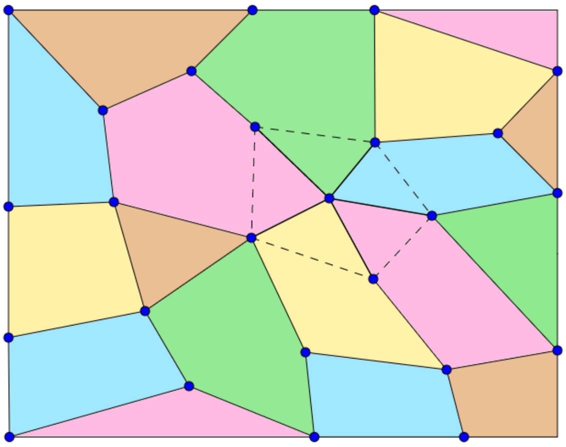
---

[5] Alfred Kempe, (1849. – 1922.) engleski matematičar
[6] Percy John Heawood, (1861. – 1955.) britanski matematičar

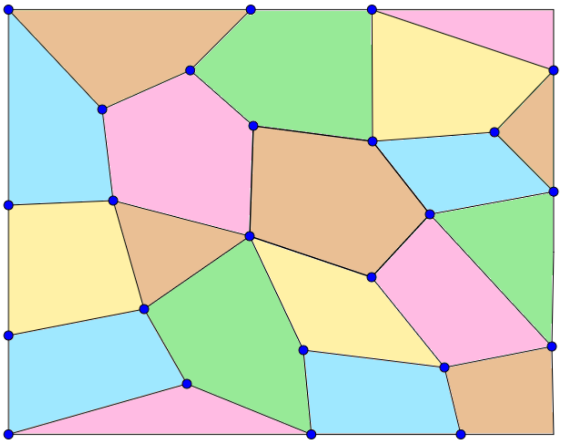


Slika 1.5

Dobijene delove oblasti podelimo među susedima na takav način da svaki deo ujedinimo sa jednom od susednih oblasti. Dobijena mapa će imati manji broj oblasti. Tada na osnovu indukcijske hipoteze dobijamo da je mapa 5-obojiva. Ako oblast ima četiri ili manje suseda, uvek ćemo moći da ih obojimo sa četiri boje. Slično, ako ima pet suseda, tada zbog leme 1.42 znamo da će biti dovoljno četiri boje za obojenje susednih oblasti (slika 1.6), dakle i tada ostaje jedna boja.



Slika 1.6



Slika 1.7

Vraćamo podeljenu oblast i obojimo sa preostalom bojom (slika 1.7). ∎

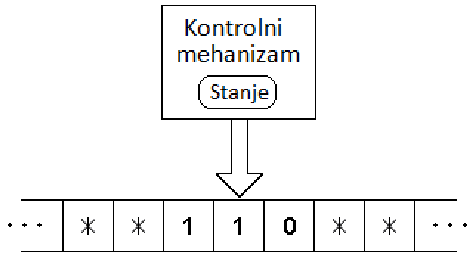
Problem četiri boje su 1976. godine pomoću računara dokazali, a za dokaz je trebalo oko 1.200 sata isprobavanjem 1.936 tipova mogućih karta. To su posle smanjili na 633 moguće karte. Tako je teorema četiri boje postao prvi od većih teorema koji su dokazali pomoću računara.

2 Uvod u teoriju algoritama

Teorija algoritama je oblast matematike koja izučava algoritamsku rešivost problema i složenost konkretnih algoritama za njihovo rešavanje.

U XX. veku konstruisanje Tjuringove mašine je donela revoluciju u računanju. Tjuringova mašina i danas daje osnovu za kompjuterske algoritme.

Tjuringova mašina (TM) se sastoji iz *kontrolnog mehanizma* koji može da bude u jednom od nekoliko stanja, *trake* koja je beskonačna s obe strane i *glave* koja u svakom trenutku pokaziva na jedno od polja trake Slika 2.1. U konačno mnogo polje trake se nalazi neki simbol iz unapred zadate azbuke, ostala polja sadrže simbol *blanko* (prazan simbol).



Slika 2.1

Glava u svakom trenutku čita simbol jednog polja trake. Za dato stanje kontrolnog mehanizma program mašine određuje novo stanje i glava umesto trenutnog simbola upiše novi simbol ili se pomeri na traci za jedno mesto udesno ili ulevo.

Definicija 2.1 *Tjuringova Mašina* je uređena sedmorka:

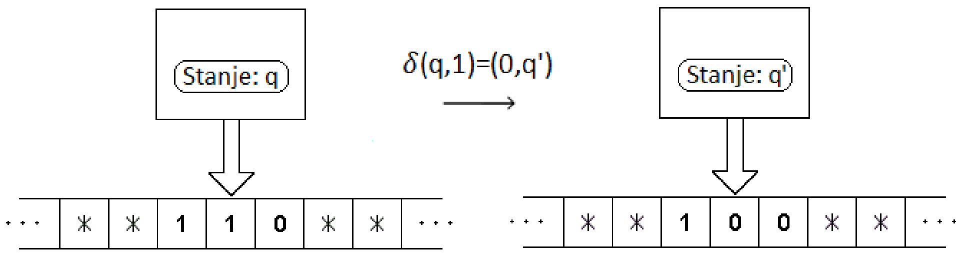
$$\mathcal{M}=(Q, \Sigma, \Gamma, \delta, q_0, \top, \bot),$$

gde je Q konačan skup stanja, Σ je ulazna azbuka, Γ je azbuka trake i važi $\Sigma \subset \Gamma$, $q_0 \in Q$ je početno stanje, $\top, \bot \in Q$ su redom stanja prihvatanja i odbijanja i δ je funkcija prelaza TM:

$$\delta: Q \times \Gamma \rightarrow (\, \Gamma \cup \{L, R, H\}\,) \times Q,$$

gde je L pomeranje glave levo, R pomeranje glave desno i H komanda zaustavljanja.

Funkcija prelaza određuje da li će se glava pomeriti na traci ili upisati novi simbol i naredno stanje mašine. Na primer δ(q,1)=(0,q') znači da ako je mašina u stanju q, a glava čita na traci simbol 1, simbol briše i umesto njega će upisati simbol 0 i pređe u stanje q' (slika 2.2).



Slika (2.2)

Definicija 2.2 *Konfiguracija* Tjuringove mašine $\mathcal{M}$ je uređena trojka:

$$(w_1, q, w_2) \in \Gamma^* \times Q \times \Gamma^*,$$

što znači da se kontrolni mehanizam nalazi u stanju q, na traku je upisana reč w_1w_2, a glava pokazuje na prvi simbol u reči w_2. Početna konfiguracija je (w, q_0, λ), gde je λ prazna reč.

Rad mašine možemo prikazati relacijom $\vdash$. Pišemo $(w_1, q, w_2) \vdash (w_1, q', w_2)$ ako iz jedne konfiguracije na osnovu funkcije prelaza možemo dobiti drugu konfiguraciju.
$(w_1, q, w_2) \vdash^* (w_1, q', w_2)$ zanči da TM može da pređe iz jedne konfiguracije u drugu u nekoliko koraka.

Definicija 2.3 *Jezik Tjuringove mašine* je skup reči koje TM prihvata:
$L(\mathcal{M}) = \{w \in \Sigma^*, (w, q_0, \lambda) \vdash^* (w_1, T, w_2)$, za neke reči $w_1, w_2 \in \Gamma^*\}$.

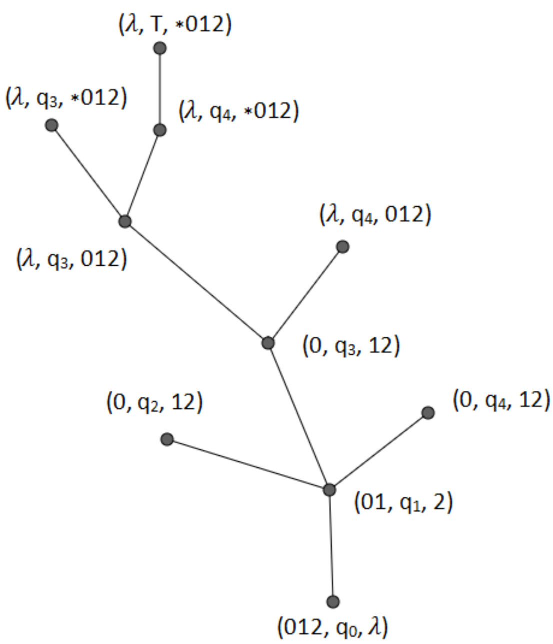
Pored osnovnog modela postoje višetračne Tjuringove mašine.

Nedeterminističke Tjuringove mašine:

Nedeterministička Tjuringova mašina (NTM) je Tjuringova mašina kod kojeg naredno stanje nije jednoznačno određen, nego postoji više mogućnosi među kojima se bira naredno stanje. Funkcija prelaza NTM je definisana na sledeći način:

$$\delta: Q \times \Gamma \to \mathcal{P}((\Gamma \cup \{L, R, H\}) \times Q),$$

odnosno za δ(q,a) je skup mogućih opcija za jedan korak TM. Skup mogućih koraka rada mašine predstavlja se drvetom izračunavanja. Koren drveta je početna konfiguracija (w, q_0, λ), a čvorovi su moguće konfiguracije. Primer drveta izračunavanja za funkciju prelaza δ i početnu konfiguraciju (012, q_0, λ) na slici 2.3, gde δ ima sledeće vrednosti:



Slika 2.3

$q_0 * L\, q_1$	$q_1\, 2\, L\, q_2$	$q_1\, 2\, L\, q_3$	$q_1\, 2\, L\, q_4$
$q_3\, 1\, L\, q_3$	$q_3\, 1\, L\, q_4$	$q_3\, 0\, L\, q_3$	$q_3\, 0\, L\, q_4$
$q_3 * L\, q_2$	$q_4 * H\, T$.		

Zapis: $q_i\ a\ b\ q_j$ znači $\delta(q_0, *) \supset (L, q_1)$.

Definicija 2.4 *Jezik nedeterminističke Tjuringove mašine* je skup reči koje NTM prihvata, odnosno skup svih reči $w \in \Sigma^*$ takvih da za početnu konfiguraciju (w, q_0, λ) postoji čvor u drvetu izračunavanja oblika (u_1, T, u_2).

Teorema 2.5 Za svaku NTM $\mathcal{M}$ postoji 3-tačna TM $\mathcal{D}$ takva da je $L(\mathcal{D})=L(\mathcal{M})$.

Dokaz: Pretpostavimo da za mašinu $\mathcal{M}$ važi $|\delta\ (q,a)| \leq b$, za neko $b \in \mathbb{N}$ za sve $q \in Q$, $a \in \Gamma$. Neka su trake mašine $\mathcal{D}$ redom A,B,C . Na početku na traci A je upisana ulazna reč w, a B i C su prazne. Traka B će služiti da izvršimo izračunavanja, a na traku C upisaćemo „adrese" - nizove skupa {1, …, b} u sledećem poretku: niz x<y ako je x kraći od y, ili su x i y iste dužine i za najmanje *i* takvo da je $x_i \neq y_i$ važi $x_i < y_i$.

Sledeći algoritam opisuje rad mašine $\mathcal{D}$:

1. Kopiramo reč w sa trake A na traku B i postavimo glavu trake B iza kopirane reči.

2. Traka B simulira rad mašine $\mathcal{M}$. Ako je u k-tom koraku mašina u stanju q i čita slovo a sa funkcijom prelaza: $\delta(q,a)=\{(\alpha_1,q_1), \dots , (\alpha_c,q_c)\}$,

gde $c \leq b$, i na k-tom polju trake C je broj $m_k \leq c$, tada primenimo komandu: $q\ a\ \alpha_{m_k} q_{m_k}$.

3. Ako na traci B stignemo do stanja prihvatanja (T), prihvatimo ulaznu reč i stanemo.

4. Inače, upišemo na traku C, brišemo sadržaj trake B i vratimo se na korak 1.

Ovaj algoritam je deterministički i prahvata ulaznu reč w ako i samo ako prihvata i mašina $\mathcal{M}$. ∎

Vremenska i prostorna složenost:

Vremenska i prostorna složenost algoritma je funkcija koja povezuje veličinu ulaza problema sa brojem koraka izvršavanja algoritma ili brojem potrebnih polja trake.

