

# Nalaženje grafova sa zadatim spektrom pomoću metode promenljivih okolina

Petar Ćirković<sup>1</sup>, Predrag Đorđević<sup>1</sup>, Tatjana Davidović<sup>2</sup>

<sup>1</sup> Prirodno-matematički fakultet, Univerzitet u Nišu  
{petar.cirkovic,predrag.djordjevic}@pmf.edu.rs

<sup>2</sup> Matematički institut SANU, Knez-Mihailova 36/III, Beograd  
tanjad@mi.sanu.ac.rs

**Abstract.** Karakterizacija grafa na osnovu spektra je aktuelan istraživački problem koji ima brojne primene. Pokazano je da graf nije nužno jednoznačno određen svojim spektrom, tj. postoje neizomorfni kospektralni grafovi, ali je obično važno da se pronađe barem jedan graf čiji je spektar jednak datom. Taj proces se naziva spektralna rekonstrukcija grafa (Spectral Reconstruction of Graphs, SRG) i spada u NP-teške optimizacione probleme. U ovom radu je na problem spektralne rekonstrukcije primenjena osnovna varijanta metode promenljivih okolina (Basic Variable Neighborhood Search, BVNS) koja je implementirana tako da uzima u obzir neke poznate osobine grafa koje se mogu izvesti na osnovu odgovarajućeg spektra. Dobijena heuristika nazvana je SRG-BVNS. Rezultati rekonstrukcije nekih poznatih grafova primenom SRG-BVNS metode poređeni su sa rezultatima dobijenim primenom AutoGraphiX paketa. Ovaj paket takođe koristi neke varijante metode promenljivih okolina, ali korisniku nije dozvoljeno da utiče na njihov izbor i redosled.

**Ključne reči:** Spektralna teorija grafova; spektralno rastojanje; spektralna rekonstrukcija grafa; metaheuristike.

## 1 Uvod

Grafovi su matematički objekti koji se koriste za modeliranje raznih problema u inženjerstvu, nauci i industriji. Definišu se [9] kao uređeni parovi  $G = (V, E)$ , pri čemu je  $V = \{v_1, v_2, \dots, v_n\}$ , konačan skup čiji elementi  $v_i$  se nazivaju *čvorovi*, a  $E \subseteq V \times V$  predstavlja skup veza među čvorovima koje nazivamo *grane*. Ukoliko između čvorova  $v_i$  i  $v_j$  postoji grana, tj.  $\{v_i, v_j\} \in E$  kažemo da su ti čvorovi *susedni*. Graf se najjednostavnije zapisuje pomoću *matrice susedstva*  $A$  čiji su elementi 0 ili 1, tj.

$$a_{ij} = \begin{cases} 1, & \text{ukoliko } \{v_i, v_j\} \in E; \\ 0, & \text{inače.} \end{cases}$$

*Stepen čvora*  $v_i$  u grafu  $G$  predstavlja broj čvorova susednih čvoru  $v_i$ , tj. broj grana u kojima se taj čvor pojavljuje i označava se sa  $d_i$ . Ako se posmatra matrica susedstva  $A$ ,  $d_i = \sum_{j=1}^n a_{ij}$ , tj. stepen čvora  $v_i$  jednak je broju jedinica u  $i$ -toj vrsti matrice  $A$ .

*Sopstvene vrednosti*  $\lambda_i$ ,  $i = 1, 2, \dots, n$  grafa  $G$  su ustvari sopstvene vrednosti matrice  $A$ , tj. nule njenog karakterističnog polinoma  $P_G(x) = \det(xI - A)$ . Matrica susedstva  $A$  je simetrična pa će sopstvene vrednosti biti realni brojevi. Skup svih sopstvenih vrednosti grafa  $G$  naziva se *spektar*. Uobičajeno je da se spektar grafa predstavlja u monotono nerastućem poretku, tj.  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ . Najveća sopstvena vrednost  $\lambda_1$  grafa  $G$  naziva se *indeks*. Vektor  $x$  takav da važi  $Ax = \lambda x$  naziva se *sopstveni vektor* ( koji odgovara sopstvenoj vrednosti  $\lambda$ ) grafa  $G$ , odnosno matrice  $A$ .

Spektralna teorija grafova [7, 12, 20] proučava grafove na osnovu njihove matrice susedstva, preciznije na osnovu sopstvenih vrednosti i sopstvenih vektora ove matrice. U novijoj literaturi definišu se i analiziraju i neke druge matrice pridružene grafovima, npr. Laplasova i neoznačena Laplasova ([12], odeljak 1.3), ali one u ovom radu neće biti razmatrane. Spektralna teorija grafova ima značajne primene u različitim oblastima računarstva [14], uključujući i probleme prepoznavanja grafova [18] jer su grafovi prirodni modeli za razne vrste objekata. U literaturi je pokazano da su neke specijalne klase grafova, npr. kompletni grafovi, putevi, konture određeni spektrom u odnosu na matricu susedstva  $A$  do na izomorfizam. Međutim, dokazano je da se proizvoljni graf ne može u potpunosti okarakterisati svojim spektrom, tj. da postoje neizomorfni grafovi koji imaju isti spektar. Konkretno, pokazano je da se stabla ne mogu okarakterisati spektrom, kao ni molekuli u hemiji. Neizomorfni grafovi koji imaju identične spektre nazivaju se *kospektralni*. U [16, 21] analizira se broj neizomorfni kospektralnih grafova u odnosu na tri pomenute matrice za sve grafove sa  $n \leq 11$  čvorova. Grafovi sa  $n = 12$  čvorova razmatrani su u radu [4]. U navedenim radovima uočeno je da počev od  $n = 10$  broj grafova koji imaju neizomorfne kospektralne opada u odnosu na ukupan broj grafova sa istim brojem čvorova i postavljena je hipoteza da su grafovi sa velikim brojem čvorova ( $n \rightarrow \infty$ ) možda ipak određeni svojim spektrom. Hipoteza je još uvek otvoren problem spektralne teorije grafova.

