LDPC dekoderi sa reinicijalizacijama koji objedinjuju tvrde odluke i razmenu mekih poruka

Predrag Ivaniš, *Senior Member, IEEE*, Srđan Brkić, *Member, IEEE* i Bane Vasić, *Fellow, IEEE*

***Apstrakt* — U ovom radu predloženi su postupci koji kombinuju dve klase iterativnih algoritama koje se uobičajeno koriste za dekodovanje zaštitnih kodova sa proverama parnosti male gustine (eng. Low Density Parity Check, LDPC). Prva strategija zasnovana je na bit-fliping algoritmu male kompleksnosti, a predložena modifikacija omogućava značajno poboljšanje performansi dekodera uz zadržavanje male prosečne računske kompleksnosti. Druga strategija zasnovana je na algoritmu sa propagacijom verodostojnosti i za cilj ima dodatno poboljšanje korekcionih sposobnosti dekodera, naročito kada se koriste kodovi sa malom dužinom kodne reči.**

***Ključne reči* — Bit-fliping algoritam, belief-propagation algoritam, iterativno dekodovanje, kodovi sa proverama parnosti male gustine.**

I. UVOD

KODOVI sa proverama parnosti male gustine (eng. Low Density Parity Check, LDPC) predloženi su od strane Roberta Galagera 1960. godine [1], a posebno veliku popularnost su doživeli u poslednjoj deceniji XX veka [2], [3]. Njihova sposobnost da dostignu kapacitet kanala uz relativno nisku kompleksnost dekodovanja [4] učinila ih je poželjnim rešenjem za obezbeđivanje pouzdanog prenosa podataka u bežičnim telekomunikacionim sistema novije generacije. Ovi kodovi se danas koriste u digitalnoj televiziji (eng. Digital Video Broadcasting, DVB) [5], bežičnim lokalnim mrežama (eng. Wireless Local Area Networks, WLAN) [6], a od skora su uvršteni i u standard za petu generaciju širokopojasnih mobilnih mreža (5G) [7].

Poboljšanje performansi sistema usled primene LDPC kodova pre svega je diktirano algoritmom koji se koristi za njihovo dekodovanje. Ovi algoritmi su po pravilu iterativni i implementiraju se nad Tanerovim grafom [8], koji je u potpunosti određen matricom provere parnosti koja definiše kod. U najvećem broju slučajeva, optimalne performanse se postižu ako se pri dekodovanju koristi tzv. algoritam propagacije verodostojnosti (eng. belief-propagation, BP), kod koga čvorovi u grafu razmenjuju poruke na osnovu kojih se procena kodne reči po pravilu poboljšava u svakoj narednoj iteraciji. Verzija BP algoritama pogodna za primenu u dekoderima LDPC kodova poznata je i pod nazivom algoritam sumiranja i množenja (eng. Sum Product Algorithm, SPA), a više detalja vezanih za njegovu efikasnu implementaciju može se naći u radu [9].

Predrag Ivaniš – Elektrotehnički fakultet, Univerzitet u Beogradu, Bulevar Kralja Aleksandra 73, 11020 Beograd, Srbija (e-mail: predrag.ivanis@ etf.bg.ac.rs).

Srđan Brkić – Elektrotehnički fakultet, Univerzitet u Beogradu, Bulevar Kralja Aleksandra 73, 11020 Beograd, Srbija (e-mail: srdjan.brkic@ etf.bg.ac.rs).

Bane Vasić – Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 USA (e-mail: vasic@ece.arizona.edu).

Mana BP algoritma je velika složenost implementacije. Računske operacije koje se izvode u čvorovima Tanerovog grafa mogu biti veoma kompleksne, a pritom se podrazumeva da su poruke koje se razmenjuju realni brojevi. Pri hardverskoj implementaciji poželjno je da se operacije u čvorovima pojednostave i da se poruke predstave sa konačnom preciznošću (implementacija sa fiksnim zarezom). Stoga se u većini postojećih sistema koriste podoptimalne verzije BP algoritma, kao što su razne vrste min-sum algoritma [10]. Drugi značajan nedostatak BP algoritma je vezan za činjenicu da je on optimalan samo ako Tanerov graf ima oblik stabla. Ako se u grafu pojavljuju ciklusi male dužine, performanse BP algoritma su daleko od performansi koje bi bilo moguće dobiti primenom algoritma koji postiže maksimalnu verodostojnost pri odlučivanju (eng. Maximum Likelihood, ML). Ovo je posebno izraženo za kratke kodove. U literaturi se mogu naći rešenja koja obezbeđuju bolje performanse od BP algoritma za isti broj iteracija [11], a koja sa povećanjem broja iteracija obezbeđuju performanse bliske ML granici [12].