Korak algoritma je izvršenje jedne komande TM koja rešava odgovarajući problem. Broj koraka koji je potreban da bi mašina završila svoj rad zavisi od ulaznog podatka.

Možemo meriti i broj polja trake koja data TM koristi tokom određenog izračunavanja (polja koja zauzima ulazni podatak se ne računaju, već samo polja u koje mašina piše tokom rada). Ova mera je prostorna složenost algoritma.

Vremenska i prostorna složenost algoritama se procenjuje u asimptotskom smislu, odnosno funkcija složenosti se procenjuje za dosta velike dužine ulaza. Za to se koriste "veliko O" notacija ($\mathcal{O}()$). Za funkcije f, g: $\mathbb{R} \to \mathbb{R}$ pišemo da je f= $\mathcal{O}$(g), ako postoji konstanta C > 0 i
$x_0 \in \mathbb{R}$ tako da za sve $x \geq x_0$ važi jednakost

$$f(x) \leq C\, g(x).$$

Definicija 2.6 Vremenske klase složenosti:

TIME(f(n))= $\{\mathrm{L} : \text{ postoji } \mathcal{M} \text{ tako da } \mathrm{L}(\mathcal{M}) = \mathrm{L}, \mathrm{T}_{\mathcal{M}}(\mathrm{n}) = \mathcal{O}(\mathrm{f}(\mathrm{n}))\}$, gde $\mathrm{T}_{\mathcal{M}}(\mathrm{n})$ znači najviše potrebno vreme za rad Tjuringove mašine $\mathcal{M}$ za ulazni podatak dužine najviše n.

Klasa jezika odlučivih u vremenu koje se može oceniti polinomnom fonkcijom:
P = $\bigcup_{k \geq 0} \mathrm{TIME}\ (n^k)$.

Definicija 2.7 Prostorne klase složenosti:

SPACE(f(n))= $\{\mathrm{L} : \text{ postoji } \mathcal{M} \text{ tako da } \mathrm{L}(\mathcal{M}) = \mathrm{L}, \mathrm{S}_{\mathcal{M}}(\mathrm{n}) = \mathcal{O}(\mathrm{f}(\mathrm{n}))\}$, gde $\mathrm{S}_{\mathcal{M}}(\mathrm{n})$ znači najviše potreban prostor na traci za rad Tjuringove mašine $\mathcal{M}$ za reč dužine najviše n.

Prostorne klase algoritama:
L= SPACE($\log^n$),

PSPACE=$\bigcup_{k \geq 0} \mathrm{SPACE}\ (n^k)$.

Definicija 2.8 Nedetermininstičke klase složenosti:

NTIME(f(n))= $\{\mathrm{L} : \text{ postoji NTM } \mathcal{M} \text{ tako da } \mathrm{L}(\mathcal{M}) = \mathrm{L}, \mathrm{T}_{\mathcal{M}}(\mathrm{n}) = \mathcal{O}(\mathrm{f}(\mathrm{n}))\}$,

NSPACE(f(n))= $\{\mathrm{L} : \text{ postoji NTM } \mathcal{M} \text{ tako da } \mathrm{L}(\mathcal{M}) = \mathrm{L}, \mathrm{S}_{\mathcal{M}}(\mathrm{n}) = \mathcal{O}(\mathrm{f}(\mathrm{n}))\}$,

NP = $\bigcup_{k \geq 0} \mathrm{NTIME}\ (n^k)$.

NL= NSPACE($\log^n$),

NPSPACE=$\bigcup_{k \geq 0} \mathrm{NSPACE}\ (n^k)$.

Problem P=NP je jedan od najznačajnijih otvorenih matematičkih problema. Inkluzija
P $\subseteq$ NP trivijalno važi, pošto za Problem P=NP se svodi na pitanje: da li se svaki nedeterministički algoritam može determinizovati tako da ostane polinomne složenosti? Fondacija Clay je 2000. godine raspisala nagradu od US$ 1 000 000 prvoj osobi koja dokaže rešenje, a problem je i još danas otvoreno.

Klase NP-kompletnih problema

Definicija 2.9 Neka je $\mathcal{A}$=(Γ, A) problem odlučivanja. Za problem $\mathcal{B}$ kažemo da *proverava* (verifikuje) problem $\mathcal{A}$ ako je $\mathcal{B}$ oblika (Γ × Δ, B) za neki skup Δ i ako za sve x ∈ Γ važi:

$$x \in A \Leftrightarrow \exists\, (c \in \Delta)\ (x,c) \in B.$$

Pri čemu ako (x,c) ∈ B, tada se c zove *sertifikat* (dokaz) za x.

Teorema 2.10 Neka je $\mathcal{A}$=(Γ, A) problem odlučivanja. $\mathcal{A}$ ∈ NP ako i samo ako $\mathcal{A}$ ima verifikator $\mathcal{B}$ tako da je $\mathcal{B}$ ∈ P.

<u>Dokaz</u>: U [3], strana 121.

Definicija 2.11 Neka su $\mathcal{A}$=(Γ_1, A) i $\mathcal{B}$=(Γ_2,B) dva problema odlučivanja. Kažemo da se $\mathcal{A}$ (polinomno) redukuje na $\mathcal{B}$, u oznaci $\mathcal{A} \leq_p \mathcal{B}$, ako postoji redukciona funkcija σ: $\Gamma_1 \to \Gamma_2$ koja ima sledeće osobine:

1) σ je izračunljiva u determinističkom polinomnom vremenu,
2) za sve x ∈ Γ_1 važi: x ∈ A ⟺ σ(x) ∈ B.

Lema 2.12 Ako $\mathcal{A} \leq \mathcal{B}$ i $\mathcal{B}$ ∈ P, tada $\mathcal{A}$ ∈ P.

<u>Dokaz</u>: Pretpostavimo da $\mathcal{A} \leq \mathcal{B}$. Tada znamo da postoji TM $\mathcal{M}_\sigma$ koja u polinomnom vremenu izračuna funkciju σ, koja redukuje problem $\mathcal{A}$ na problem $\mathcal{B}$. Dalje, iz $\mathcal{B}$ ∈ P sledi da postoji TM $\mathcal{M}$ koja rešava $\mathcal{B}$ u polinomom vremenu. Kompozicija ove dve funkcije rešava problem $\mathcal{A}$ u polinomnom vremenu, dakle dobijamo da $\mathcal{A}$ ∈ P. ∎

Definicija 2.13 Za problem $\mathcal{A}$ kažemo da je NP-težak ako za sve probleme $\mathcal{B}$ ∈ NP važi da je $\mathcal{B} \leq \mathcal{A}$. Problem $\mathcal{A}$ je NP-kompletan ako $\mathcal{A}$ ∈ NP i $\mathcal{A}$ je NP-težak.

Teorema 2.14 Neka je $\mathcal{A}$ NP-kompletan problem. Tada važi: $\mathcal{A}$ ∈ P ⟺ P=NP.

<u>Dokaz</u>: (⟸) Trivijalno: $\mathcal{A}$ ϵ NP i NP = P ⟹ $\mathcal{A}$ ϵ P.

(⟹) Neka je problem $\mathcal{B}$ ϵ NP. Tada za problem $\mathcal{A}$ ϵ NP važi $\mathcal{B} \leq \mathcal{A}$. Iz pretpostavke $\mathcal{A}$ ϵ P na osnovu prethodne leme dobijamo $\mathcal{B}$ ∈ P, dakle imamo NP ⊆ P. Pošto smo već rekli da druga inkluzija (P ⊆ NP) trivijalno važi, sledi tražena jednakost. ∎

Definicija 2.15 Neka je S skup iskaznih slova. *Literal* je svako iskazno slovo ili negacija iskaznog slova tj. svaki p odnosno ¬ p, gde p ϵ S. Literal *suprotan* literalu p jeste literal ¬ p i obrnuto.

Definicija 2.16 *Klauza* c_i je disjunkcija konačnog broja literala c_i : $L_1^i \vee L_2^i \vee \ldots \vee L_k^i$, k ≥ 0. Klauza koja sadrži samo jedan literal naziva se *jedinična klauza*.

Definicija 2.17 Kažemo da je iskazna formula F u *konjunktivnoj formi* ako je F konačna konjunkcija iskaznih formula: F = $\bigwedge c_i$, *i*= 1, ..., n, gde su c_i klauze za svako *i.*

Definicija 2.18 *Problem SAT* (problem zadovoljivosti iskaznih formula) je problem ispitivanja postojanja valuacije u kojoj je data formula u konjunktivnoj formi tačna.

Definicija 2.19 *Problem k-SAT* je problem SAT za formule u kojima svaka klauza ima najviše po k literala (k-konjunktivna forma).

Primer
Formula F je u 3-konjunktivnoj formi:

$F = (p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$

Istinosne tablice se upotrebljavaju već od 1920. godine i to je najjednostavnija potpuna metoda rešavanja problema SAT. Ali za ovaj problem do danas nije nađeno efikasan polinomni algoritam. Sledeća teorema kaže da je problem SAT NP-kompletan problem. To žnači da nalaženje odgovarajućeg polinomnog algoritma za SAT rešilo bi problem P=NP, a dokaz da takav algoritam ne postoji značilo bi: P$\neq$NP.

Teorema 2.20 (Kuk[7], Levin[8]) SAT je NP-kompletan problem.

Dokaz: U [3], strana 125.

Ovo tvrđenje je značajno zbog toga što svođenjem brojnih problema na SAT dobijamo niz NP-kompletnih problema.

Teorema 2.21 Neka je $\mathcal{A}$ NP-kompletan problem. Ako $\mathcal{A} \leq \mathcal{B}$ i $\mathcal{B} \in$ NP, tada je i problem $\mathcal{B}$ takođe NP-kompletan.

Dokaz: U [3], strana 128.

Teorema 2.22 1) 2-SAT $\in$ P
2) SAT $\leq$ 3-SAT
3) k-SAT je NP-kompletan problem za sve k $\geq$ 3 .

Dokaz U [3], strane 132-135.

[7] Stephen Arthur Cook, (1939 -) američki informatičar
[8] Leonid Levin (Леони́д Анато́льевич Ле́вин), (1948 -) sovjetsko-američki naučnik, informatičar

3 NP-kompletnost problema k-obojivosti

Problem bojenje grafa: Neka je dat graf G. Da li možemo čvorove grafa G obojiti sa k ili manje boja tako da su susedni čvorovi obojeni različitim bojem?

Neka je C= {c1, ... , ck} – skup boja
bojenje čvorova: f: V(G) $\rightarrow$ C

pravilno bojenje čvorova: f: V(G) $\rightarrow$ C, uv $\in$ E(G) $\Rightarrow$ f(u) $\neq$ f(v)

Problem k-obojivosti (označimo sa k-COL):

ULAZ: Graf G
IZLAZ: Da li je G k-obojiv?

Problem bojenje grafa je NP-kompletan problem:

Za k=1: 1-COL trivijalno: G je 1-obojiv akko E=$\emptyset$
Za k=2: Treba proveriti da li je graf bipartitan. Pomoću BFS ili DFS algoritma možemo dobiti polinomni algoritam. To ćemo detaljnije videti u glavi 4.
Za k≥3: Problem je NP-kompletan. Da bismo pokazali ovo tvrđenje, dovoljno je pokazati da se problem k-obojivosti može svesti na problem 3-SAT.

Svođenje problema k-obojivosti na problem 3-SAT

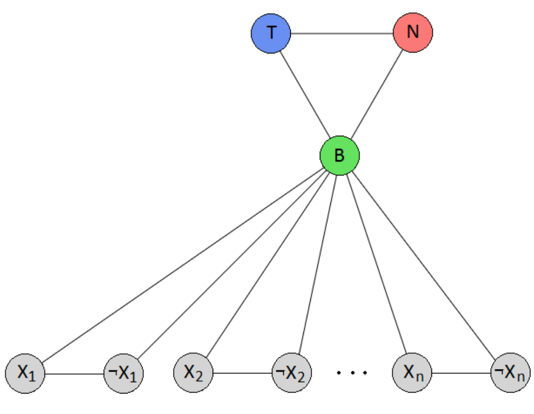
Teorema 3.1 3-COL je NP-kompletan.