Ovaj rad je inspirisan rezultatima objavljenim u [5] i predstavlja njihovu generalizaciju i proširenje. U prvom delu rada [5], autori su razmatrali problem spektralne rekonstrukcija grafova (Spectral Reconstruction of Graphs, SRG) [10] uz pomoć programskog paketa za rad sa grafovima AutoGraphiX (AGX) razvijenog na GERAD institutu u Montrealu [2, 6]. AGX koristi metodu promenljivih okolina (Variable Neighborhood Search, VNS) [17, 19] za nalaženje grafova koji imaju ekstremalne vrednosti izabranih invarijanata ili njihovih kombinacija. Mi smo optimizovali rad AGX paketa dodavanjem novih ograničenja čime se smanjuje prostor pretrage i brže dolazi do rezultata.

Imajući na umu da je AGX softver opšte namene, pa ima mnogo pomoćnih funkcija koje se ne koriste za razmatrani problem, implementirali smo osnovnu verziju VNS metode za nalaženje grafa sa zadatim spektrom. Metoda je nazvana SRG-BVNS. Stohastička priroda VNS metode omogućava nam da restartovanjem programa, počev od različitih slučajnih početnih grafova, generišemo međusobno neizomorfne kospektralne grafove (ukoliko postoje). Međutim, nalaženje svih kospektralnih grafova i dalje ostaje izazovan zadatak jer se svodi na

NP-težak optimizacioni problem. Naime, potrebno je pretražiti sve grafove sa zadatim brojem čvorova ( $n$ ) i grana ( $m$ ) kojih ima  $\binom{n(n-1)/2}{m}$ .

Ostatak rada je organizovan na sledeći način. U odeljku 2 dat je kratak pregled relevantne literature. Problem spektralne rekonstrukcije grafa detaljno je opisan u odeljku 3 pri čemu je dat i osvrt na složenost ovog problema i neke specijalne slučajeve kod kojih postoje efikasni algoritmi za nalaženje svih neizomorfni grafova kospektralnih sa zadatim. Odeljak 4 sadrži opis implementacije SRG-BVNS metode za nalaženje grafova zadatog spektra i rezultate dobijene primenom na neke poznate grafove iz literature. Zaključna razmatranja i smernice za budući rad date su u odeljku 5.

## 2 Pregled literature

Izučavanje grafova na osnovu njihovog spektra postalo je veoma popularno u prethodne dve decenije jer je spektar grafa moguće odrediti relativno brzo (složenost izračunavanja je u opštem slučaju  $O(n^3)$ , a za specijalne klase grafova ta složenost može da bude manja). Na osnovu spektra mogu se utvrditi razne informacije o strukturi odgovarajućeg grafa [10, 18], naročito o nekim parametrima grafa čije izračunavanje zahteva eksponencijalno vreme. Kao što je već navedeno u uvodu, grafovi nisu jednoznačno određeni svojim spektrom, tj. za neke grafove postoje njima neizomorfni kospektralni grafovi. Ipak, zbog velike važnosti, spektralno prepoznavanje grafova je oblast koja se trenutno intenzivno izučava u literaturi [10, 16, 21].

U preglednom radu [10] definisana su 4 osnovna problema koja se razmatraju u vezi sa spektralnim prepoznavanjem grafova: karakterizacija grafa sa zadatim spektrom; rekonstrukcija grafa sa zadatim spektrom; spektralna sličnost grafova; i spektralne perturbacije grafova. Napomenimo još jednom da su svi ovi problemi vezani za spektar grafa i da koriste *spektralno rastojanje*. Ono se definiše između grafova sa istim brojem čvorova kao rastojanje između njihovih spektara. Pri tome to rastojanje može biti Euklidsko, Menhetn ili neko drugo rastojanje između uređenih nizova njihovih sopstvenih vrednosti, tj. vektora u  $n$ -dimenzionom koordinatnom sistemu.

Prvi problem (karakterizacija grafa zadatog spektra) podrazumeva opisivanje što je moguće više njegovih osobina na osnovu tog spektra. Kod drugog problema potrebno je pronaći graf čiji je spektar predstavljen zadatim vektorom (a to je osnovna tema ovog rada). Karakterizacija grafa na osnovu spektra postiže se rešavanjem optimizacionog problema koji predstavlja minimizaciju spektralnog rastojanja između zadatog vektora i spektra grafa koji konstruišemo. Rešenje ovog problema ne mora biti jedinstveno, može se dobiti više kospektralnih grafova. Dakle, kospektralni grafovi se nalaze na rastojanju jednakom nuli. Spektralno rastojanje grafova može predstavljati meru sličnosti između grafova sa jednakim brojem čvorova, tj. kažemo da su dva grafa slična ako im je spektralno rastojanje malo. Slični grafovi dobijaju se jedni iz drugih malim perturbacijama (promenama u strukturi ili spektru grafa). Primeri perturbacija su uklanjanje ili dodavanje čvora ili grane, premeštanje grana i slično.

Jedan od prvih algoritama za spektralnu rekonstrukciju grafa na osnovu Laplasove matrice razvijen je u [8]. Baziran je na metaheuristici tabu pretraživanje i polazi od slučajnog grafa sa  $n$  čvorova i koristi spektralno rastojanje čija minimizacija omogućava rekonstrukciju raznih grafova (mreža). Algoritam je testiran na raznim klasama mreža (slučajnim, regularnim, klaster-grafovima i sl.).

Metoda promenljivih okolina korišćena je u [5] indirektno, kroz AGX programski paket. Autori su se bavili rekonstrukcijom nekih klasa grafova na osnovu euklidskog i Menhetn spektralnog rastojanja definisanog u odnosu na razne matrice pridružene grafu. Analizirani su grafovi do 20 čvorova i prijavljivan je broj uspešnih rekonstrukcija u 100 ponavljanja. Kriterijum zaustavljanja u jednom izvršavanju bio je 100 000 evaluacija funkcije cilja, a polazilo se od slučajno generisanog grafa.

Rad [5] poslužio je kao inspiracija za implementaciju predloženu u ovom radu. Cilj je da se maksimalno iskoriste informacije koje se o traženom grafu mogu dobiti na osnovu njegovog spektra i da se razvije efikasna implementacija metode koja će za što kraće vreme generisati što je moguće više međusobno neizomorfni kospektralni grafova.