Sa druge strane, algoritam koji u svakoj iteraciji invertuje određene bitove kodne reči (eng. bit-flipping, BF) predstavlja postupak izuzetno niske kompleksnosti. Iako originalni BF algoritam ima prilično loše performanse po pitanju ispravljanja grešaka, isto se ne može reći i za nedavno predloženu modifikaciju BF algoritma nazvanu bit fliping algoritam sa gradijentnim spustom (eng. Gradient Descent Bit Flipping, GDBF) [13]. U našem prethodnom radu [14] predložen je probabilistički GDBF algoritam (PGDBF), kod koga primena slučajne sekvence pri dekodovanju omogućava ispravljanje kombinacije grešaka koje klasičan GDBF algoritam ne može korigovati. Dalje unapređenje ove ideje je razmotreno u radu [15]. Kombinovanje PGDBF algoritma i slučajnih reinicijalizacija može u velikoj meri poboljšati performanse dekodera (odgovarajući algoritam MUDRI je izložen u radu [16]). Analiza data u radovima [17]-[20] pokazuje da dekoderi bazirani na tvrdom odlučivanju uz slučajne reinicijalizacije obezbeđuju performanse bliske ML granici, ukoliko se može dozvoliti veliki broj iteracija pri dekodovanju.

Nedavno je pokazano da se performanse GDBF algoritma mogu približiti performansama BP algoritma, ako se destimuliše invertovanje istog bita u nekoliko uzastopnih iteracija [21]. Kombinacija tog pristupa sa determinističkim reinicijalizacijama ima potencijal da omogući čak i bolje performanse od BP algoritma, uz znatno manju računsku kompleksnost [22].

U nastavku će biti razmotreno nekoliko strategija za kombinovanje BP i GDBF algoritma, sa ciljem da se postigne veća pouzdanost odlučivanja ili manja kompleksnost implementacije, koristeći prednosti jednog i drugog pristupa.

 ISBN 978-86-7466-930-3

II. MODEL SISTEMA I PREGLED TIPIČNIH ALGORITAMA ZA DEKODOVANJE LDPC KODOVA

U ovom radu posmatramo binarne LDPC kodove kod kojih je informaciona reč $\boldsymbol{i}$ dužine k, dok je odgovarajuća kodna reč $\boldsymbol{x}$ dužine n. Ovi kodovi se označavaju sa (n,k), a njihov kodni količnik iznosi $R=k/n$.

LDPC kod se opisuje matricom provere parnosti $\boldsymbol{H}$, dimenzija $n\times m$, koja je po pravilu retka (broj jedinica u ovoj matrici je znatno manji od broja nula). U ovom radu će biti razmotreni kodovi kod kojih u svakoj koloni matrice $\boldsymbol{H}$ ima tačno γ jedinica, dok svaka vrsta matrice $\boldsymbol{H}$ sadrži tačno ρ jedinica (regularni kodovi). Parametri γ i ρ nazivaju se težina kolona i težina vrsta, respektivno. Odgovarajući bipartitni graf $\mathcal{G}$ sastoji se od skupa varijabilnih čvorova $\boldsymbol{V} = \{v_1, v_2, ..., v_n\}$ i skupa čvorova provere parnosti $\boldsymbol{C} = \{c_1, c_2, ..., c_m\}$. Čvorovi v_i i c_j su susedi ako su povezani na grafu, tj. ako je ispunjeno $\boldsymbol{H}_{ij}=1$. Skup suseda čvora v_i je označen sa N_{v_i}, dok je skup suseda čvora c_j označen sa N_{c_j}.

Nakon što LDPC koder od informacione reči $\boldsymbol{i}$ formira kodnu reč $\boldsymbol{x}$, ona se prenosi kroz binarni simetrični kanal (BSC). Ovaj kanal unosi nasumične, tj. vremenski nekorelisane, greške sa verovatnoćom p. Odgovarajuća sekvenca greške $\boldsymbol{e}$ može se dobiti uz pomoć generatora slučajne promenljive sa Bernulijevom raspodelom $B(1, p)$. U ovom slučaju, primljena reč $\boldsymbol{y}$ na izlazu kanala formira se tako što se poslata kodna reč $\boldsymbol{x}$ sabere bit-po-bit sa sekvencom greške $\boldsymbol{e}$, tj. $\boldsymbol{y}=\boldsymbol{x}\oplus\boldsymbol{e}$.

Primljena reč se dovodi na ulaz dekodera koji ima zadatak da verno rekonstruiše poslatu kodnu reč, odnosno poslatu informacionu sekvencu (koju treba dostaviti odredištu). Pošto je izdvajanje informacione sekvence prilično jednostavno ako se ispravno rekonstruiše poslata kodna reč, pri analizi dekodera se teži da procena kodne reči bude što je moguće pouzdanija. Drugim rečima, treba minimizovati verovatnoću da se procena poslate kodne reči, označena sa $\hat{\boldsymbol{x}}$, razlikuje od poslate reči $\boldsymbol{x}$. Ova verovatnoća se obično naziva verovatnoća greške po kodnoj reči (eng. Word Error Rate, WER). Iterativni dekoder procenu poslate kodne reči po pravilu poboljšava iz iteracije u iteraciju, čime se omogućava da i za prilično duge kodne reči kompleksnost dekodera ostane relativno mala [10].