<u>Dokaz</u> Da bi to važilo, treba da bude zadovoljen da 3-COL $\in$ NP i da je NP težak.

3-COL $\in$ NP: Važi na osnovu teoreme 2.9, pošto za vremenske složenosti $\mathcal{O}$(m) možemo proveriti da li je 3-bojenje pravilno. Sertifikat problema 3-COL će biti dato 3-bojenje.

3-COL je NP težak: pokažimo 3-SAT ≤ 3-COL:

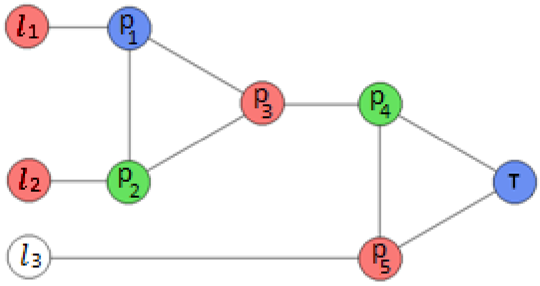
Neka je ϕ instanca za 3-SAT. Tada važi da svaka klauza u ϕ ima najviše po 3 literala. Neka su C_1, C_2, ..., C_m klauze u ϕ nad promenljivama {x_1, ..., x_n}. Konstruišemo graf G tako da bude 3-obojiv akko je ϕ zadovoljiv: G će sadržati čvorove koji će biti označeni kao promenljive u ϕ: x_1, ..., x_n i negaciju svake promenjive: $\neg x_1$, ..., $\neg x_n$. Parove čvorova x_i i $\neg x_i$ međusobno povezujemo da bi bili različite boje. Zatim dodamo još u G čvo-



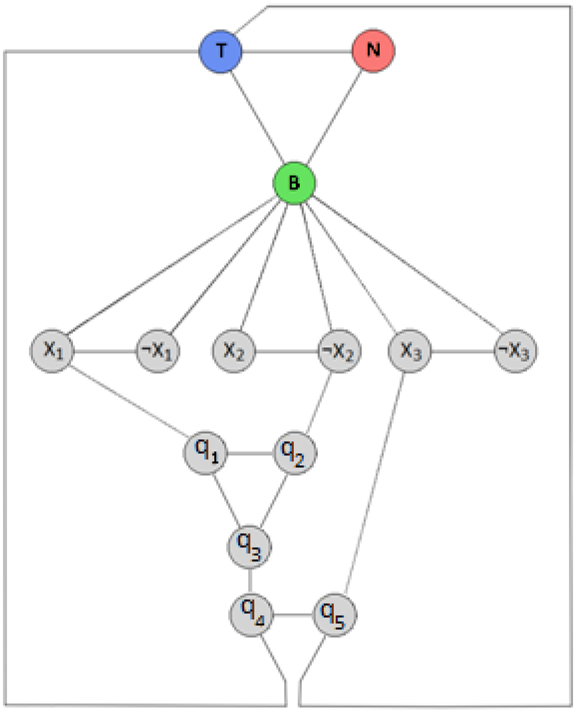
Slika 3.1

rove za istinitosne vrednosti tačno (T), netačno (N) i čvor B (baza) koji obezbeđuje da čvorovi x_i i $\neg x_i$ budu obojeni kao T i N tako što ćemo povezati B sa svakim od x_i i $\neg x_i$, za svako i=1, …, n; odnosno povezani će još biti čvorovi T, N i B međusobno kao na slici 3.1. Dakle i T, N, B će biti različite boje. Neka su ta boja plava, crvena i zelena redom.

Čvorovi x_i, $\neg x_i$, i=1, …, n će dobiti boje na osnovu vrednosti promenljive x_i u datoj valuaciji, tj. ako imaju vrednost T, obojimo ih istom bojom (plavom) kao i čvor T, a čvorove vrednosti ⊥ drugom bojom (crvenom) kao i čvor N.



Slika 3.2



Slika 3.3

Dodamo još za svaku klauzu $(l_1 \vee l_2 \vee l_3)$, gde su l_i literali, po pet pomoćnih čvorova: p_1, p_2, p_3, p_4, p_5 (slika 3.2).

Vidimo da ako želimo da istinosna vrednost klauze bude tačna, tj. da poslednji čvor dobije plavu boju, u klauzi bar jedan od čvorova $\{l_1, l_2, l_3\}$ mora da bude plave boje, odnosno jedan od literala $\{l_1, l_2, l_3\}$ mora da ima vrednost T.

Na slici 3.3 vidimo primer kako izgleda graf G za klauzu: $C_1=x_1 \vee \neg x_2 \vee x_3$.

Formula F je zadovoljiva ako i samo ako je graf G 3-obojiv:

(⟹) Ako imamo formulu F koja za neku interpretaciju tačna, tada u svakoj klauzi će se pojaviti promenljiva vrednosti T. To znači da za svaku klauzu formule F među čvorovima $\{x_j, \neg x_k$, j=1, …, r; k=1, …, t$\}$ koji čine klauzu, biće različite boje (u našem primeru crvene i plave). Tada imamo dva slučaja:

1° l_1, l_2 su crveni: l_3 mora da bude plav. Čvorove p_1 i p_2 možemo obojiti plavom i zelenom bojom, pa će p_3 biti crven kao i p_5 (slika 3.2). Ostaje p_4 kojim ćemo pridružiti zelenu boju.

2° l_1 ili l_2 plav: tada p_1 ili p_2 može biti crven. Dobijamo da i p_4 može biti crven, a p_3 i p_5 zeleni.

(⟸) Treba pokazati da ako je moguće pravilno 3-bojenje grafa G, sledi zadovoljivost formule F. Vrednost formule F će biti tačna, ako za neku klauzu formule među literalima postoji bar jedna koja ima vrednost T, dakle uslov je da za svaku klauzu imamo bar jedan čvor u grafu G među l_1, l_2, l_3 koji je plav. A to smo već spominjali da uvek mora da važi. U suprotnom, ako bi bili l_1, l_2, l_3 crveni, jedan od p_1 i p_2 bi morao biti plav a drugi zelen. Tada p_3 bi bio crven, a pošto je i l_3 crven, opet jedan od p_4 i p_5 bi morao biti plav a drugi zelen, pa ne bismo mogli povezati p_4 i p_5 sa čvorom T koji je plav (slika 3.2). Dakle, ne bi bilo izvodljivo pravilno 3-bojenje, a to je u kontradikciji sa pretpostavkom trvđenja.

Time smo redukovali 3-COL na 3-SAT i dobili da je 3-COL NP kompletan. ∎

Teorema 3.2 k-COL je NP-kompletan za svako k$\in \mathbb{N}$.

<u>Dokaz</u>

<u>3-COL ≤ 4-COL</u>:
Neka je G dati graf. Definišemo graf G' tako da važi da je G 3-obojiv akko je G' 4-obojiv, gde je G'=(V', E'), V'=V U {v'}, E'= E U {v'v_i, $v_i \in$V}.

Ako je G 3-obojiv, tada je G' 4-obojiv.
Ako G nije 3-obojiv, tada G' ne može biti 4-obojiv zbog definicije grafa G'.

Na osnovu definicije 2.10 problem 4-COL se polinomno redukuje na 3-COL, gde je σ(G)=G'. Tada na osnovu teoreme 2.21 imamo da je 4-COL je NP-kompletan.

Slično, 4-COL ≤ 5-COL, pa je 5-COL NP-kompletan.
Nastavljajući postupak dobijamo da je (k-1)-COL ≤ k-COL što implicira da je k-COL NP-kompletan za svako k$\in \mathbb{N}$. ∎

4 Algoritmi za nalaženje hromatskog broja

U ovom poglavlju ćemo opisati algoritme koji će nakon konačnog broja koraka odrediti hromatski broj grafa i dati postupak pravilnog bojenja čvorova grafa.

4.1 Algoritam za određivanje 2-obojivosti

Da bi smo odredili da li je graf 2-obojiv ili ne, dovoljno je videti da li je graf bipartitan. Ako je bipartitan, čvorove možemo poređati u dve klase tako da nikoja dva čvora u istoj klasi nisu povezani. Tada čvorove iz jedne klase obojimo sa jednom bojom, a čvorove iz druge klase sa drugom bojom.

Konstruišimo algoritam koji će za svaki graf odrediti da li je graf bipartitan. U algoritmu podrazumevamo da je graf povezan. U slučaju kada imamo nepovezan skup, algoritam lako možemo dopuniti da radi i za nepovezane na osnovu teoreme 1.19. Nepovezan graf je k-obojiv ako i samo ako mu je svaka komponenta k-obojiva. Zbog toga, algoritam treba da primenimo na svaku komponentu posebno i na osnovu toga ćemo videti da li je ceo graf 2-obojiv ili ne.

Problem 2-obojivosti: Za svaki čvor treba odrediti da li je na parnom ili na neparnom rastojanju od datog čvora. Pored toga, treba proveriti da li graf sadrži neparnu konturu.

Opis algoritma: Fiksiramo čvor $s \in V(G)$. Neka je L lista čvorova u koju upišemo čvorove koje ispitujemo i X lista čvorova koje smo već obradili. Koristimo još pomoćne skupove P i N, u kojoj će biti čvorovi koji su na parnom (P) ili neparnom (N) rastojanju od čvora s grafa G. Na početku u listama L i P je samo čvor s, a X i N je prazan. U svakom koraku obrađujemo jedan čvor iz L tako što dodajemo njegove susede u liste L, X i u N ili P, a njega izbacujemo iz L (čvorove koji su susedni sa nekim čvorom iz P ubacujemo u N, a susede iz P ubacujemo u N). U svakom koraku proveravamo da li su skupovi P i N disjunktni. Algoritam se završava kada smo već obradili sve čvorove grafa.

Algoritam:

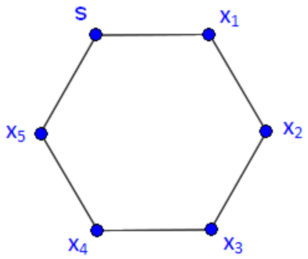
Korak 1. $L:=\langle s\rangle$, $X:=\emptyset$, $P:=\{s\}$, $N:=\emptyset$

Korak 2. Za prvi čvor $u \in L$ ubacujemo njegove susede u L. Zatim još susede čvora *u* ubacujemo u skupu P ili N (susede čvorova iz skupa P ubacujemo u N, a susede iz skupa N u P). Obrađeni čvor *u* izbacujemo iz L i ubacujemo u X.

Korak 3. Proverimo da li su skupovi P i N disjunktni. Ako nisu, kraj algoritma, i graf nije bipartitan. Ako su disjunktni, pređemo na korak 2, sve dok $X \neq V$. Ako je $X=V$, kraj algoritma, i ako su P i N disjunktni skupovi onda je graf bipartitan.

Primer 1

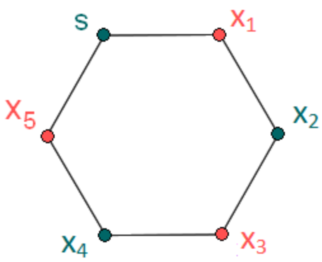
Neka je graf G dat na slici 4.1, V=$\{s, x_1, x_2, x_3, x_4, x_5\}$.



Slika 4.1

Korak 1. L:=$\langle s\rangle$, X:=$\emptyset$, P:=$\{s\}$, N:=$\emptyset$
Korak 2. L:=L U $\{x_1, x_5\}$ \ $\{s\}$=$\langle x_1, x_5\rangle$, X:=$\{s\}$, P:=$\{s\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_5, x_2\rangle$, X:=$\{s, x_1\}$, P:=$\{s, x_2\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_2, x_4\rangle$, X:=$\{s, x_1, x_5\}$, P:=$\{s, x_2, x_4\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_4, x_3\rangle$, X:=$\{s, x_1, x_5, x_2\}$, P:=$\{s, x_2, x_4\}$, N:=$\{x_1, x_5, x_3\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_3\rangle$, X:=$\{s, x_1, x_5, x_2, x_4\}$, P:=$\{s, x_2, x_4\}$, N:=$\{x_1, x_5, x_3\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\emptyset$, X:=V, P:=$\{s, x_2, x_4\}$, N:=$\{x_1, x_5, x_3\}$
Korak 3. X=V, kraj algoritma. Skupovi P i N su disjunktni.