### 3 Nalaženje grafa sa zadatim spektrom

Kao što je već navedeno, spektralna rekonstrukcija grafa podrazumeva nalaženje (jednog ili više) grafova čiji spektar je jednak zadatom vektoru. U ovom radu korišćemo Euklidsko rastojanje, kao što je to slučaj i u radu [5]. Neka je  $C = (c_1, c_2, \dots, c_n)$  zadati vektor, neka je  $G = (V, E)$  graf koji ima  $n$  čvorova, a  $S = (\lambda_1, \lambda_2, \dots, \lambda_n)$  njegov spektar. Potrebno je vršiti transformacije grafa  $G$  tako da se svede na nulu spektralno rastojanje definisano sa

$$D = \sqrt{(c_1 - \lambda_1)^2 + (c_2 - \lambda_2)^2 + \dots + (c_n - \lambda_n)^2}.$$

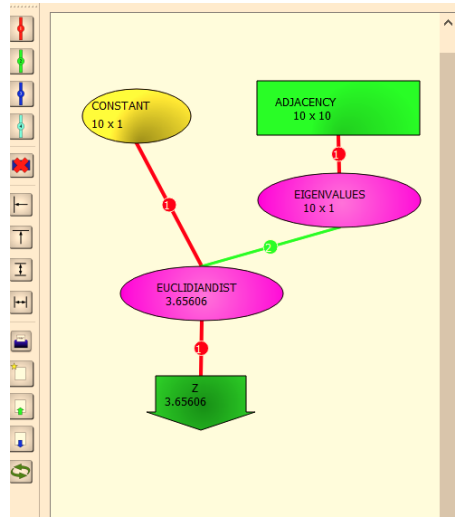
Za to smo prvo koristili AGX softver [2, 6]. To je interaktivni softverski paket namenjen za pronalaženje ekstremalnih grafova, tj. grafova koji minimizuju ili maksimizuju neku grafovsku invarijantu ili neku funkciju grafovskih invarijanti. Grafovska invarijanta je parametar grafa koji je nezavisan od redosleda numeracije čvorova i grana grafa. Grafovske invarijante su, na primer, indeks grafa (tj. najveća sopstvena vrednost  $\lambda_1$ ), minimalni  $\delta$  i maksimalni  $\Delta$  stepen čvora u grafu, hromatski broj  $\xi$  (minimalni broj boja potrebnih za bojenje čvorova grafa tako da nikoja dva susedna čvora ne budu iste boje), itd [9, 12]. U potrazi za ekstremalnim grafovima AGX koristi moduo za optimizaciju koji je zasnovan na VNS metaheuristici. Kada se dobiju odgovarajući grafovi u razmatranim specijalnim slučajevima (za neke zadate vrednosti  $n$ ), moguće je postaviti hipoteze za opšti slučaj i onda pokušati da se one teorijski dokažu, "ručno" ili pomoću nekih automatskih dokazivača teorema [3, 5].

Najnoviju verziju AGX softverskog paketa (AGX 3.3.9) kao i prateću dokumentaciju moguće je preuzeti sa Internet adrese

<https://www.gerad.ca/Gilles.Caporossi/agx/AGX/AutoGraphiX.html>.

Naslovni ekran aktiviranog softvera prikazan je na Sl. 1.





**Fig. 3.** Minimizacija spektralnog rastojanja u AGX softveru

Osim činjenice da je AGX softver opšte namene za rad sa grafovima, suočili smo se i sa problemom izbora okolina. Naime, kako je specificirano u [6, 18] i još nekim referencama, VNS u okviru AGX paketa koristi 10 različitih okolina ([6], Fig. 1). Pretraživanje kroz ovako obiman skup okolina u našem slučaju se pokazalo kao veoma zahtevno. U nekim relevantnim radovima, navedeno je da korisnik može da izabere podskup okolina, ali iz komunikacije sa autorom, prof. Caporossi-jem, saznali smo da je ta opcija izbačena iz novijih verzija AGX paketa. Da bismo olakšali posao AGX-u iskoristili smo činjenicu ([12], str. 85) po kojoj je broj grana u grafu jednak polovini zbira kvadrata sopstvenih vrednosti, tj.  $m = \frac{1}{2} \sum_{i=1}^n \lambda_i^2$ . Kada se ta činjenica iskoristi, optimizacioni zadatak za graf sa  $n = 10$  čvorova u AGX-u izgleda kao na Sl. 4.

Broj grana u grafu jednak je polovini zbira svih elemenata matrice susedstva  $A$ , tako da je bilo dovoljno izjednačiti zbir svih elemenata matrice  $A$  sa kvadratom spektra (tj. skalarnim proizvodom spektra sa samim sobom). Ovo je naš AGX program koji smo izvršavali za različite primere grafova.

Važno je napomenuti da je AGX stohastički pretraživač, zasnovan je na VNS metodi, pa stoga svako njegovo puštanje može dati različit rezultat, bilo u odnosu na samo rešenje (kada dobijamo kospektralni neizomorfnu graf), bilo u odnosu na vreme potrebno da se pronađe isti graf. Zbog toga je neophodno ponavljati izvršavanja i odrediti neke srednje (očekivane) rezultate. Iako AGX ne može da garantuje potpunu pretragu prostora rešenja (grafova sa zadatim brojem čvorova i grana), ponavljanje izvršavanja omogućava da se pronađu neki neizomorfnu kospektralni grafovi (ukoliko takvi postoje). Naravno, to što AGX nije uspeo

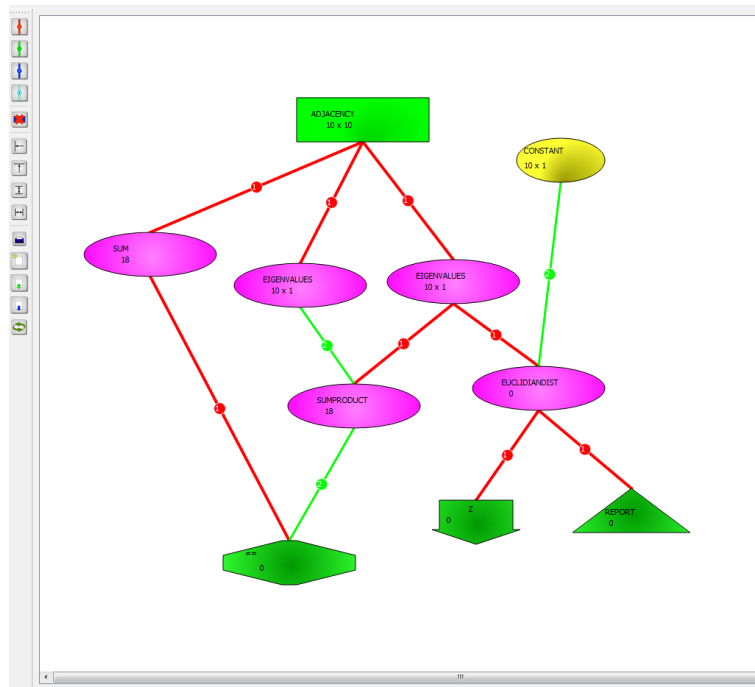


Fig. 4. Optimizovana minimizacija spektralnog rastojanja u AGX softveru

da pronad̄e kospektralni graf, tj. što je u svim puštanjima dobio isto rešenje, ne mora da znači da dati graf nema neizomorfnih kospektralnih grafova.