Dekoder u svakoj iteraciji proverava da li su sve provere parnosti zadovoljene. Kada je ovaj uslov ispunjen, proces dekodovanja se prekida (tada procena odgovara kodnoj reči). Obično se zadaje maksimalan broj iteracija koje se mogu iskoristiti za dekodovanje, označen sa L.

Neka je procena kodne reči u l-toj iteraciji označena sa $\hat{\boldsymbol{x}}^{(l)}$, $l=1,2,...,L$. Odgovarajući sindrom $\boldsymbol{S}^{(l)}=\hat{\boldsymbol{x}}^{(l)}\boldsymbol{H}^T$ ima sve komponente ravne nuli samo u slučaju da su sve provere parnosti zadovoljene. Ako dekoder tokom L uzastopnih iteracija ne formira procenu za koju je $\mathbf{S}^{(l)}=\mathbf{0}$, dekodovanje se proglašava neuspešnim. Ukoliko se dekodovanje prekine za $l_0\le L$ i ako je $\hat{\boldsymbol{x}}^{(l_0)}=\boldsymbol{x}$, smatra se da je ta kodna reč uspešno dekodovana nakon l_0 iteracija.

U današnjim telekomunikacionim sistemima i sistemima za zapis podataka obično se za dekodovanje LDPC kodova koriste dve klase algoritama. Jedna je zasnovana na BP algoritmu, koga odlikuju visoka pouzdanost pri dekodovanju i velika složenost implementacije. Druga klasa algoritama je zasnovana na BF postupku, koji je daleko jednostavniji ali uz nešto veću verovatnoću greške pri odlučivanju. Osnovne karakteristike dve pomenute klase algoritama, kao i njihovih varijanti, navedene su u nastavku.

A. *Algoritmi sa propagacijom verodostojnosti*

Poznato je da se LDPC kodovi mogu uspešno dekodovati korišćenjem algoritama kod kojih se poruke iterativno razmenjuju između povezanih čvorova u grafu (eng. message-passing). Poruke koje se razmenjuju mogu biti veoma jednostavne, kao kod Galager-A/B algoritma (binarne poruke), ali se znatno bolji rezultati dobijaju ako ove poruke imaju oblik realnih brojeva.

Osnovna ideja BP algoritma će biti ilustrovana na grafu prikazanom na slici 1, gde su ispravno primljeni biti kodne reči označeni belim varijabilnim čvorovima, dok pogrešno primljenim bitima odgovaraju crni čvorovi. Zadovoljene provere parnosti označene su belim, a nezadovoljene crnim kvadratima. U BP algoritmu poruke koje varijabilni čvorovi inicijalno šalju ka povezanim proverama parnosti zavise samo od verovatnoće greške u binarnom kanalu

$$m^{(1)}_{v_i\to c_j}(v_i=0\,|\,y_i)=P(v_i=0\,|\,y_i)=\begin{cases}p, & y_i=1,\\ 1-p, & y_i=0,\end{cases} \qquad (1)$$

a pošto su u pitanju verovatnoće, jasno je da se može pisati $m^{(1)}_{v_i\to c_j}(v_i=1\,|\,y_i)=1-m^{(1)}_{v_i\to c_j}(v_i=0\,|\,y_i)$. Intenzitet ovih poruka srazmeran je uverenosti čvora v_i da bi mogao da ima vrednost $v_i=a$, kada mu je poznata primljena vrednost y_i.

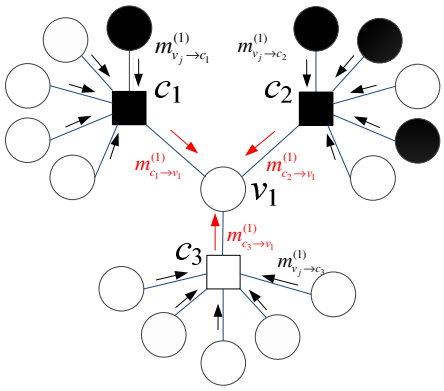
Imajući u vidu da vrednosti varijabilnih čvorova treba da odgovaraju bitima validne kodne reči (za koju su sve provere parnosti ravne nuli), u narednim iteracijama su poruke koje se razmenjuju između čvorova na Tanerovom grafu određene izrazima [10]

$$m^{(l)}_{c_j\to v_i}(v_i=0)=\frac{1}{2}\left(1-\prod_{v_k\in N_j\setminus v_i}\left(2m^{(l)}_{v_k\to c_j}(v_k=1)-1\right)\right), \qquad (2)$$

$$m^{(l+1)}_{v_i\to c_j}(v_i=0)=P(v_i=0\,|\,y_i)\prod_{c_z\in N_i\setminus c_j}m^{(l)}_{c_z\to v_i}(v_i=0), \qquad (3)$$

a jasno je da i dalje važi $m^{(l)}_{c_j\to v_i}(v_i=1)=1-m^{(l)}_{c_j\to v_i}(v_i=0)$, $m^{(l+1)}_{v_i\to c_j}(v_i=1\,|\,y_i)=1-m^{(l+1)}_{v_i\to c_j}(v_i=0\,|\,y_i)$. Verovatnoća da je v_i=0 se sada ažurira na osnovu uverenja povezanih čvorova

$$P^{(l)}(v_i=0)=P(v_i=0\,|\,y_i)\prod_{c_z\in N_i}m^{(l)}_{c_z\to v_i}(v_i=0). \qquad (4)$$



Sl 1. Ilustracija BP algoritma na grafu koji odgovara regularnom kodu sa parametrima γ=3, ρ=5.