Dakle, graf G je bipartitan, stoga i 2-obojiv. Za pravilno bojenje uzećemo jednu boju za čvorove iz P, a drugu boju za čvorove iz N (Slika 4.3).

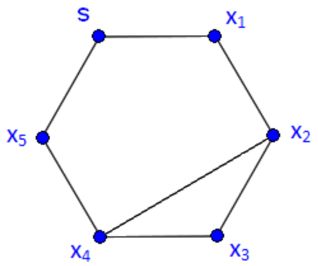


Slika 4.2

Korake algoritma možemo prikazati i u tablici:

L	X	P	N
$\langle s\rangle$	$\emptyset$	$\{s\}$	$\emptyset$
$\langle x_1, x_5\rangle$	$\{s\}$	$\{s\}$	$\{x_1, x_5\}$
$\langle x_5, x_2\rangle$	$\{s, x_1\}$	$\{s, x_2\}$	$\{x_1, x_5\}$
$\langle x_2, x_4\rangle$	$\{s, x_1, x_5\}$	$\{s, x_2, x_4\}$	$\{x_1, x_5\}$
$\langle x_4, x_3\rangle$	$\{s, x_1, x_5, x_2\}$	$\{s, x_2, x_4\}$	$\{x_1, x_5, x_3\}$
$\langle x_3\rangle$	$\{s, x_1, x_5, x_2, x_4\}$	$\{s, x_2, x_4\}$	$\{x_1, x_5, x_3\}$
$\emptyset$	V	$\{s, x_2, x_4\}$	$\{x_1, x_5, x_3\}$

Primer 2



Slika 4.3

Neka je graf G dat na slici 4.3, V=$\{s, x_1, x_2, x_3, x_4, x_5\}$.

Slično kao u prethodnom primeru:
Korak 1. L:=$\langle s\rangle$, X:=$\emptyset$, P:=$\{s\}$, N:=$\emptyset$
Korak 2. L:=$\langle x_1, x_5\rangle$, X:=$\{s\}$, P:=$\{s\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_5, x_2\rangle$, X:=$\{s, x_1\}$, P:=$\{s, x_2\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_2, x_4\rangle$, X:=$\{s, x_1, x_5\}$, P:=$\{s, x_2, x_4\}$, N:=$\{x_1, x_5\}$
Korak 3. Skupovi P i N su disjunktni.
Korak 2. L:=$\langle x_4, x_3\rangle$, X:=$\{s, x_1, x_2, x_5\}$, P:=$\{s, x_2, x_3\}$, N:=$\{x_1, x_3, x_4, x_5\}$
Korak 3. Pošto skupovi P i N nisu disjunktni (x_4 pripada i skupu P i skupu N), kraj algoritma, graf G nije bipartitan.

L	X	P	N
$\langle s\rangle$	$\emptyset$	$\{s\}$	$\emptyset$
$\langle x_1, x_5\rangle$	$\{s\}$	$\{s\}$	$\{x_1, x_5\}$
$\langle x_5, x_2\rangle$	$\{s, x_1\}$	$\{s, x_2\}$	$\{x_1, x_5\}$
$\langle x_2, x_4\rangle$	$\{s, x_1, x_5\}$	$\{s, x_2, x_4\}$	$\{x_1, x_5\}$
$\langle x_4, x_3\rangle$	$\{s, x_1, x_2, x_5\}$	$\{s, x_2, x_4\}$	$\{x_1, x_3, x_4, x_5\}$

Složenost algoritma

Za izračunavanje složenosti potrebno je znati kako se graf čuva u memoriji računara. Graf možemo predstaviti i čuvati u memoriji preko matrica susedstva i liste susedstva.

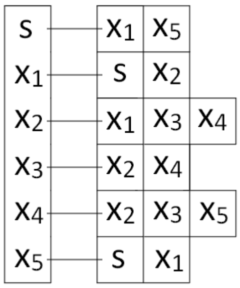
Matricu susedstva za graf G=(V,E), |V|=n, definišemo na sledeći način:

$$A(G)=[a_{ij}],\ \text{gde}\ a_{ij} = \begin{cases} 1, & (i,j) \in E \\ 0, & \text{inače.} \end{cases}$$

Matrica susedstva grafa iz primera 2. (slika 4.3) je tada:

$$A(G)= \begin{array}{c} \\ s \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccccc} s & x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Lista susedstva je niz velične n (gde je n broj čvorova) i-ti element sadrži povezanu listu u kojoj se nalaze svi čvorovi između kojih i čvora v_i postoji grana.
Za graf iz primera 2. lista susedstva izgleda ovako (slika 4.3):



Slika 4.4

Složenost algoritma računamo za broj čvorova *n* i broj grana *m*. Prikazivanje grafa u matričnom obliku nije najpogodnije za grafove sa manjim brojem grana (odnosno kada je broj *m* znatno manji od n^2), pošto tada čuvamo u memoriji veliki broj nula i samo nekoliko jedinica. Ako se graf čuva u memoriji preko liste susedstva, tada čuvamo samo skup suseda svakog čvora i za to u praksi je ponekdad dovoljno znatno manje prostora.

Složenost vremena prethodnog algoritma računamo na sledeći način:

Korak 1. je složenosti vremena $\mathcal{O}(1)$. U koraku 2. pretraživanje suseda čvorova kod matričnog zapisa je složenosti $\mathcal{O}(n)$ za svaki čvor, ukupno $\mathcal{O}(n^2)$, a kod liste susedstva $\mathcal{O}(m)$: svaku granu uv dvaput računamo, jednom za čvor u, i jednom za v. Dalje, svaki čvor najviše jednom ubacimo u L, izbacimo iz njega i ubacimo ili u P ili u N - to zahteva $\mathcal{O}(n)$ vremena. Za korak 3. za proveru da li su P i N disjunktni za sve čvorove je složenosti $\mathcal{O}(n)$. Ukupno za algoritam potrebno: u matričnom obliku $\mathcal{O}(1) + \mathcal{O}(n^2) + \mathcal{O}(n) = \mathcal{O}(n^2)$ vremena, a preko liste susedstva $\mathcal{O}(1) + \mathcal{O}(m) + \mathcal{O}(n) = \mathcal{O}(m) + \mathcal{O}(n)$ vremena.

4.2 Rekurentna relacija

Sledeća teorema sa pomoćnom definicijom daje dovoljan uslov za r-obojenje čvorova grafa. To ćemo koristiti u konkretnim algoritmima.

Definicija 4.2 *r-podgraf* grafa G = (V, E) je r-hromatski podgraf S, gde je $V(S) \subseteq V(G)$. Ako ne postoji podgraf $H \subseteq V(G)$ tako da je $V(S) \subset V(H)$ i da je H r-hromatski, tada se S naziva *maksimalan r-podgraf* grafa G.
Familiju svih maksimalnih r-podgrafova označimo sa $Q_r[G]$, a j-ti element te familije sa $S_r^j[G]$.

Teorema 4.3 Ako je graf G r-hromatski, tada se G može pravilno obojiti sa r (ili manje) boja bojeći prvo sa jednom bojom maksimalan nezavisan skup $S_1[G]$, pa sa drugom bojom skup $S_1[\langle V - S_1 \rangle]$ i tako nastavljajući dok nije svaki čvor obojen.

Dokaz: U [1], strana 63.

Ovu teoremu možemo koristiti za uspostavljanje rekurentne relacije tako da ćemo maksimalan r-podgraf dobiti od maksimalnih r-1 – podgrafova.

Neka je $Q_{r-1}[G]$ familija svih maksimalnih r-1 – podgrafova grafa G i neka je $S^{j}_{r-1}[G]$ j-ti element te familije. Skup $S^{k}_{1}[G]$ je tada skup čvorova k-tog maksimalnog 1-podgrafa (nezavisan skup) grafa G. Odnosno, skup $S^{k}_{1}[G^{j}]$ je skup čvorova k-tog maksimalnog 1-podgrafa grafa $G^{j}_{r} = \langle V - S^{j}_{r} \rangle$.

Familiju $Q_r[G]$ svih maksimalnih r-podgrafova od G dobijamo na sledeći način:
Definišemo skup $H^{j,k}$ sa $H^{j,k} = S^{j}_{r-1}[G] \cup S^{k}_{1}[G^{j}_{r}]$, za svako j = 1, ..., q_{r-1} i k=1,..., q^{j}_{1}, gde su q_{r-1} i q^{j}_{1} su brojevi r-1 – podgrafova od G i 1-podgrafova od G^{j} redom.

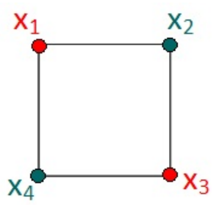
Familija maksimalnih r-podgrafova $Q_r[G]$ je tada sadržan u familiji Θ (familiji skupova H^{i}) i $Q_r[G]$ možemo dobiti od njih, izostavljajući one skupove koji su već sadržani u drugim skupovima:
$Q_r[G] = \{H^{j,k} \mid H^{j,k} \in \Theta, \text{ i } H^{j,k} \not\subset H^{l,m} \text{ za svako } H^{l,m} \in \Theta, (j,k) \neq (l,m)\}$.

Kod bojenja za razdvajanje skupova $S^{j}_{r-1}[G]$ i $S^{k}_{1}[G^{j}_{r}]$ stavićemo marker između ta dva skupova da bismo mogli čvorove istog nezavisnog skupa obojiti jednom bojom (a različitom od ostalih), odnosno pisaćemo:
$H^{j,k} = \{S^{j}_{r-1}[G] | S^{k}_{1}[G^{j}_{r}]\} = \{v_{i_1}, v_{i_2}, \ldots | v_{j_1}, v_{j_2}, \ldots\}$, za neke $v_{i_1}, v_{i_2}, \ldots, v_{j_1}, v_{j_2}, \ldots \in V$.
Čvorovi ispred markera će biti obojeni jednom bojom, a čvorovi posle markera drugom bojom.

Hromatski broj grafa G se dobija za najmanju vrednost r tako da je $V \in Q_r[G]$.

Primer

Neka je graf G=(V,E), gde $V=\{x_1, x_2, x_3, x_4\}$.
Tada je na primer za $S_1[G]$ možemo uzeti: $S_1[G]=\{x_1, x_3\}$.
Konačno dobijamo da je $S_1[\langle V - S_1 \rangle]=S_1[\{x_2, x_4\}]=\{x_2, x_4\}$.
Jedno pravilno bojenje grafa G tada je prikazana na slici 4.1.



Slika 4.5

4.3 Algoritam za određivanje svih maksimalnih nezavisnih podskupova

Algoritmi za određivanje hromatskog broja su složeniji, zbog toga prvo opisujemo pomoćni algoritam za određivanje svih maksimalnih nezavisnih podskupova koji ćemo koristiti za konstruisanje ostalih algoritama u ovoj glavi.

Nalaženje maksimalnih nezavisnih podskupova i najmanjeg dominantnog skupa u grafu su korisna sredstva kod izučavanja karakteristike grafa, odnosno kod ispitivanja obojivosti grafa. Na osnovu ovih skupova ćemo u daljem radu definisati probleme za određivanje hromatskog broja grafa. Pomoću sledećeg algoritma dobijamo sve maksimalne nezavisne skupove. Kasnije ćemo videti da do konačnog rešenja bojenja ćemo stići pomoću onih maksimalnih nezaviznih skupova koji su od tih najveći po kardinalnosti da bismo čvorove grafa mogli obojiti sa što manje boja.

Backtracking (korak unazad)

U algoritmu za traženje rešenja ćemo koristiti metod bektreking , gde se isprobavaju sve moguće kombinacije čvorova. Pri čemu se zaustavlja kod svih kandidata za koje se ispostavi da ne vode do tačnog rešenja. To možemo postići pogodnom definisanjem koraka algoritma. Algoritam prolazeći kroz sve moguće kombinacije daje sva moguća rešenja, pa i samim tim i traženo rešenje (maksimalan nezavisan skup). Bektreking garantuje pronalaženje svih rešenja za neki konačan problem, u određenom vremenskom roku.