Kao što je već navedeno, da bi se pronašli svi kospektralni grafovi datom grafu potrebno je izvršiti potpunu pretragu grafova sa istim brojem čvorova  $n$  i grana  $m$  čiji broj je određen binomnim koeficijentom kojim se definiše na koliko načina se  $m$  grana može rasporediti na  $n(n-1)/2$  mesta. Na primer, za graf sa  $n = 8$  čvorova i  $m = 9$  grana potrebno je ispitati 6,906,900 grafova. Jasno je da se sa povećanjem broja čvorova problem potpune pretrage nelinearno usložnjava.

Međutim, postoje algoritmi koji se razvijaju za specijalne klase grafova i koriste se *a priori* znanje o tim grafovima da bi se smanjio broj grafova koji se analiziraju. Primer takvog algoritma opisan je u [13]. Autori su razmatrali Smitove grafove (kojima je spekatar ograničen na interval  $[-2,2]$ ) i za razmatrani graf tražili međusobno neizomorfne kospektralne grafove. Identifikovali su transformacije koje jedan Smitov graf prevode u drugi, kospektralan sa polaznim i razvili algoritam za generisanje svih takvih grafova. Prikazani su rezultati izvršavanja programa za dva odabrana primera.

## 4 Metoda promenljivih okolina za nalaženje grafa zadatog spektra

U ovom odeljku opisana je ukratko metoda promenljivih okolina, a zatim i konkretna implementacija metode za problem nalaženja grafa zadatog spektra.

### 4.1 Metoda promenljivih okolina

Metoda promenljivih okolina (Variable Neighborhood Search, VNS) je poznata metaheuristika zasnovana na lokalnom pretraživanju. Predložili su je Mladenović i Hansen [19], a zbog svoje jednostavnosti opšte je prihvaćena u međunarodnim istraživačkim krugovima i uspešno korišćena za rešavanje raznih optimizacionih problema [17].

Metoda koristi sistematsku promenu okolina, kako unutar lokalnog pretraživanja tako i prilikom izbegavanja zamke lokalnog optimuma. Zasnovana je na tri jednostavne činjenice [17]:

- (i) lokalni minimum u odnosu na jednu okolinu ne mora biti i lokalni minimum u odnosu na neku drugu okolinu;
- (ii) globalni minimum je lokalni minimum u odnosu na sve okoline;
- (iii) za većinu problema lokalni minimumi u odnosu na razne okoline su međusobno bliski.

Dakle, osnovni preduslov primene VNS metode je mogućnost uvođenja više okolina, bilo da se menja metrika u odnosu na koju se definiše okolina, bilo da se povećava rastojanje u odnosu na istu metriku. Osnovna metoda promenljivih okolina (Basic Variable Neighborhood Search, BVNS) podrazumeva da se koristi uvek ista metrika, a da se tokom pretrage prostora rešenja menja rastojanje u odnosu na tu metriku. Pseudokod BVNS metode predstavljen je algoritmom 1. BVNS se sastoji od tri koraka koji se ciklično izvršavaju do ispunjenja nekog unapred zadatog kriterijuma zaustavljanja. Ti koraci su [19]: razmrdavanje (shaking), lokalna pretraga (local search) i promena okolina (neighborhood change).

Korak razmrdavanja predstavlja diversifikaciju BVNS algoritma i generiše slučajno rešenje iz  $k$ -te okoline. Slučajnost se uvodi kako bi se obezbedilo pretraživanje različitih regiona prilikom sledećeg razmatranja okoline  $k$ . Rešenje generisano razmrdavanjem postaje početno rešenje u odnosu na koje se izvršava procedura lokalnog pretraživanja sa ciljem poboljšanja ne samo tog početnog rešenja nego i trenutno najboljeg rešenja  $x_{min}$ . Nakon završetka lokalne pretrage, proverava se kvalitet dobijenog lokalnog minimuma i shodno tome menja indeks okoline  $k$ .

BVNS metoda u toku svog izvršavanja imitira "najbrži spust", prilikom svakog poboljšavanja uvek se prelazi u to novo rešenje. Osnovni parametar BVNS metode je  $k_{max}$ —maksimalan broj okolina [19]. Kriterijum zaustavljanja može biti maksimalno dozvoljeno vreme izvršavanja, maksimalni broj iteracija, tj. koliko puta se dostigne  $k_{max}$ , broj iteracija između dve popravke globalno najboljeg rešenja, a mogu se koristiti kombinacije dva ili više kriterijuma.



**Algorithm 1** Pseudokod za BVNS metodu

---

```

procedure BVNS(Ulaz za problem,  $k_{max}$ , STOP)
   $x \leftarrow$  INITSOLUTION
   $x_{min} \leftarrow x$ 
   $f_{min} \leftarrow f(x)$ 
  repeat
     $k \leftarrow 1$ 
    repeat
       $x' \leftarrow$  RANDOMSOLUTION( $x_{min}, \mathcal{N}_k$ )           ▷ // Razmrđavanje
       $x'' \leftarrow$  LS( $x'$ )                               ▷ // Lokalno pretraživanje
      if ( $f(x'') < f_{min}$ ) then                         ▷ // Promena okoline
         $x_{min} \leftarrow x''$ 
         $f_{min} \leftarrow f(x'')$ 
         $k \leftarrow 1$ 
      else
         $k \leftarrow k + 1$ 
      end if
      ZAUSTAVLJANJE(STOP)
    until ( $k > k_{max} \vee STOP$ )
  until (STOP)
  RETURN( $(x_{min}, f_{min})$ )
end procedure

```

---

Postoje brojne modifikacije i proširenja originalno definisane VNS metode o kojima ovde neće biti reči. Za više detalja o VNS metodi i njenim varijantama zainteresovanog čitaoca upućujemo na [17].