 ISBN 978-86-7466-930-3

Na ovaj način se uverenost čvora o vrednostima koje na njega povezani čvorovi treba da imaju, pod datim uslovima, po pravilu poboljšava iz iteracije u iteraciju, što obično rezultuje sve pouzdanijim odlukama. Da bi se ublažili numerički problemi, BP algoritam se obično implementira u logaritamskom domenu, tako što se u svakoj iteraciji određuje logaritamski količnik verodostojnosti [8]

$$LLR_i^{(l)} = \log\left(P^{(l)}(v_i=1)/P^{(l)}(v_i=0)\right). \quad (5)$$

Ipak, uvek treba imati na umu da je BP algoritam optimalan samo u slučaju da je graf oblika stabla. Ovo nije ispunjeno kada u grafu postoje ciklusi male dužine, a posebno je kritično kada je dužina kodne reči mala.

B. *Bit-fliping algoritmi*

Bit-fliping algoritam je jednostavan postupak iterativnog dekodovanja LDPC kodova, koji je predložio Galager [1]. Algoritam je zasnovan na određivanju broja nezadovoljenih provera parnosti koje su povezane na svaki varijabilni čvor. Ovo izračunavanje se izvodi nad trenutno dostupnom procenom kodne reči, pa tako čvoru v_i u l-toj iteraciji odgovara energetska funkcija oblika

$$\Lambda_{BF}^{(l)}(v_i) = \sum_{c_j \in N_{v_i}} \bigoplus_{v_k \in N_{c_j}} \hat{x}_k^{(l)}. \quad (4)$$

Ukoliko je ova veličina za i-ti bit (i=1,2,…,n) veća od unapred definisanog praga T, vrši se invertovanje tog bita, čime se formira njegova procena u narednoj iteraciji

$$\hat{x}_i^{(l+1)} = \begin{cases} \hat{x}_i^{(l)} \oplus 1, & \Lambda^{(l)}(v_i) \geq T, \\ \hat{x}_i^{(l)}, & \Lambda^{(l)}(v_i) < T. \end{cases} \quad (5)$$

Polazeći od primljenog vektora ($\hat{\boldsymbol{x}}^{(0)} = \boldsymbol{y}$), na ovaj način se u svakoj iteraciji po pravilu formiraju sve pouzdanije procene poslate kodne reči. Ovaj algoritam često greši, pa bi za tipičnu vrednost $T = \gamma/2$ u primeru sa slike 1 bila doneta pogrešna odluka da čvor v_1 treba invertovati.

GDBF algoritam uvodi dve modifikacije – u energetsku funkciju se dodaje korelacioni član koji zavisi od odgovarajućeg bita primljene reči i invertuju se samo vrednosti onih varijabilnih čvorova koje u toj iteraciji imaju maksimalnu vrednost energetske funkcije [13, 14]:

$$\Lambda_{GDBF}^{(l)}(v_i) = y_i \oplus \hat{x}_i^{(l)} + \sum_{c_j \in N_{v_i}} \bigoplus_{v_k \in N_{c_j}} \hat{x}_k^{(l)}, \quad (6)$$

$$\hat{x}_i^{(l+1)} = \begin{cases} \hat{x}_i^{(l)} \oplus 1, & \Lambda^{(l)}(v_i) = \max_i \Lambda^{(l)}(v_i), \\ \hat{x}_i^{(l)}, & \Lambda^{(l)}(v_i) < \max_i \Lambda^{(l)}(v_i). \end{cases} \quad (7)$$

Kod GDBF algoritma sa momentumom (GDBF-w/m) energetska funkcija ima oblik [21]

$$\Lambda^{(l)}(v_i) = \alpha\left(y_i \oplus \hat{x}_i^{(l)}\right) + \beta \sum_{c_j \in N_{v_i}} \bigoplus_{v_k \in N_{c_j}} \hat{x}_k^{(l)} + \mu_i^{(l)}, \quad (8)$$

gde se korelacionom članu i broju nezadovoljenih provera parnosti pridružuju težinski koeficijenti (označeni sa α i β, respektivno), a na čitavu funkciju se dodaje novi član $\mu_i^{(l)}$ (momentum). Vrednost momentuma zavisi od broja iteracija koje su protekle od prethodnog invertovanja tog bita (koji je označen sa w_i) i može uzeti vrednosti definisane momentum vektorom $\mu = [\mu(1), \mu(1), ..., \mu(w_{\max})]$, pri čemu je $\mu(w_i) = 0$ ako je $w_i > w_{\max}$. Pokazano je da ovaj algoritam za neke kodove ima performanse uporedive sa BP algoritmom, naročito za male vrednosti parametra p [21].