Opis algoritma za određivanje svih maksimalnih nezavisnih podskupova:

Dat je skup V, skup čvorova grafa G. Konstruišimo listu S od čvorova skupa V, tako da skup čvorova iz S bude nezavisan. Čvorovi u S će biti u tom redosledu, kako smo ih ubacili u S. Pomoćni skupovi će biti skup Q^+ - pomoćna lista sa čvorovima koje još nismo obradili (u svakom koraku kada ubacimo u S jedan čvor, izbrišemo ga iz Q^+), i Q_k^-, k=1,2,... su liste koje sadrže čvorove koje smo već ispitivali u k-tom koraku. Za konstruisanje S krenemo od praznog skupa i u svakom koraku u S dodamo jedan čvor iz skupa $Q^+ \backslash(Q_1^- \cup Q_2^- \cup ...)$ tako da skup čvorova iz S ostane nezavisan. Kad dobijamo da je $Q^+ = \emptyset$, to će značiti da smo odradili sve čvorove iz V, odnosno da je S maksimalan nezavisan skup. Odštampamo ga i brišemo iz njega jedan čvor na osnovu backtrackinga. Umesto tog čvora ubacimo drugi čvor sve dok je moguće, da bismo tako dobili drugi maksimalan nezavisan skup.

Algoritam:

Korak 1. k:=0, S:= $\emptyset$, Q^+:=V, Q_k^-:= $\emptyset$.

Korak 2. i) Sve dok je $Q^+\backslash(Q_1^- \cup Q_2^- \cup \ldots) \neq \emptyset$, uzmemo $v_{i_{k+1}} \in Q^+\backslash Q_k^-$. Stavimo S:= S $\cup \{v_{i_{k+1}}\}$, Q^+:=$Q^+\backslash\{v_{i_{k+1}}\}\backslash$ N($v_{i_{k+1}}$) , Q_{k+1}^-:=$Q_{k+1}^- \cup \{v_{i_{k+1}}\}$, k:=k+1. Pređemo na korak 3.

ii) Ako je $Q^+\backslash(Q_1^- \cup Q_2^- \cup \ldots)=\emptyset$, pređemo na korak 4.

Korak 3. i) Sve dok je $Q^+\backslash(Q_1^- \cup Q_2^- \cup \ldots) \neq \emptyset$, pređemo na korak 2.

ii) Ako je $Q^+ = \emptyset$, odštampamo maksimalan nezavisan skup i pređemo na korak 4.

iii) Ako je $Q^+ \neq \emptyset$, a $Q^+\backslash(Q_1^- \cup Q_2^- \cup ...) =\emptyset$, pređemo na korak 4.

Korak 4. k:=k-1, S:=S$\backslash\{v_{i_{k+1}}\}$, za Q^+ vraćamo vrednost koju je imao za k=k-1, i stavimo Q_{k+2}^-=$\emptyset$.

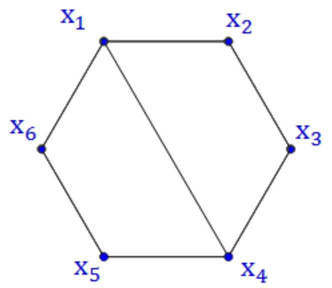
i) Ako je k=-1, kraj algoritma. Dobili smo sve maksimalne nezavisne skupove.

ii) Inače, pređemo na korak 2.

<u>Primer</u>

Neka je graf G=(V,E) dat na slici 4.4.

Skup čvorova graf je: $V=\{x_1, x_2, x_3, x_4, x_5, x_6\}$.



Slika 4.4

Korak 1. $S:= \emptyset$, $Q^+:=V$, $Q_0^-:= \emptyset$.

Korak 2. i) $x_1 \in Q^+\backslash Q_1^-$, $S= S \cup \{x_1\}=\{x_1\}$,

$Q^+=Q^+\backslash\{x_1\}\backslash N(x_1)=\{x_3, x_5\}$, $Q_1^-=\{x_1\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_3 \in Q^+\backslash Q_1^-$, $S=\{x_1, x_3\}$, $Q^+=\{x_5\}$, Q_1^- se ne menja, $Q_2^-=\{x_3\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_5 \in Q^+\backslash Q_1^- \cup Q_2^-$, $S=\{x_1, x_3, x_5\}$, $Q^+=\emptyset$, $Q_3^-=\{x_5\}$.

Korak 3. ii) $Q^+= \emptyset$, odštampamo maksimalan nezavisan skup $S=\{x_1, x_3, x_5\}$ i pređemo na korak 4.

Korak 4. k=2, $S=S\backslash\{x_5\}=\{x_1, x_3\}$, $Q^+=\{x_5\}$; Q_1^-, Q_2^-, Q_3^- ostaje.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=1, $S=S\backslash\{x_3\}=\{x_1\}$, $Q^+=\{x_3, x_5\}$, $Q_3^-= \emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_5 \in Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)$, $S=\{x_1, x_5\}$, $Q^+=\{x_3\}$, $Q_2^-=\{x_3, x_5\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, $S=S\backslash\{x_5\}=\{x_1\}$, $Q^+= \{x_3, x_5\}$.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=0, $S=\emptyset$, $Q^+= V$, $Q_1^-=\{x_1\}$, $Q_2^-=\emptyset$, $Q_3^-=\emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_2 \in Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)$, $S=\{x_2\}$, $Q^+=\{x_4, x_5, x_6\}$, $Q_1^-=\{x_1, x_2\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_4 \in Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)$, $S=\{x_2, x_4\}$, $Q^+=\{x_6\}$, Q_1^- se ne menja, $Q_2^-=\{x_4\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_6 \in Q^+\backslash(Q_1^- \cup Q_2^- \cup Q_3^-)$, $S=\{x_2, x_4, x_6\}$, $Q^+=\emptyset$, $Q_3^-=\{x_6\}$.

Korak 3. ii) $Q^+ = \emptyset$, odštampamo maksimalan nezavisan skup S=$\{x_2, x_4, x_6\}$ i pređemo na korak 4.

Korak 4. k=2, S=S$\setminus\{x_6\}$=$\{x_2, x_4\}$, Q^+=$\{x_6\}$; Q_1^-, Q_2^-, Q_3^- ostaje.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\setminus\{x_4\}$=$\{x_2\}$, Q^+=$\{x_4, x_5, x_6\}$, $Q_3^- = \emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_5 \in Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_2, x_5\}$, $Q^+=\emptyset$, Q_2^- =$\{x_4, x_5\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\setminus\{x_5\}$=$\{x_2\}$, $Q^+ = \{x_4, x_5, x_6\}$.

ii) Pređemo na korak 2.

Korak 2. i) $x_6 \in Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_2, x_6\}$, $Q^+=\emptyset$, Q_2^- =$\{x_4, x_5, x_6\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\setminus\{x_6\}$=$\{x_2\}$, $Q^+ = \{x_4, x_5, x_6\}$.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=0, S=$\emptyset$, Q^+= V, Q_1^- =Q_1^-=$\{x_1, x_2\}$, Q_2^- =$\emptyset$, Q_3^- =$\emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_3 \in Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_3\}$, Q^+=$\{x_1, x_5, x_6\}$, Q_1^-=$\{x_1, x_2, x_3\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_5 \in Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)$, S =$\{x_3, x_5\}$, $Q^+=\emptyset$, Q_1^- se ne menja, Q_2^- =$\{x_5\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\setminus\{x_5\}$=$\{x_3\}$, Q^+=$\{x_1, x_5, x_6\}$.

ii) Pređemo na korak 2.

Korak 2. i) $x_6 \in Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)$, S =$\{x_3, x_6\}$, $Q^+=\emptyset$, Q_1^- se ne menja, Q_2^- =$\{x_5, x_6\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\setminus\{x_6\}$=$\{x_3\}$, Q^+=$\{x_1, x_5, x_6\}$.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+\setminus(Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=0, S=$\emptyset$, Q^+= V, Q_1^- =$\{x_1, x_2, x_3\}$, Q_2^- =$\emptyset$, Q_3^- =$\emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_4 \in Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_4\}$, $Q^+=\{x_2, x_6\}$, $Q_1^-=\{x_1, x_2, x_3, x_4\}$.

Korak 3. i) $Q^+ \neq \emptyset$, pređemo na korak 2.

Korak 2. i) $x_6 \in Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)$, S =$\{x_4, x_6\}$, $Q^+=\emptyset$, Q_1^- se ne menja, Q_2^- =$\{x_6\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=1, S=S$\backslash\{x_6\}$=$\{x_4\}$, $Q^+=\{x_2, x_6\}$.

ii) Pređemo na korak 2.

Korak 2. ii) $Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=0, S=$\emptyset$, Q^+= V, $Q_1^-=\{x_1, x_2, x_3, x_4\}$, Q_2^- =$\emptyset$, Q_3^- =$\emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_5 \in Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_5\}$, $Q^+=\{x_1, x_2, x_3\}$, $Q_1^-=\{x_1, x_2, x_3, x_4, x_5\}$.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=0, S=$\emptyset$, Q^+= V, $Q_1^-=\{x_1, x_2, x_3, x_4, x_5\}$, Q_2^- =$\emptyset$, Q_3^- =$\emptyset$.

ii) Pređemo na korak 2.

Korak 2. i) $x_6 \in Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)$, S=$\{x_6\}$, $Q^+=\{x_2, x_3, x_4\}$, Q_1^-=V.

Korak 3. ii) $Q^+ \neq \emptyset$, ali $Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-) = \emptyset$, pređemo na korak 4.

Korak 4. k=0, S=$\emptyset$, Q^+= V, Q_1^-=V, Q_2^- =$\emptyset$, Q_3^- =$\emptyset$.

ii) Pređemo na korak 2.

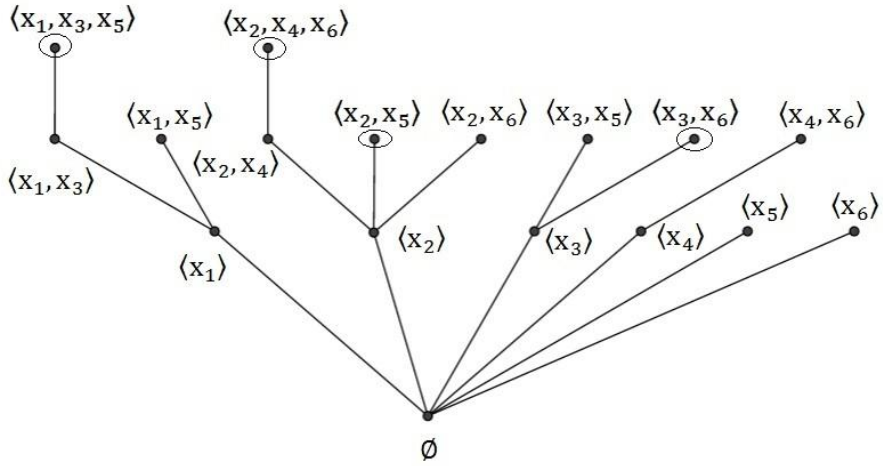
Korak 2. ii) $Q^+ \backslash (Q_1^- \cup Q_2^- \cup Q_3^-)=\emptyset$, pređemo na korak 4.

Korak 4. k=-1, kraj algoritma.