## 4.2 Implementacija za problem spektralne rekonstrukcije grafova

Imajući na umu činjenicu da je poznat broj grana u grafu čiji spektar nam je zadat, implementirali smo osnovnu metodu promenljivih okolina (BVNS), jer je dovoljno koristiti samo jedan tip okoline (koji čuva broj grana u grafu): zamena jedne grane nekom drugom. Rezultujući grafovi ne moraju biti povezani jer taj uslov nije postavljen ni za polazne grafove (iako uglavnom svi analizirani primeri jesu povezani, ali mogu imati nepovezane kospektralne).

Rešenje razmatranog problema je graf koji obeležavamo sa  $g$  i koji treba da ima spektralno rastojanje od zadatog konstantnog vektora manje od neke unapred zadate konstante  $\varepsilon$ . Početno rešenje bira se slučajno među svim grafovima koji imaju zadati broj čvorova  $n$  i grana  $m = \frac{1}{2} \sum_{i=1}^n \lambda_i^2$ . Nakon toga vrše se transformacije polaznog grafa u skladu sa koracima BVNS metode. Rešenja koja se tokom rada BVNS metode dobijaju predstavljena su pomoću nekoliko struktura podataka kako bi se omogućila što efikasnija transformacija jednog grafa u neki njemu susedni ili u odgovarajući graf na datom rastojanju. Žrtvuemo memorijsko zauzeće na račun efikasnosti.

Kao prva struktura za reprezentaciju grafa korišćena je matrica susedstva  $A = [a_{ij}]_{n \times n}$ . Ona se koristi prilikom računanja spektra grafa. Zatim su tu liste

suseda  $g[i].ls$  i liste nesuseda  $g[i].ln$  za svaki čvor  $i$  grafa. Potrebno je u svakom trenutku znati i broj suseda/nesuseda čvora  $i$ , pa se to čuva u nizovima  $g[i].ns$  i  $g[i].nn$ . To znači da je graf reprezentovan na tri različita načina, ali to omogućava da se bilo koja transformacija grafa izvrši u konstantnom broju koraka  $O(1)$ .

Konkretno, ako treba da granu između čvora  $i$  i njegovog  $j$ -tog suseda pre- selimo tako da spaja čvor  $i_1$  sa njegovim susedom koji je na mestu  $j_1$  u listi nesuseda, potrebno je izvršiti sledeće naredbe (zapisane u C-ovskoj notaciji):

```

jp = g[i].ls[j]; // jp je j-ti sused čvora i
g[i].ls[j] = g[i].ls[-- ns[i]]; // briše se jp iz liste suseda čvora i
g[i].ln[nn[i]++] = jp; // dodaje se jp na listu nesuseda čvora i
js = g[i1].ln[j1]; // js je j1-vi nesused čvora i1
g[i1].ls[ns[i1]++] = js; // dodaje se js na listu suseda čvora i1
g[i1].ln[j1] = ln[-- nn[i1]]; // briše se js iz liste nesuseda čvora i1
A[i][jp] = A[jp][i] = 0; // ažuriranje elemenata matrice A
A[i1][js] = A[js][i1] = 1;

```

Čvor koji se briše iz liste zamenjuje se poslednjim čvorom u listi i odgovarajući broj elemenata u listi se smanjuje za jedan ( $--ns[i]$  i  $--nn[i_1]$ ). Novi čvor dodaje se na kraj liste pri čemu se i broj elemenata liste povećava za jedan ( $nn[i]++$  i  $ns[i_1]++$ ). Naravno, mora se još proveriti da neka od korišćenih listi nije prazna. Navedenih 8 operacija izvršava se nad slučajno izabranim parovima čvorova  $(i, j)$  i  $(i_1, j_1)$  u koraku razmrđavanja, dok se u lokalnoj pretrazi primenjuju na sve parove čvorova koji čine okolinu trenutnog rešenja.

### 4.3 Eksperimentalna analiza

U ovom odeljku prikazani su rezultati primene AGX softvera i predložene implementacije BVNS metode na rekonstrukciju nekih poznatih grafova sa manjim brojem čvorova.

BVNS metoda za spektralnu rekonstrukciju grafa implemetirana je u programskom jeziku R ([15]) koristeći razvojno okruženje RStudio Ver. 1.3.1073 for Windows, a izvršavana je na računaru sa sledećim karakteristikama Intel Core i7-9750H 2.66 GHz (12 MB Cache, up to 4.5 GHz), 16 GB DDR4, 512 GB M.2 SSD, NVIDIA GeForce GTX 1050, GDDR5 4 GB. AGX softver izvršavan je na istom računaru kako bi se obezbedilo fer poređenje.

Grafovi koji su služili kao primeri za poređenje predstavljani su na Sl. 5 i Sl. 6. To su grafovi sa 8, 9 i 10 čvorova koji su u [11] identifikovani kao pogodni modeli za arhitekture višeprosesorskih sistema.

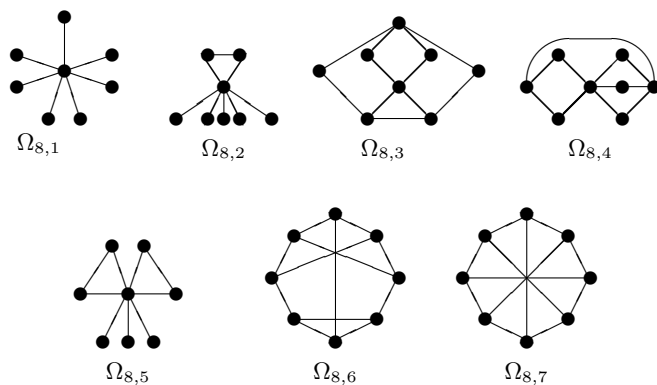


Fig. 5. Test grafovi sa 8 čvorova

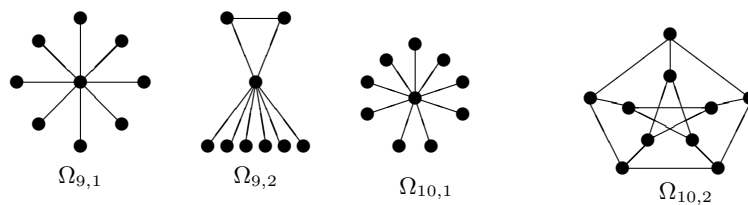


Fig. 6. Test grafovi sa 9 i 10 čvorova

#### 4.4 Primer određivanja neizomorfni kospektralnih grafova

Tokom eksperimentalne evaluacije utvrđeno je da graf  $\Omega_{8,5}$  nije jedinstveno određen svojim spektrom. Oba programa, AGX i BVNS, pronašli su još jedan neizomorfan graf prikazan na Sl. 7.