III. STRATEGIJE ZA KOMBINOVANJE BF I GDBF ALGORITMA

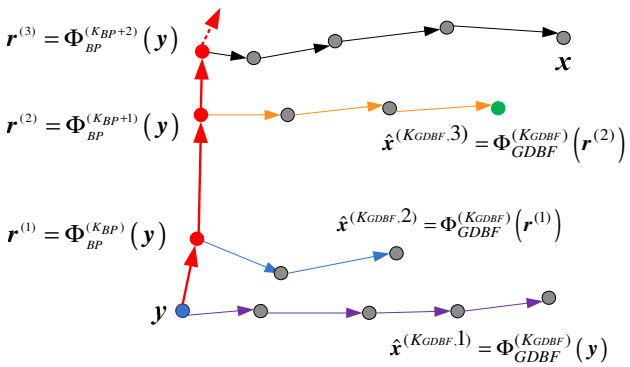
Jasno je da dve prethodno opisane klase algoritama rade na osnovu različitih principa, pa se nameće pitanje da li bi se mogle kombinovati tako da se objedini "najbolje iz dva sveta". Prvi korak u tom smeru učinjen je u radu koji smo nedavno objavili [22], pri čemu se GDBF-w/m algoritam modifikuje samo u nekim iteracijama.

Za razliku od prethodnih istaraživanja, u ovom radu je predlažemo kombinovanje BP algoritma i GDBF algoritma, pri čemu se jedan od njih koristi kao osnovno rešenje, dok se elementi drugog algoritma koriste za reinicijalizaciju ulaza dekodera. Dve takve strategije opisane su u nastavku.

A. *GDBF-w/m kao osnovni algoritam*

U ovoj varijanti se primljena reč dovodi na ulaz GDBF-w/m dekodera i za datu vrednost parametra p posmatra se kako se menja WER u zavisnosti od L. Formalno se može zapisati da GDBF-w/m dekoder nakon l iteracija preslikava $\boldsymbol{y}$ u procenjenu vrednost $\hat{\boldsymbol{x}}^{(l)} = \Phi_{GDBF}^{(l)}(\boldsymbol{y})$. Kada postane jasno da GDBF-w/m u dodatnim iteracijama neće značajno smanjiti WER, dekodovanje se prekida. Broj iteracija nakon koga se dekodovanje prekida je K_{GDBF} i određuje se empirijski. Primljena reč se zatim dovodi na ulaz BP dekodera, koji nakon K_{BP} iteracija na svom izlazu formira procenu $\boldsymbol{r}^{(1)}$, tj. $r^{(1)} = \Phi_{BP}^{(l)}(\boldsymbol{y})$, koja se koristi kao ulaz GDBF-w/m dekodera u novoj rundi. Po potrebi, postupak se ponavlja na način ilustrovan slikom 2, tako što se referenca $\boldsymbol{r}^{(q)}$ dobija kao izlaz BP dekodera nakon K_{BP}+q-1 iteracija, a energetska funkcija se računa na osnovu izraza

$$\Lambda^{(l)}(v_i) = \alpha\left(r_i^{(q)} \oplus \hat{x}_i^{(l)}\right) + \beta \sum_{c_j \in N_{v_i}} \bigoplus_{v_k \in N_{c_j}} \hat{x}_k^{(l)} + \mu_i^{(l)}.$$

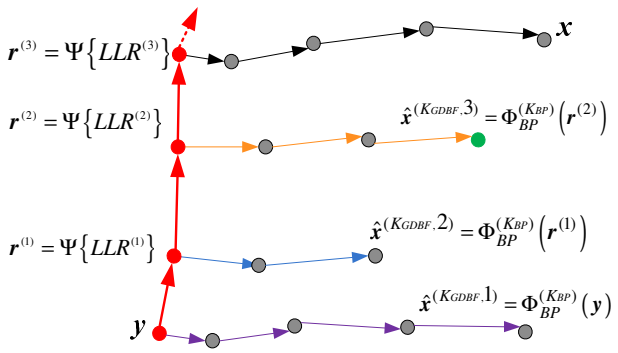


Sl 2. Ilustracija prve strategije, osnova je GDBF w/m algoritam, dok procena nakon odgovarajuće iteracije BP algoritma određuje referencu.

B. *BP kao osnovni algoritam*

U ovom scenariju se primljena reč dovodi na ulaz BP dekodera. Nakon svake iteracije pronađu se čvorovi v_i kod kojih je $LLR_i^{(l)}$ najveći po apsolutnoj vrednosti, uz uslov da bi taj bit bio invertovan kada bi se odluka donela u toj iteraciji, tj. $LLR_i^{(l)} \times LLR_i^{(0)} < 0$. Ove pozicije se redom zapisuju u odgovarajuću memoriju i čine skup kritičnih čvorova. Ako BP algoritam tokom K_{BP} iteracija ne obavi dekodovanje, postupak se ponavlja za novi ulaz dekodera. Odgovarajuća referenca r_q se formira tako što se u reči $\boldsymbol{y}$ invertuje samo jedan od kritičnih čvorova, pa se može formalno pisati $\boldsymbol{r}^{(q)} = \Psi\{LLR^{(q)}\}$ (videti sliku 3).