U sledećoj tablici su prikazani koraci algoritma i označeni maksimalni nezavisni skupovi grafa G.

k	S	Q^+	Q_1^-	Q_2^-	Q_3^-
0	$\emptyset$	V	$\emptyset$	$\emptyset$	$\emptyset$
1	$\langle x_1 \rangle$	$\{x_3, x_5\}$	$\{x_1\}$	$\emptyset$	$\emptyset$
2	$\langle x_1, x_3 \rangle$	$\{x_5\}$	$\{x_1\}$	$\{x_3\}$	$\emptyset$
3	$\langle x_1, x_3, x_5 \rangle$	$\emptyset$	$\{x_1\}$	$\{x_3\}$	$\{x_5\}$
2	$\langle x_1, x_3 \rangle$	$\{x_5\}$	$\{x_1\}$	$\{x_3\}$	$\{x_5\}$
1	$\langle x_1 \rangle$	$\{x_3, x_5\}$	$\{x_1\}$	$\{x_3\}$	$\emptyset$
2	$\langle x_1, x_5 \rangle$	$\{x_3\}$	$\{x_1\}$	$\{x_3, x_5\}$	$\emptyset$
1	$\langle x_1 \rangle$	$\{x_3, x_5\}$	$\{x_1\}$	$\{x_3, x_5\}$	$\emptyset$
0	$\emptyset$	V	$\{x_1\}$	$\emptyset$	$\emptyset$
1	$\langle x_2 \rangle$	$\{x_4, x_5, x_6\}$	$\{x_1, x_2\}$	$\emptyset$	$\emptyset$
2	$\langle x_2, x_4 \rangle$	$\{x_6\}$	$\{x_1, x_2\}$	$\{x_4\}$	$\emptyset$
3	$\langle x_2, x_4, x_6 \rangle$	$\emptyset$	$\{x_1, x_2\}$	$\{x_4\}$	$\{x_6\}$
2	$\langle x_2, x_4 \rangle$	$\{x_6\}$	$\{x_1, x_2\}$	$\{x_4\}$	$\{x_6\}$
1	$\langle x_2 \rangle$	$\{x_4, x_5, x_6\}$	$\{x_1, x_2\}$	$\{x_4\}$	$\emptyset$
2	$\langle x_2, x_5 \rangle$	$\emptyset$	$\{x_1, x_2\}$	$\{x_4, x_5\}$	$\emptyset$
1	$\langle x_2 \rangle$	$\{x_4, x_5, x_6\}$	$\{x_1, x_2\}$	$\{x_4, x_5\}$	$\emptyset$
2	$\langle x_2, x_6 \rangle$	$\{x_4\}$	$\{x_1, x_2\}$	$\{x_4, x_5, x_6\}$	$\emptyset$
1	$\langle x_2 \rangle$	$\{x_4, x_5, x_6\}$	$\{x_1, x_2\}$	$\{x_4, x_5, x_6\}$	$\emptyset$
0	$\emptyset$	V	$\{x_1, x_2\}$	$\emptyset$	$\emptyset$
1	$\langle x_3 \rangle$	$\{x_1, x_5, x_6\}$	$\{x_1, x_2, x_3\}$	$\emptyset$	$\emptyset$
2	$\langle x_3, x_5 \rangle$	$\{x_1\}$	$\{x_1, x_2, x_3\}$	$\{x_5\}$	$\emptyset$
1	$\langle x_3 \rangle$	$\{x_1, x_5, x_6\}$	$\{x_1, x_2, x_3\}$	$\{x_5\}$	$\emptyset$
2	$\langle x_3, x_6 \rangle$	$\emptyset$	$\{x_1, x_2, x_3\}$	$\{x_5, x_6\}$	$\emptyset$
1	$\langle x_3 \rangle$	$\{x_1, x_5, x_6\}$	$\{x_1, x_2, x_3\}$	$\{x_5, x_6\}$	$\emptyset$
0	$\emptyset$	V	$\{x_1, x_2, x_3\}$	$\emptyset$	$\emptyset$
1	$\langle x_4 \rangle$	$\{x_2, x_6\}$	$\{x_1, x_2, x_3, x_4\}$	$\emptyset$	$\emptyset$
2	$\langle x_4, x_6 \rangle$	$\{x_2\}$	$\{x_1, x_2, x_3, x_4\}$	$\{x_4\}$	$\emptyset$
1	$\langle x_4 \rangle$	$\{x_2, x_6\}$	$\{x_1, x_2, x_3, x_4\}$	$\{x_4, x_6\}$	$\emptyset$
0	$\emptyset$	V	$\{x_1, x_2, x_3\}$	$\emptyset$	$\emptyset$
1	$\langle x_5 \rangle$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\emptyset$	$\emptyset$
0	$\emptyset$	V	$\{x_1, x_2, x_3, x_4, x_5\}$	$\emptyset$	$\emptyset$
1	$\langle x_6 \rangle$	$\{x_2, x_3, x_4\}$	V	$\emptyset$	$\emptyset$
0	$\emptyset$	V	V	$\emptyset$	$\emptyset$
-1	Kraj				

Algoritam možemo prikazati i u obliku stabla pretraživanja koja vizuelno prikazuje korak po korak rad algoritma. Nezavisne skupove grafa G prikazujemo čvorovima stabla. Kao i u algoritmu krenemo od praznog skupa i za svaki novi dobijeni skup S dodajemo jedan čvor u stablo. U ovom primeru ćemo imati stablo pretraživanja veličine 17 i dubine 3, sa korenom $\emptyset$ i listovima $\langle x_1, x_3, x_5\rangle$, $\langle x_1, x_5\rangle$, $\langle x_2, x_4, x_6\rangle$, $\langle x_2, x_5\rangle$, $\langle x_2, x_6\rangle$, $\langle x_3, x_5\rangle$, $\langle x_3, x_6\rangle$, $\langle x_4, x_6\rangle$, $\langle x_5\rangle$, i $\langle x_6\rangle$ (slika 4.5). Zaokruženi listovi su rešenja algoritma, to jest maksimalni nezavisni skupovi početnog grafa koje smo našli tokom algoritma.



(Slika 4.5)

4.4 Algoritam svođenjem na r-nezavisne skupove

Hromatski broj grafa G=(V,E) je najmanja vrednost r tako da je skup čvorova $S_r[G]$ nekog maksimalnog r-pografa ceo skup čvorova grafa G. Pomoću rekurentne relacije možemo generisati maksimalne 1-podgrafove, 2-podgrafove i tako dalje, a u svakom koraku proveriti da li neki od podgrafova sadrži sve čvorove grafa G.

Koraci algoritma pomoću r-nezavisne podskupove za nalaženje hromatskog broja G i bojenja korišćenjem nađenog hromatskog broja su sledeći:

Korak 1. Neka je r = 1. Naći skup čvorova $S_r^j[G]$, j = 1, ..., q_r maksimalnog r-podgrafa od G (ukoliko imamo q_r takvih skupova). Neka je Q_r [G]= $\{S_j^r\ [G]\ \mid j = 1, \ldots, q_r\}$, i neka je j=1.

Korak 2. Naći maksimalne nezavisne skupove $S_1^i[G_r^j]$ grafa G^j=$\langle V - S_r^j[G]\rangle$. Ako takav postoji, preći na korak 3, a ako smo već našli sve takve skupove preći na korak 6.

Korak 3. Izračunati S= $S_r^j[G] \cup S_1^i[G_r^j]$.

Korak 4. Ako je S=V, stop. Hromatski broj je r+1. Ako je S≠V, preći na korak 5.

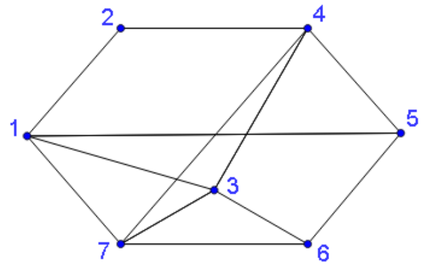
Korak 5. i) Ako je S ⊆ S' za neko S'∈Q_r[G] vratiti se na korak 2.
ii) Ako je S ⊃ S' za neko S'∈Q_r[G], staviti Q_r[G]=Q_r[G] - $\{S'\}$ za svako S'. Staviti Q_r[G]=Q_r[G] ∪ $\{S\}$ i vratiti se na korak 2.

iii) Ako uslovi ni u i) ni u ii) nisu zadovoljeni, staviti $Q_r[G]=Q_r[G] \cup \{S\}$ i vratiti se na korak 2.

Korak 6. Ako je $j<q_r$ staviti $j:=j+1$ i preći na korak 2. Ako je $j=q_r$, staviti $r:=r+1$, gde je q_r broj skupova u $Q_r[G]$, i preći na korak 2.

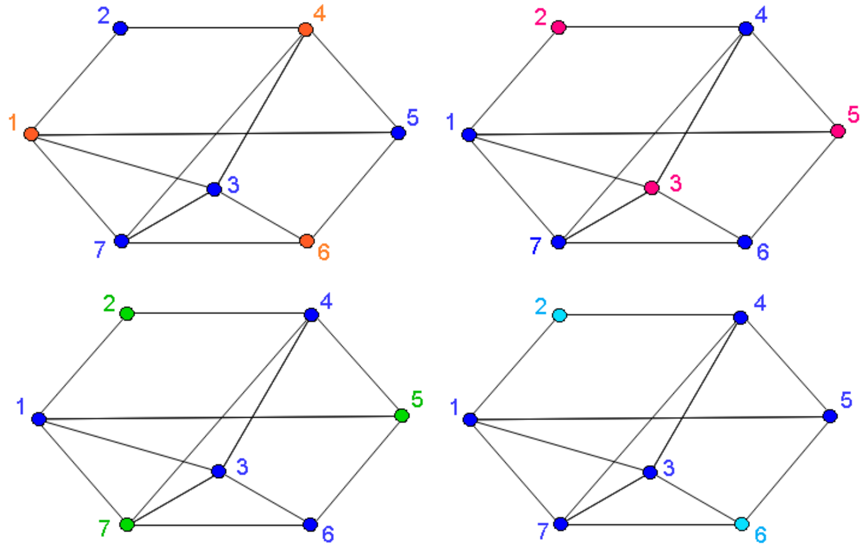
Ako se algoritam ne završava prvi put kada dobijamo S=V u koraku 4, nastavljamo sa alternativnim bojenjem sa r+1 boje, ukoliko takvo pravilno bojenje postoji.
Iako ovim algoritmom ne dobijamo listu svih mogućih bojenja sa r+1 boja, rezultovaće nam pravilno bojenje grafa.

Primer



Slika 4.1

Neka je graf G dat na slici 4.1. Maksimalni nezavisni skupovi čvorova su obojeni na slici 4.2.



Slika 4.2

Korak 1. Maksimalni 1-podgrafovi grafa su:

$S_1^1[G]=\{1,4,6\}$ (Slika 4.2 a)), $S_1^2[G]=\{2,3,5\}$, $S_1^3[G]=\{2,5,7\}$, $S_1^4[G]=\{2,6\}$.

Dobijamo $q_1=4$ i $Q_1[G]=\{S_1^1[G], S_1^2[G], S_1^3[G], S_1^4[G]\}$.

Za $S_1^1[G]$:

Korak 2. $G_1^1=\langle V - S_1^1[G]\rangle = \langle\{2,3,5,7\}\rangle$, $S_1^1[G_1^1] = S_1^1[V - S_1^1[G]] = \{2,3,5\}$

Korak 3. S= $S_1^1[G] \cup S_1^1[G_1^1] = \{1,4,6|2,3,5\}$
Korak 5. $Q_2[G]$= $Q_2[G] \cup \{S'\}$ = $[\{1,4,6|2,3,5\}]$

Korak 2. $S_1^2[G^1] = \{2,5,7\}$
Korak 3. $S = \{1,4,6|2,5,7\}$
Korak 5. $Q_2[G]$= $[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}]$

Za $\mathbf{S_1^2[G]}$:
Korak 2. G_1^2=$\langle V - S_1^2[G]\rangle$ = $\langle\{1,4,6,7\}\rangle$, $S_1^1[G_1^2] = S_1^1[V - S_1] = \{1,4,6\}$
Korak 3. S= $S_1^2[G] \cup S_1^1[G_1^2] = \{2,3,5|1,4,6\}$
Korak 5. Pošto je $\{2,3,5|1,4,6\} \subseteq \{1,4,6|2,3,5\}$, zbog 5. i) $Q_2[G]$ se ne menja:
$Q_2[G]$=$[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}]$

Korak 2. $S_1^2[G_1^2] = \{7\}$
Korak 3. $S = \{2,3,5|7\}$
Korak 5. $Q_2[G]$ = $[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}]$

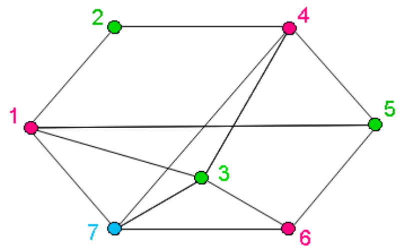
Za $\mathbf{S_1^3[G]}$:
Korak 2. G_1^3=$\langle V - S_1^3[G]\rangle$ = $\langle\{1,3,4,6\}\rangle$, $S_1^1[G_1^3] = \{1,4,6\}$
Korak 3. S= $S_1^3[G] \cup S_1^1[G_1^3] = \{2,5,7|1,4,6\}$
Korak 5. Slično, kao malopre: pošto je $\{2,5,7|1,4,6\} = \{1,4,6|2,5,7\}$, i zbog 5. i)
$Q_2[G]$=$[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}]$