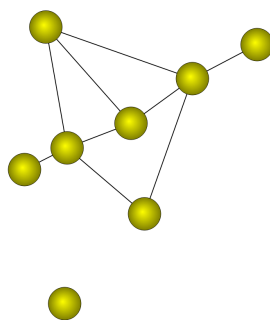


Fig. 7. Graf kospektralan sa  $\Omega_{8,5}$

Da bismo utvrdili da li su to jedina dva kospektralna grafa, izvršili smo detaljnu pretragu svih 6,906,900 grafova sa  $n = 8$  čvorova i  $m = 9$  grana. Među njima pronađeno je 10,900 grafova na spektralnom rastojanju 0 u odnosu na spektar grafa  $\Omega_{8,5}$ , ali su svi oni bili izomorfni ili sa  $\Omega_{8,5}$  ili sa grafom prikazanim na Sl. 7. Dakle, pokazano je da zdatom spektru odgovaraju samo dva pronađena grafa. Detaljna pretraga trajala je oko sat i po (1.4997 h).

Pokušali smo i drugi način detaljne pretrage: generisati prvo sve neizomorfne grafove, a onda među njima potražiti kospektralne. Ispostavilo se da je ovo duži put. Za 1.5 h pretraženo je 206,011 grafova (od 6,906,900) i među njima je pronađeno 402 neizomorfni, ali ni jedan na rastojanju 0 od zdatog vektora koji predstavlja spektar grafa  $\Omega_{8,5}$ . Za ovaj način pretrage potrebno je više vremena jer je ukupni broj neizomorfni grafova veći od broja kospektralni, pa proverava da li je novo-generisani graf izomorfan sa nekim koji već postoji traje duže nego kad se ispituje da li je takav graf kospektralan sa postojećima. Moguće je za generisanje neizomorfni grafova koristiti postojeće programske pakete (npr. *nauty* [1]) koji su maksimalno optimizovani za taj zadatak i tako skratiti potrebno vreme, ali to izlazi iz okvira ovog rada.

Da bismo uporedili koliko brzo AGX i SRG-BVNS mogu da pronađu oba kospektralna grafa, izvršavali smo oba programa 100 puta do trenutka kada se grafovi pronađu i merili potrebno vreme. Rezultati su sumirani u tabeli 1. Tabela je organizovana na sledeći način: u prvoj koloni opisani su podaci koji se nalaze u tabeli (korišćene metode, tip početnog rešenja i vreme rada metode). U koloni 2 dati su podaci za AGX metodu, ona uvek polazi od slučajnog rešenja (rnd vrednost u odgovarajućem polju). Naredne dve grupe od po dve kolone sadrže informacije vezane za SRG-BVNS. Tu smo probali dva načina za izbor novog početnog rešenja: slučajno i rešenje dobijeno razmrdavanjem trenutnog na rastojanju  $k_{max} = m$  (označeno sa  $Sh(k_{max})$  u tabeli 1). Dodatno, testirane su dve varijante SRG-BVNS metode, sa lokalnom pretragom do prvog poboljšanja (First Improvement Local Search) i sa lokalnom pretragom do najboljeg poboljšanja (Best Improvement Local Search). Dve varijante su označene sa (FI)SRG-BVNS i (BI)SRG-BVNS. Vreme rada dato je u sekundama i predstavlja prosečno vreme u 100 izvršavanja potrebno da svaka od metoda pronađe oba neizomorfna kospektralna grafa.

**Table 1.** Poređenje AGX-a sa (FI)SRG-BVNS i (BI)SRG-BVNS na primeru grafa  $\Omega_{8,5}$

Metoda	AGX	(BI)SRG-BVNS	(FI)SRG-BVNS	(BI)SRG-BVNS	(FI)SRG-BVNS
poč. rešenje	rnd	rnd	rnd	$Sh(k_{max})$	$Sh(k_{max})$
vreme (s)	1.116	1.834	0.504	2.154	0.632

Kao što se iz tabele 1 može videti, testirani programi su pronašli željene grafove za značajno kraće vreme nego detaljna pretraga. Naravno, broj neizomorfni kospektralni grafova nam je bio unapred poznat. U situacijama kada to nije slučaj, nije moguće garantovati da su pronađeni svi kospektralni grafovi. Tabela 1

pokazuje da je (FI)SRG-BVNS više nego duplo brži od AGX-a, dok je (BI)SRG-VNS za oko 64% sporiji. Zato smo odlučili da u nastavku eksperimentalne evaluacije radimo samo sa (FI)SRG-BVNS i izbacili prefiks (FI) radi kompaktnijeg zapisa. Na ovom primeru, bolje performanse pokazala je varijanta SRG-VNS koja svako izvršavanje počinje od potpuno slučajnog početnog rešenja.

Ovde je važno napomenuti da smo prilikom izvršavanja obe varijante SRG-BVNS metode kontrolisali vrednosti za inicijalizaciju generatora slučajnih brojeva (seed), tako da je dobijene rezultate moguće ponoviti. Kod AGX-a to nije moguće, tako da svaki novi eksperiment može generisati drugačije rezultate. Takođe, provera izomorfizma grafova ugrađena je u SRG-BVNS program (pozivom odgovarajuće funkcije R-a), dok se rezultati AGX-a moraju vizuelno proveravati na izomorfizam.

#### 4.5 Rezultati spektralne rekonstrukcije malih grafova

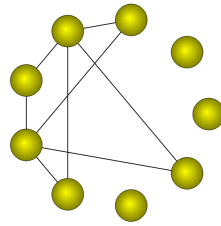
U ovom odeljku, predstavljeni su rezultati spektralne rekonstrukcije grafova prikazanih na Sl. 5 i Sl. 6. Kako se ne zna broj neizomorfni kospektralnih grafova u svim slučajevima, cilj je bio dobiti baš odgovarajući graf (do na izomorfnost). AGX i SRG-BVNS puštani su po 100 puta i mereno je vreme potrebno da se generiše željeni graf. U tabeli 2 predstavljeni su dobijeni rezultati, tj. prosečno vreme (u sekundama) koje su AGX i SRG-BVNS radili dok nisu generisali traženi graf.