 ISBN 978-86-7466-930-3



Sl 3. Ilustracija prve strategije, osnova je GDBF w/m algoritam, dok procena nakon odgovarajuće iteracije BP algoritma određuje referencu.

IV. NUMERIČKI REZULTATI

Efekat dve predložene strategije će biti ilustrovan za dva regularna koda, koji su konstruisani koristeći različite postupke, pri čemu oba imaju težinu kolona γ=3 i dužinu najkraćeg ciklusa jednaku g=8. Kodovi imaju male dužine kodnih reči, a poznato je da tada BP algoritam predstavlja podoptimalno rešenje.

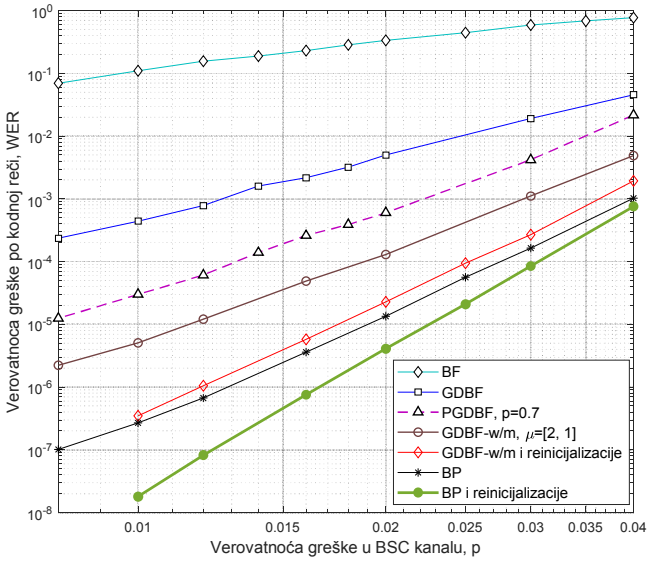
Verovatnoća greške po kodnoj reči određena je Monte Karlo simulacijom, pri čemu se generišu kodne reči dužine n i na svaku od njih se superponira slučajno generisan vektor greške iste dužine, generisan u skladu sa zadatom verovatnoćom greške u BSC. Kako bi bili sigurni da je simetrija dekodera zadovoljena, ne šalje se uvek kodna reč "sve nule", već se na izlaz kodera emituju različite kodne reči. U slučaju kada se dekodovanje završi za najviše L iteracija, porede se poslata i procenjena kodna reč, da bi bili sigurni da procena ne odgovara kodnoj reči koja je različita od poslate kodne reči. U slučaju kada se tokom L uzastopnih iteracija ne formira procena kodne reči kod koje su sve provere parnosti jednake nuli, dekodovanje se proglašava neuspešnim. Simulacija za jedan skup parametara se zaustavlja kada se detektuje 200 pogrešno primljenih kodnih reči, što određuje jednu tačku na grafiku.

Performanse dekodera će prvo biti prikazane za regularni Tanerov kod (155,64), koji ima težinu kolona γ=3 i težinu vrsta ρ=5. U pitanju je kvazi-ciklični LDPC kod, a odgovarajući metod konstrukcije predložen je u radu [23]. Ovaj kod se često koristi za testiranje algoritama dekodovanja [11, 24, 25].

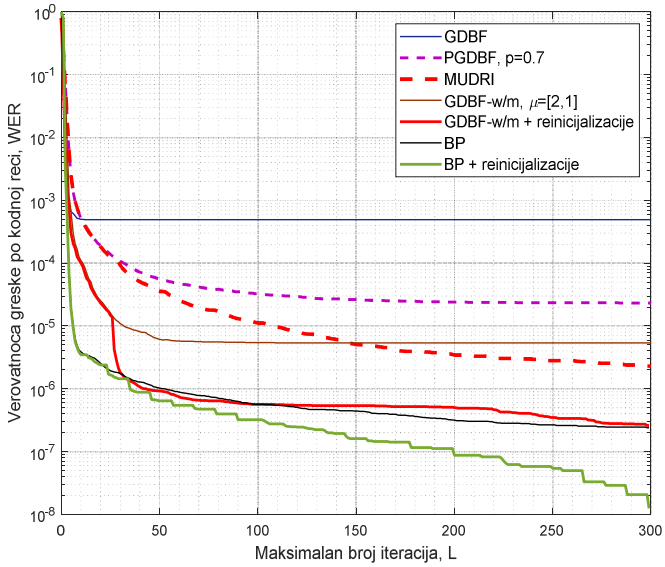
Zavisnost verovatnoće greške po kodnoj reči od verovatnoće greške u BSC prikazana je na slici 4. U slučaju kada BP algoritam određuje samo reinicijalizovane reference od kojih GDBF-w/m počinje dekodovanje u svakoj rundi, kriva je prikazana crvenom punom linijom. Svaka runda traje po K_{GDBF}=25 iteracija i ako se za to vreme ne detektuje da su sve provere parnosti jednake nuli, referenca se reinicijalizuje (sa K_{BP}=5) i ponovo se pokreće GDBF-w/m algoritam za nešto izmenjenu ulaznu sekvencu. Zelena kriva odgovara slučaju kada se koristi samo BP algoritam, ali se u svakoj rundi od K_{BP}=10 iteracija kao ulaz dekodera koristi reinicijalizovana referenca. Za male vrednosti parametra p, prva strategija obezbeđuje performanse uporedive sa BP algoritmom dok drugoj strategiji odgovaraju superiorne performanse.