Korak 2. G_1^3= $\langle\{1,3,4,6\}\rangle$, $S_1^2[G_1^3] = \{3\}$
Korak 3. $S = \{2,5,7|3\}$
Korak 5. Pošto $\{2,5,7|3\}$ i $\{1,4,6|2,5,7\}$ daju isto bojenje ($\{1,4,6|2,5,7\} \in Q_2[G]$), zbog 5. i)
Q se neće promeniti: $Q_2[G]$=$[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}]$

Za $\mathbf{S_1^4[G]}$:
Korak 2. G_1^4=$\langle V - S_1^4[G]\rangle$ = $\langle\{1,3,4,5,7\}\rangle$, $S_1^1[G_1^4] = \{1,4\}$
Korak 3. $S = \{2,6|1,4\}$
Korak 5. $Q_2[G]$ = $[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}, \{2,6|1,4\}]$

Korak 2. $G^4 = \langle\{1,3,4,5,7\}\rangle$, $S_1^2[G_1^4] = \{3,5\}$
Korak 3. $S = \{2,6|3,5\}$
Korak 5. $Q_2[G]$ = $[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}, \{2,6|1,4\}, \{2,6|3,5\}]$

Korak 2. G^4= $\langle\{1,3,4,5,7\}\rangle$, $S_1^3[G_1^4] = \{1,4\}$
Korak 3. $S = \{2,6|5,7\}$
Korak 5. Q_2 = $[\{1,4,6|2,3,5\}, \{1,4,6|2,5,7\}, \{2,3,5|7\}, \{2,6|1,4\}, \{2,6|3,5\}, \{2,6|5,7\}]$
Konačno, dobili smo familiju $Q_2[G]$ – familiju skupova $S_2^j[G]$. Ovi skupovi su maksimalni 2-podgrafovi od G i određuju pravilno bojenje podgrafova od G. Sada pređemo na nalaženje maksimalne 3-podgrafove.
Korak 1. $S_2^1[G]$=$\{1,4,6|2,3,5\}$, $Q_2[G]$ =$[\{1,4,6|2,3,5\}]$.
Korak 2. $S_1^1[G_1^1]$ = $S_1^1\langle V - S_2^1[G]\rangle$ = $S_1^1\langle\{7\}\rangle$ = $\{7\}$
Korak 3. S= $S_3^1[G] \cup S_1^1[G_1^1]$=$[\{1,4,6|2,3,5|7\}]$
Korak 4. S=V, stop.



Slika 4.3

Dobili smo da je hromatski broj grafa G, χ(G)=3. Ako bismo nastavljali traženje 3-podgrafova, dobili bismo ili ekvivalentne skupove sa $S_3^1[G]$ ili sa $S_3^2[G]$, ili ne bismo pokrivali sve čvorove grafa G.
Jedan od pravilnih bojenja ćemo dobiti tako što prvo jednom bojom obojimo čvorove 1, 4 i 6, drugom bojom čvorove 2, 3 i 5, a trećom bojom čvor 7 (Slika 4.3).

4.5 Set Covering Problem

Drugi problem za bojenje grafova je Set Covering Problem (SCP). To je pokrivanje čvorova sa maksimalnim nezavisnim skupovima.

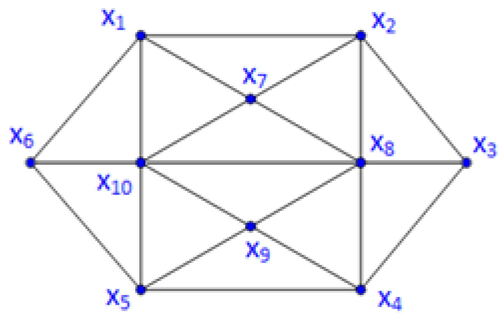
Kao što smo videli kod prethodnog algoritma, da bi bilo izvodljivo bojenje grafa G, skup čvorova obojenih istom bojom mora da formira nezavisan skup. Dakle, svako pravilno bojenje grafa se može smatrati kao podela čvorova grafa u takve skupove. Tako gledajući, bojenje grafa je pokrivanje čvorova grafa maksimalnim nezavisnim skupovima, odnosno proširenje nezaviznih skupova do maksimalnih takvih skupova.

Neka je G=(V,E), $|V|$=n, i neka je t broj maksimalnih nezavisnih skupova u G.
Definišimo matricu M_{nxt} = $[m_{ij}]$, tako da kolone matrice određuju maksimalne nezavisne skupove grafa G. Elementi matrice su definisani na sledeći način:

$$m_{ij} = \begin{cases} 1, & \text{ako je čvor } x_i \text{ sadržan u } j-\text{tom maksimalnom skupu} \\ 0, & \text{inače} \end{cases}$$

Primer 1.

Neka je graf G=(V,E) dat na slici 4.4.



Slika 4.4

Maksimalni nezavisni skupovi su:

$S_1^1[G]$={x_1, x_3, x_5}, $S_1^2[G]$={x_1, x_3, x_9},
$S_1^3[G]$={x_1, x_4}, $S_1^4[G]$={x_1, x_5, x_8},
$S_1^5[G]$={x_2, x_4, x_6}, $S_1^6[G]$={x_2, x_4, x_{10}}, $S_1^7[G]$={x_2, x_5},
$S_1^8[G]$={x_2, x_6, x_9}, $S_1^9[G]$={x_3, x_5, x_7}, $S_1^{10}[G]$={x_3, x_6, x_7, x_9}, $S_1^{11}[G]$={x_3, x_{10}},
$S_1^{12}[G]$={x_4, x_6, x_7}, $S_1^{13}[G]$={x_6, x_8}.

Na osnovu tih skupova možemo napraviti matricu M kao što smo definisali u prethodnoj definiciji:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Problem bojenja svodi se na problem nalaženja najmanjeg broja kolona koji pokrivaju sve čvorove.

Neka su sa k_i označeni kolone matrice, i neka je T_i unija kolona u kojima u i vrsta imamo jedinice. Tada je na primer:
$k_1 \cup k_2 = T_4^1$

Ako krenemo od prve kolone i redom dodamo u uniju sledeću kolonu, vidimo da će nam trebati 9 kolona da bismo pokrivali sve vrte sa jedinicama:

$$\begin{array}{cc} & \begin{array}{ccccccccccccc} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 & k_7 & k_8 & k_9 & k_{10} & k_{11} & k_{12} & k_{13} \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\ & \begin{array}{ccccccccccccc} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & & & & \end{array} \end{array}$$

$T_{10}^1 = k_1 \cup k_2 \cup k_3 \cup k_4 \cup k_5 \cup k_6 \cup k_7 \cup k_8 \cup k_9 = \bigcup_{i=1}^{9} k_i$

Odmah vidimo da u ovoj uniji postoje kolone koje možemo izostaviti tako da sve vrste matrice ostaju pokrivene sa jedinicama. Na primer (ako idemo redom sa izostavljanjem):

$T_{10}^2 = \bigcup_{i=1}^{9} k_i \setminus \{ k_2, k_3, k_5, k_7\} = k_1 \cup k_4 \cup k_6 \cup k_8 \cup k_9$.

Za odrđivanje hromatskog broja grafa tražimo najmanji broj nezavisnih skupova koji je dovoljno za pokrivanje svih čvorova grafa, odnosno pokrivanje sve vrste matrice.

Algoritam za pokrivanje matrice nezavisnih skupova

Slično kao u algoritmu za određivanje svih maksimalnih nezavisnih podskupova, i u ovom algoritmu pretražujemo sve moguće kombinacije unije skupova da bismo našli rešenje. Listom S ćemo označiti listu kolona matrice M koja će pokrivati vrste matrice. U početku S je prazan, pa redom u svakom koraku dodamo po jednu kolonu koju ćemo izabrati tako da sadrži bar jedan još ne pokriven čvor x_i. Pomoćni skupovi će nam biti skup čvorova A_k (koje pokrivaju prvih k skupova iz S), odnosno skupovi $Q_{\bar{k}}$ (za svako k=1, … n, skup do k-tog koraka izabranih kolona koji sadrži k-ti čvor). Pored toga, skup Min će označiti kardinalni broj najmanjeg nađenog lista S koja pokriva sve čvorove grafa.

Korak 1. $k:=0$, $S:=\emptyset$, $A_k:=\emptyset$, $Min:=\emptyset$, $Q_{\bar{k}}:=\emptyset$.

Korak 2. i) Sve dok za $x_i \in V \backslash A_k$ postoji $k_j \notin Q_{\bar{k}}$ tako da $x_i \in k_j$, stavimo $S:=S \cup \{k_j\}$, $Q_{\bar{k}}:=Q_{\bar{k}} \cup \{k_j\}$, $k:=k+1$. Ako $A_k=V$, $Min:=\min\{Min, |S|\}$, pređemo na korak 3.

ii) Ako $V \backslash A_k \neq \emptyset$, a ne možemo naći kolonu tako da $x_i \in k_j$ ($k_j \notin Q_{\bar{k}}$), to znači da taj skup kolona ne vodi do traženog rešenja. Pređemo na korak 3.

Korak 3. Iz S brišemo poslednju kolonu k_j, i stavimo $A_k=\emptyset$, $Q_{\overline{k+1}}=\emptyset$, $k:=k-1$.

i) Ako je k=-1, kraj algoritma.

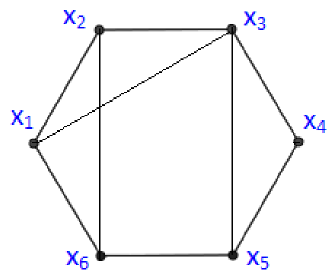
ii) Inače, vraćamo se na korak 2.

Primer 2.

Neka je graf G=(V,E), $V=\{x_1, x_2, x_3, x_4, x_5, x_6\}$, slika 4.5. Maksimalni nezavisni skupovi grafa G su:

$S_1^1[G]=\{x_1, x_4\}$, $S_1^2[G]=\{x_1, x_5\}$, $S_1^3[G]=\{x_2, x_4\}$,

$S_1^4[G]=\{x_2, x_5\}$, $S_1^5[G]=\{x_3, x_6\}$, $S_1^6[G]=\{x_4, x_6\}$,



Slika 4.5

Matrica maksimalnih nezavisnih skupova M grafa G je je tada:

$$M=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array}\begin{array}{c} \begin{matrix} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Pošto x_3 sadržan samo u jednom od skupova S_j^i, odnosno samo u jednoj koloni matrice (k_5), da bismo dobili rešenje za manje koraka, tu kolonu možemo izostaviti. Tada k_5 će biti sadržan u svakom od rešenja. Pošto k_5 sadrži čvorove x_3 i x_6, algoritam primenimo na skup čvorova $V'=V \backslash \{x_3, x_6\}$.

Matrica M' tada izgleda ovako:

$$M' = \begin{array}{c} \\ x_1 \\ x_2 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & k_3 & k_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Algoritam:

Korak 1. k:=0, S:=∅, A_k:=∅, Min:=∅, Q_k^- :=∅.

Korak 2. i) $x_1 \in V' \backslash A_1$, S:=S ∪{$k_1$} =$\langle k_1 \rangle$, A_1:= A_1∪{x_1, x_4}={x_1, x_4}, Q_1^- := Q_1^-∪{k_1}, k:=1.

Korak 2. i) $x_2 \in V' \backslash A_2$, S:=$\langle k_1, k_3 \rangle$, A_2:={x_1, x_2, x_4}, Q_2^- :={k_3}, k:=2.

Korak 2. i) $x_5 \in V' \backslash A_3$, S:=$\langle k_1, k_3, k_2 \rangle$, A_2:= V', Min:=3, Q_3^- :={k_2}, k:=3.

Korak 3. S:=$\langle k_1, k_3 \rangle$, A_3:=∅, k:=2.

Korak 2. i) $x_5 \in V' \backslash A_3$, S:=$\langle k_1, k_3, k_4 \rangle$, A_3:= V', Min:=3, Q_3^- :={k_2, k_4}, k:=3.