**Table 2.** Poređenje vremena rada za AGX i SRG-BVNS

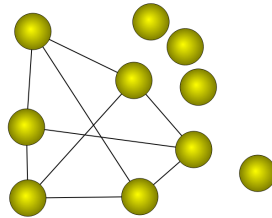
Graf	AGX	SRG-BVNS	Jedinstven
$\Omega_{8,1}$	26.67098	0.83801	–
$\Omega_{8,2}$	12.86802	0.93151	–
$\Omega_{8,3}$	0.18713	0.14256	–
$\Omega_{8,4}$	5.68055	0.37765	–
$\Omega_{8,5}$	1.25551	0.53264	NE
$\Omega_{8,6}$	0.3004	0.16999	–
$\Omega_{8,7}$	0.40173	0.16864	–
$\Omega_{9,1}$	10.88903	3.27195	NE
$\Omega_{9,2}$	N/A	3.33437	–
$\Omega_{10,1}$	N/A	4.51852	NE
$\Omega_{10,2}$	3.42404	0.63841	–

U poslednjoj koloni tabele 2 naveden je podatak koji govori o tome da li je za zadati graf pronađen neizomorfan kospektralni graf. Kolona nosi naziv "Jedinstven", mada je bez detaljne pretrage nemoguće utvrditi da li dati graf nema kospektralnih ili samo testirani programi nisu uspeali da ih pronađu. Zato je, u slučaju da datom grafu nije pronađen odgovarajući kospektralni, u poslednju kolonu stavljeno "–". Ukoliko je pronađen jedan ili više grafova neizomorfni

i kospektralnih datom, u poslednju kolonu je upisano "NE" kao odgovor na pitanje da li je razmatrani graf jedinstven. Kao što vidimo, identifikovan je po jedan kospektralni graf grafovima  $\Omega_{8,5}$  (prikazan na Sl. 7),  $\Omega_{9,1}$  (Sl. 8) i  $\Omega_{10,1}$  (Sl. 9).



**Fig. 8.** Graf kospektralni sa  $\Omega_{9,1}$



**Fig. 9.** Graf kospektralni sa  $\Omega_{10,1}$

Na osnovu prosečnih vremena prikazanih u tabeli 2 može se zaključiti da je implementirani SRG-BVNS višestruko brže pronalazio zadati graf od AGX-a. Posebno, u dva primera  $\Omega_{9,2}$  i  $\Omega_{10,1}$  AGX-u često treba i više od pola sata izvršavanja da bi uspeo da pronađe zadate grafove. Poređenje sa SRG-BVNS u tim slučajevima nema smisla i to je označeno sa N/A u odgovarajućim vrstama tabele 2. Dobijeni rezultati su bili očekivani imajući na umu da je AGX softver opšte namene i da koristi 10 tipova okolina od kojih je većina nepotrebna u ovom slučaju.

#### 4.6 Primena na Smitove grafove

U radu [13] razvijen je algoritam za nalaženje svih kospektralnih grafova nekog zadatog Smitovog grafa i prikazani su rezultati izvršavanja algoritma za dva izabrana primera. Algoritam je implementiran u okruženju .NET Framework v4.7.2, C#, XML, LinQ and Visual Studio 2019 pod Windows 10 operativnim sistemom. Izvršavan je na računaru AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx 2.10Gx, 8 GB RAM. Kako bismo obezbedili fer poređenje, passmark test (<https://www.passmark.com/products/performancetest/download.php>) izvršen je na oba računara i zaključeno je da su približno istih perofrmansi. Prvi testirani graf je  $T_5 + T_6 + 2P_1$  i prikazan je na slici 10. Za taj graf pronađeno je 25 neizomorfni kospektralnih grafova u radu [13], a potrebno vreme je bilo 25ms.

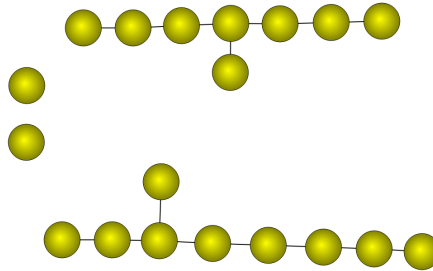
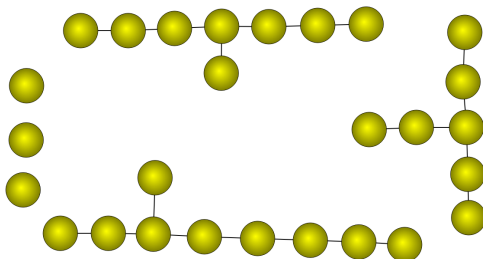


Fig. 10. Smitov graf  $T_5 + T_6 + 2P_1$

Za razliku od algoritma prikazanog u [13], SRG-BVNS je stohastička metoda pretraživanja, pa smo kao i ranije vršili 100 ponavljanja. Vreme rada ograničili smo na 30 min, pri čemu se nakon svakog pronađenog neizomorfni kospektralni graf vršilo njegovo razmrđavanje u okolini  $k_{max}$ . Naravno, provera kospektralnosti i izomorfizma su sastavni deo SRG-BVNS metode i zahtevaju dodatno vreme za svoje izvršavanje. Cilj nam je bio da procenimo koliko je SRG-BVNS uspešan u potrazi za neizomorfni kospektralni grafovima. Prosečan broj nađenih neizomorfni kospektralni grafova bio 19.22. Maksimalni broj pronađenih neizomorfni grafova je 21 (od 26 koliko ih ukupno ima).

Drugi testirani graf,  $T_4 + T_5 + T_6 + 3P_1$  prikazan na slici 11, ima 86 neizomorfni kospektralni grafova. Algoritmu za detaljnu pretragu iz [13] bilo je potrebno 590ms da ih sve pronađe. SRG-BVNS je u 100 puštanja našao prosečno 23.82 neizomorfni kospektralni grafova, a maksimalni pronađeni broj je 31. Svakako treba imati na umu da je SRG-BVNS heuristička metoda, dakle ne garantuje sistematičnost pretrage prostora kospektralni grafova. Takođe, razvijena je da radi za proizvoljne grafove, dok algoritam predložen u [13] koristi sva *a priori* znanja o Smitovim grafovima i neprimenljiv je u nekom drugom slučaju.