Verovatnoća greške po kodnoj reči u zavisnosti od maksimalnog dozvoljenog broja iteracija pri dekodovanju ilustrovana je na slici 5. Vidi se da već nakon prve reinicijalizacije (L>30) performanse modifikovanog GDBF-w/m dekodera postaju uporedive sa performansama BP dekodera. Sa druge strane, modifikovani BP dekoder za bilo koju vrednost parametra L ima bolje performanse od ostalih razmatranih rešenja.

Dok strategija br 1. obezbeđuje verovatnoću greške po kodnoj reči koja je praktično ista kao u slučaju primene znatno složenijeg BP algoritma, jasno je da strategija br. 2 obezbeđuje dodatno poboljšanje performansi u odnosu na BP algoritam. Ovo je rezultat reinicijalizacija kod kojih se invertuju vrednosti najkritičnijih varijabilnih čvorova, što je princip sličan onom koji se primenjuje u GDBF algoritmu.



Sl. 4. Zavisnost verovatnoće greške po kodnoj reči od verovatnoće greške u BSC, za Tanerov kod (115,64) i L=300. Prikazane su performanse za GDBF-w/m i BP, kao i dve strategije njihovog kombinovanja.



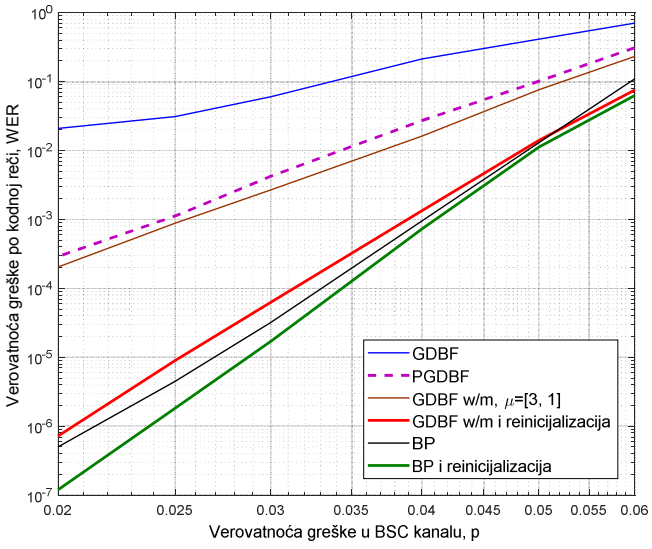
Sl. 5. Zavisnost verovatnoće greške po kodnoj reči od broja iteracija, za Tanerov kod (115,64) i p=0.01. Prikazane su performanse za GDBF-w/m i BP, kao i dve strategije njihovog kombinovanja.

 ISBN 978-86-7466-930-3

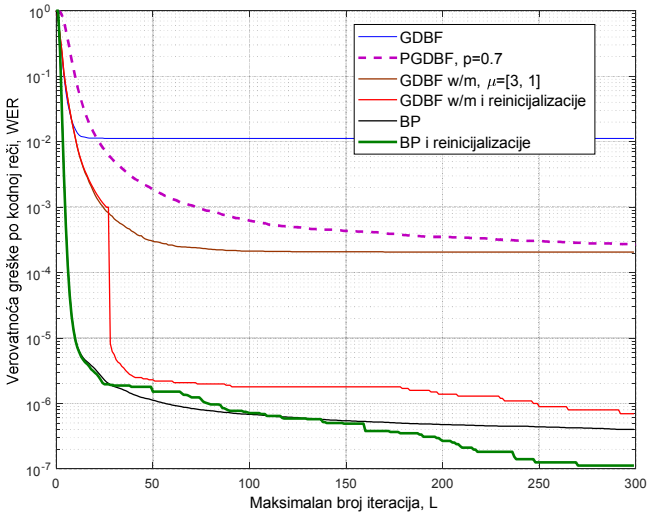
Numerički rezultati za regularni PEG kod (504,252), koji ima težinu kolona γ=3 i težinu vrsta ρ=6. U pitanju je kod formiran pomoću tehnike sa progresivnim formiranjem ivica grafa (eng. Progressive Edge Growth, PEG), koja je prvobitno predložena u radu [26].