Korak 3. S:=$\langle k_1, k_3 \rangle$, A_3:=∅, A_2={x_1, x_2, x_4}, k:=2.

Korak 2. ii) $x_5 \in V' \backslash A_3$, a pošto ne postoji k_j iz M tako da $x_5 \in k_j$, a $k_j \notin Q_3^-$, pređemo na korak 3.

Korak 3. S:=$\langle k_1 \rangle$, A_2:=∅, Q_3^-=∅, k:=1.

Korak 2. i) $x_2 \in V' \backslash A_2$, S:=$\langle k_1, k_4 \rangle$, A_2:= V', Min:=2, Q_2^- :={k_3, k_4}, k:=2.

Korak 3. S:=$\langle k_1 \rangle$, A_2:=∅, k:=1.

Korak 2. Za x_2 ne postoji kolona $k_j \notin Q_2^-$ koja ga sadrži, pređemo na korak 3.

Korak 3. S:=∅, A_0:=∅, Q_2^- :=∅, k:=0.

Korak 2. i) $x_1 \in V' \backslash A_1$, S:=$\langle k_2 \rangle$, A_2:={x_1, x_5}, Q_1^- :={k_1, k_2}, k:=1.

Korak 2. i) $x_2 \in V' \backslash A_2$, S:=$\langle k_2, k_3 \rangle$, A_2:= V', Min:=2, Q_2^- :={k_3}, k:=2.

Korak 3. S:=$\langle k_2 \rangle$, A_2:=∅, k:=1.

Korak 2. i) $x_2 \in V' \backslash A_2$, S:=$\langle k_2, k_4 \rangle$, A_2:={x_1, x_2, x_5}, Q_2^- :={k_3, k_4}, k:=3.

Korak 2. i) $x_4 \in V' \backslash A_3$, S:=$\langle k_2, k_4, k_1 \rangle$, A_3:= V', a Min ostaje 2, Q_3^- :={k_1, k_4}, k:=3.

Korak 3. S:=$\langle k_2, k_4 \rangle$, A_3:=∅, A_2={x_1, x_2, x_5}, k:=2.

Korak 2. Za x_4 ne postoji kolona $k_j \notin Q_3^-$ koja ga sadrži, pređemo na korak 3.

Korak 3. S:=$\langle k_2 \rangle$, A_2:=∅, A_1={x_1, x_5}, Q_3^-=∅, k:=1.

Korak 2. Za x_2 ne postoji kolona $k_j \notin Q_2^-$ koja ga sadrži, pređemo na korak 3.

Korak 3. S:=∅, A_1:=∅, k:=0.

Korak 2. Za x_1 ne postoji kolona $k_j \notin Q_1^-$ koja ga sadrži, pređemo na korak 3.

Korak 3. S:=∅, A_1:=∅, k:=-1. Kraj algoritma.

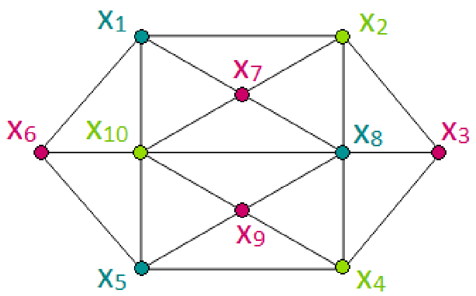
k	S	A_1	A_2	A_3	Min	Q_1^-	Q_2^-	Q_3^-
1	$\langle k_1 \rangle$	$\{x_1, x_4\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{k_1\}$	$\emptyset$	$\emptyset$
2	$\langle k_1, k_3 \rangle$	$\{x_1, x_4\}$	$\{x_1, x_2, x_4\}$	$\emptyset$	$\emptyset$	$\{k_1\}$	$\{k_3\}$	$\emptyset$
3	$\langle k_1, k_3, k_2 \rangle$	$\{x_1, x_4\}$	$\{x_1, x_2, x_4\}$	V’	3	$\{k_1\}$	$\{k_3\}$	$\{k_2\}$
3	$\langle k_1, k_3, k_4 \rangle$	$\{x_1, x_4\}$	$\{x_1, x_2, x_4\}$	V’	3	$\{k_1\}$	$\{k_3\}$	$\{k_2, k_4\}$
2	$\langle k_1, k_4 \rangle$	$\{x_1, x_4\}$	V’	$\emptyset$	2	$\{k_1\}$	$\{k_3, k_4\}$	$\emptyset$
1	$\langle k_2 \rangle$	$\{x_1, x_5\}$	$\emptyset$	$\emptyset$	2	$\{k_1, k_2\}$	$\emptyset$	$\emptyset$
2	$\langle k_2, k_3 \rangle$	$\{x_1, x_5\}$	V’	$\emptyset$	2	$\{k_1, k_2\}$	$\{k_3\}$	$\emptyset$
2	$\langle k_2, k_4 \rangle$	$\{x_1, x_5\}$	$\{x_1, x_2, x_5\}$	$\emptyset$	2	$\{k_1, k_2\}$	$\{k_3, k_4\}$	$\emptyset$
3	$\langle k_2, k_4, k_1 \rangle$	$\{x_1, x_5\}$	$\{x_1, x_2, x_5\}$	V’	2	$\{k_1, k_2\}$	$\{k_3, k_4\}$	$\{k_1\}$
3	$\langle k_2, k_4, k_3 \rangle$	$\{x_1, x_5\}$	$\{x_1, x_2, x_5\}$	V’	2	$\{k_1, k_2\}$	$\{k_3, k_4\}$	$\{k_1, k_3\}$
2	$\langle k_2, k_4 \rangle$	$\{x_1, x_5\}$	$\{x_1, x_2, x_5\}$	$\emptyset$	2	$\{k_1, k_2\}$	$\{k_3, k_4\}$	$\{k_1, k_3\}$
1	$\langle k_2 \rangle$	$\{x_1, x_5\}$	$\emptyset$	$\emptyset$	2	$\{k_1, k_2\}$	$\{k_3, k_4\}$	$\emptyset$
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	2	$\{k_1, k_2\}$	$\emptyset$	$\emptyset$

Na osnovu ovog algoritma vidimo da se sve vrste matrice M iz primera 1. može pokrivati i sa 3 kolone:

$$M = \begin{array}{ccccccccccccc} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 & k_7 & k_8 & k_9 & k_{10} & k_{11} & k_{12} & k_{13} \\ \end{array}$$
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$\begin{array}{ccccccccccccc} & & & \uparrow & & \uparrow & & & & \uparrow & & & \end{array}$$

Dakle, najmanji broj kolona koji je dovoljno za pokrivanje vrsta je 3, odnosno hromatski broj grafa G, $\chi(G)=3$. Pravilno bojenje grafa G možemo izvršiti tako što gledamo kolone koje smo koristili za pokrivanje: k_4, k_6, k_{10}.

Čvorove kod kojih su jedinice u istoj koloni, obojimo sa istim bojama, a čvorove sa jedinicama iz različite kolone će biti različite boje.



Slika 4.6

Ukoliko za neki čvor imamo jedinice u više kolona, tada od „boja kolona“ slobodno biramo boju kojom ćemo obojiti taj čvor. Dakle: $c_1=\{x_3, x_6, x_7, x_9\}$, $c_2=\{x_1, x_5, x_8\}$, $c_3=\{x_2, x_4, x_{10}\}$ (Slika 4.6).

Literatura

1. N. Christofides: Graph Theory, an Algorithmic Approach, Academic Press, 1975.
2. V. Petrović, Teorija grafova, Univerzitet u Novom Sadu, 1998.
3. I. Dolinka, Kratak uvod u analizu algoritama, Univerzitet u Novom Sadu, 2008.
4. R. A. Wilson, Graphs, Colourings and the Four-colour Theorem, Oxford University Press, 2002.
5. D. Vujanac, Bojenje grafova, master rad, PMF, 2015.

Biografija

Čisar Angela (rođena Pastor) je rođena u Senti 06. 08. 1989. godine. Završila je ekonomsku srednju školu u Senti 2008. godine i iste godine je počela studije na Prirodno-matematičkom fakultetu u Novom Sadu. 2014. godine je stekla diplomu kao diplomirani profesor matematike. U oktobru te godine je upisala master studije na istom fakultetu, smera master profesor matematike.

Novi Sad, 2016. 01. 08. Čisar Angela

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj: **RBR**	
Identifikacioni broj: **IBR**	
Tip dokumentacije: **TD**	Monografska dokumentacija
Tip zapisa: **TZ**	Tekstualni štampani materijal
Vrsta rada: **VR**	Master rad
Autor: **AU**	Angela Čisar
Mentor: **MN**	dr Boris Šobot
Naslov rada: **NR**	Algoritmi bojenja čvorova u grafovima
Jezik publikacije: **JP**	srpski (latinica)
Jezik izvoda: **JI**	srpski/engleski
Zemlja publikovanja: **ZP**	Republika srbija
Uže geografsko područje: **UGP**	Vojvodina
Godina: **GO**	2016
Izdavač: **IZ**	Autorski reprint
Mesto i adresa: **MA**	Trg Dositeja Obradovića 4, Novi Sad
Fizički opis rada: **FO**	4 poglavlja/39 strana/25 slika/4 tabele
Naučna oblast: **NO**	Matematika
Naučna disciplina: **ND**	Teorija algoritama
Predmetna odrednica/ ključne reči: **PO** **UDK**	bojenje čvorova u grafovima, hromatski broj, algoritmi, NP-kompletnost
Čuva se: **ČU**	Biblioteka departmana za fiziku, PMF-a u Novom Sadu
Važna napomena: **VN**	nema
Izvod: **IZ**	U radu su prikazani osnovne osobine grafova, osobine obojivosti i hromatskog broja, definicije klase složenosti P i NP. Prikazan je dokaz NP-kompletnosti problema k-obojivosti i nekoliko algoritama za nalaženje hromatskog broja datog grafa.
Datum prihvatanja teme od NN veća:	24.03.2015.

DP

Datum odbrane:
DO

Članovi komisije:
KO

Predsednik:	dr Vojislav Petrović, redovni profesor Prirodno-matematičkog fakulteta u Novom Sadu
član:	dr Igor Dolinka, redovni profesor Prirodno-matematičkog fakulteta u Novom Sadu
član:	dr Boris Šobot, docent Prirodno-matematičkog fakulteta u Novom Sadu

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCE AND MATHEMATICS

KEY WORDS DOCUMENTATION

Accession number: **ANO**	
Identification number: **INO**	
Document type: **DT**	Monograph publication
Type of record: **TR**	Textual printed material
Content code: **CC**	Final paper
Author: **AU**	Angela Čisar
Mentor/comentor: **MN**	dr Boris Šobot
Title: **TI**	Algorithms of coloring of vertices in graphs
Language of text: **LT**	Serbian (Latin)
Language of abstract: **LA**	English
Country of publication: **CP**	Serbia and Montenegro
Locality of publication: **LP**	Vojvodina
Publication year: **PY**	2005
Publisher: **PU**	Author's reprint
Publication place: **PP**	Faculty of Science and Mathematics, Trg Dositeja Obradovića 4, Novi Sad
Physical description: ***PD***	4 chapters/39 pages/25 pictures/4 tables
Scientific field: **SF**	Mathematics
Scientific discipline: **SD**	Theory of algorithms
Subject/ Key words: **SKW** **UC**	coloring vertices in graphs, chromatic number, algorithms, NP-completeness
Holding data: **HD**	Library of Department of Physics, Trg Dositeja Obradovića 4
Note: **N**	none
Abstract: **AB**	Basic features of graphs are presented in this paper, features of coloring and chromatic number, definitions of class complexity P and NP. Proof for NP-completeness of k-coloring problem is presented and a few algorithms for finding the chromatic number of the given graph.
Accepted by the Scientific Board: **ASB**	24 of January 2015

Defended on: **DE**	
Thesis defend board: **DB**	
President:	dr Vojislav Petrović, Full Professor, Faculty of Sciences, University of Novi Sad
Member:	dr Igor Dolinka, Full Professor, Faculty of Sciences, University of Novi Sad
Member:	dr Boris Šobot, Docent, Faculty of Sciences, University of Novi Sad