**Fig. 11.** Smitov graf  $T_4 + T_5 + T_6 + 3P_1$

Imajući na umu da u SRG-BVNS još uvek nisu iskorišćene sve informacije koje se o zadatom grafu mogu saznati na osnovu njegovog spektra, smatramo da postoji još prostora za buduća poboljšanja i da ova metoda ima velik potencijal za primene na problem spektralne rekonstrukcije grafa.

## 5 Zaključak

U ovom radu je razvijena osnovna metoda promenljivih okolina (Basic Variable Neighborhood Search, BVNS) za problem spektralne rekonstrukcije grafa, tj. za nalaženje grafa čiji spektar se poklapa sa nekim zadatim vektorom. Implementirana BVNS metoda uzima u obzir zavisnost broja grana u grafu od njegovog spektra. Rezultati primene BVNS metode na rekonstrukciju nekih poznatih grafova poređeni su sa rezultatima dobijenim primenom AutoGraphiX paketa. Oni nedvosmisleno pokazuju superiornost predložene BVNS implementacije u odnosu na brzinu izvršavanja. Kao potencijalne teme za budući rad nameću se eksperimenti sa grafovima većih dimenzija, poređenje sa sličnim metodama razvijenim za neke specijalne klase grafova i generisanje što većeg broja neizomorfnih kospektralnih grafova (ako takvi postoje). Takođe, planira se implementacija BVNS metode koja će koristiti mnoge druge poznate veze između parametara grafa i njegovog spektra kako bi se smanjio prostor pretrage i ubrzalo izvršavanje BVNS metode. Na primer, poznato je da je maksimalan stepen čvora u grafu veći ili jednak od najveće apsolutne vrednosti elemenata spektra; da zbir trećih stepena sopstvanih vrednosti podeljen sa 6 daje broj trouglova u grafu, a indeks (najveća sopstvena vrednost) može otkriti da li je graf regularan, povezan ili iz koliko komponenti povezanosti se sastoji. Druge matrice pridružene grafovima i drugi tipovi rastojanja mogu se koristiti prilikom implementacije BVNS-a, a i nekih drugih metaheurističkih metoda.

**Zahvalnica.** Rad je finansiralo Ministarstvo prosvete, nauke i tehnološkog razvoja preko Matematičkog instituta SANU i preko projekta F-159 Srpske Akademije



nauka i umetnosti. Autori se zahvaljuju akademiku Dragošu Cvetkoviću na brojnim komentarima i sugestijama koje su doprinele poboljšanju kvaliteta ovog rada.

## References

1. NAUTY – Graph Isomorphism, {<https://www3.cs.stonybrook.edu/~algorithm/Implement/nauty/Implement.shtml>}
2. Aouchiche, M., Bonnefoy, J.M., Fidahoussen, A., Caporossi, G., Hansen, P., Hiesse, L., Lacheré, J., Monhait, A.: Variable neighborhood search for extremal graphs 14: The autographix 2 system. In: *Global Optimization*, pp. 281–310. Springer (2006)
3. Aouchiche, M., Hansen, P.: A survey of automated conjectures in spectral graph theory. *Linear algebra and its applications* **432**(9), 2293–2322 (2010)
4. Brouwer, A.E., Spence, E.: Cospectral graphs on 12 vertices. *The electronic journal of combinatorics* **16**, N20:1–3 (2009)
5. Caporossi, G., Cvetković, D., Rowlinson, P.: Spectral reconstruction and isomorphism of graphs using variable neighbourhood search. *Bull. Acad. Serbe Sci. Arts, Cl. Sci. Math. Natur., Sci. Math.* **146**(39), 23–38 (2014)
6. Caporossi, G., Hansen, P.: Variable neighborhood search for extremal graphs: 1 the autographix system. *Discrete Mathematics* **212**(1-2), 29–44 (2000)
7. Chung, F.R.K., Graham, F.C.: *Spectral graph theory*. No. 92 in *CBMS Regional Conference Series in Mathematics*, American Mathematical Soc. (1997)
8. Comellas, F., Diaz-Lopez, J.: Spectral reconstruction of complex networks. *Physica A: Statistical Mechanics and its Applications* **387**(25), 6436–6442 (2008)
9. Cvetković, D.: *Teorija grafova i njene primene*. Naučna knjiga, Beograd, treće dopunjeno izdanje (1990)
10. Cvetković, D.: Spectral recognition of graphs. *Yugoslav Journal of operations research* **22**(2), 145–161 (2012)
11. Cvetković, D., Davidović, T.: Multiprocessor interconnection networks. In: *Zbornik radova, special issue Selected Topics on Applications of Graph Spectra*, pp. 35–62. second edition 14(22), Mathematical Institute SANU (2011)
12. Cvetković, D., Doob, M., Sachs, H.: *Spectra of graphs: Theory and application*. Johann Ambrosius Barth Verlag, Heidelberg–Leipzig, 3rd edition (1995)
13. Cvetković, D., Jerotijević, M.: Compositions of cospectrality graphs of smith graphs. *Kragujevac Journal of Mathematics* (accepted for publication) (2020)
14. Cvetković, D., Simić, S.: Graph spectra in computer science. *Linear Algebra and its Applications* **434**(6), 1545–1562 (2011)
15. Dessau, R.B., Phipps, C.B.: "R"-project for statistical computing. *Ugeskrift for laeger* **170**(5), 328–330 (2008)
16. Haemers, W.H., Spence, E.: Enumeration of cospectral graphs. *European Journal of Combinatorics* **25**(2), 199–211 (2004)
17. Hansen, P., Mladenović, N., Brimberg, J., Moreno Pérez, J.A.: Variable neighborhood search. In: *Handbook of metaheuristics*, pp. 57–97. Updated edition of a trailblazing volume, Springer (2019)
18. Jovanović, I.M.: *Spektralno prepoznavanje grafova i mreža*. Ph.D. thesis, Univerzitet u Beogradu, Matematički fakultet (2014)
19. Mladenović, N., Hansen, P.: Variable neighborhood search. *Computers & Operations Research* **24**(11), 1097–1100 (1997)

20. Spielman, D.: Spectral graph theory. In: Naumann, U., Schenk, O. (eds.) Combinatorial scientific computing, pp. 18:1–30. No. 18, CRC Press (2012)
21. Van Dam, E.R., Haemers, W.H.: Which graphs are determined by their spectrum? *Linear Algebra and its applications* **373**, 241–272 (2003)