Zavisnost verovatnoće greške po kodnoj reči od verovatnoće greške u BSC prikazana je na slici 6. Crvena linija odgovara slučaju kada se BP algoritam koristi samo za formiranje reinicijalizovane reference (sa K_{BP}=10), dok osnovu dekodera predstavlja GDBF-w/m algoritam (opisano u odeljku III-A). Kao i u prethodnom slučaju, svaka runda traje po K_{GDBF}=25 iteracija. Zelena puna linija odgovara slučaju kada se u svakoj rundi koristi samo BP algoritam, a kao ulaz dekodera koristi se reinicijalizovana referenca (na način opisan u odeljku III-B). U ovom slučaju, prva runda se završava nakon K_{BP}=25 iteracija, dok se nakon prve reinicijalizacije ova vrednost smanjuje na K_{BP}=10. U oba slučaja postižu se performanse uporedive sa BP algoritmom, dok druga strategija rezultuje osetnijim poboljšanjem performansi u tzv. *error floor* regionu.

Uticaj parametra L na performanse dekodera za PEG kod (504,252), kada je p=0.02, prikazan je na slici 7. Performanse dekodera u kome je primenjena strategija opisana u odeljku III-B su za malu vrednost parametra L uporedive sa performansama BP dekodera. Neznatna degradacija performansi u intervalu $25<L<75$ može da posluži kao indikacija da bi se još bolji rezultati mogli dobiti ako se u prvoj rundi dekodovanja parametar K_{BP} dodatno poveća. Sa druge strane, može se uočiti da se sa povećanjem maksimalnog broja iteracija performanse značajno poboljšavaju u odnosu na BP algoritam. Ovo je posledica reinicijalizacija u svakoj rundi dekodovanja. Taj efekat smo prethodno uočili u radu [16], za slučaj kada se koriste slučajne reinicijalizacije. U radu [17] pokazano je da se sa dovoljnim povećanjem parametra L performanse PGDBF dekodera asimptotski približavaju ML granici. U radovima [19] i [20] pokazano je da se sličan efekat postiže ako se slučajne reinicijalizacije kombinuju sa najjednostavnijim algoritmom koji koristi massage-passing princip za dekodovanje LDPC kodova (Gallager-B algoritam).



Sl. 6. Zavisnost verovatnoće greške po kodnoj reči od verovatnoće greške u BSC, za Tanerov kod (115,64) i L=300. Prikazane su performanse za GDBF-w/m i BP, kao i dve strategije njihovog kombinovanja.



Sl. 7. Zavisnost verovatnoće greške po kodnoj reči od broja iteracija, za PEG kod (504,252) i p=0.02. Prikazane su performanse za GDBF-w/m i BP, kao i dve strategije njihovog kombinovanja.

Rezultati prikazani na slikama 4-7 pokazuju da se značajno poboljšanje performansi može postići i kada su reinicijalizacije deterministčke. Za razliku od pristupa izloženog u radu [22], gde se opisan prilično složen postupak deterministčke reinicijalizacije, ovde je predložen koncept zasnovan na kombinaciji dva algoritma:

- U prvoj strategiji, reč od koje GDBF-w/m dekoder započinje dekodovanje u svakoj rundi dobija se na osnovu procena u pojedinim iteracijama BP algoritma.
- U drugoj strategiji se metoda bliska GDBF algoritmu (flipovanje najkritičnijeg bita) koristi za reinicijalizaciju, dok se BP koristi za dekodovanje u svakoj rundi. U prethodnim primerima pretpostavljeno je da se u procesu reinicijalizacije invertuju samo oni čvorovi v_i za koje je zadovoljeno $LLR_i^{(l)} = \max_i \left\{ LLR_i^{(l)} \right\}$. Ipak, preliminarni rezultati pokazuju da se za kodove sa dužim kodnim rečima bolje performanse postižu ako se ovaj skup proširi čvorovima za koje je ispunjena relacija $LLR_i^{(l)} \geq 0.9 \times \max_i \left\{ LLR_i^{(l)} \right\}$.

Treba zapaziti da se u prvoj strategiji dekodovanje najčešće završi primenom GDBF algoritma. Ukoliko je $p=10^{-2}$, prosečan broj iteracija potrebnih za dekodovanje reči Tanerovog koda (155,64) pomoću GDBF-w/m algoritma sa reinicijalizacijama iznosi $\bar{l} = 1.85$, dok je za $p=3\times10^{-2}$, prosečan broj iteracija $\bar{l} = 3.59$. Ako je $p=10^{-2}$, verovatnoća da GDBF-w/m tokom prvih K_{GDBF}=25 iteracija ne završi dekodovanje iznosi $\mathrm{WER}(25) \approx 1.2\times10^{-5}$ (videti sliku 5), iz čega se može zaključiti da se BP algoritam veoma retko pokreće, naročito u error-floor regionu, gde performanse predloženih rešenja posebno dolaze do izražaja.

Direktna posledica je da postupak predložen u prvoj strategiji obezbeđuje znatno manju prosečnu računsku kompleksnost u odnosu na slučaj kada se koristi BP dekoder. Ovaj postupak je baziran na GDBF-w/m algoritmu, a u radu [27] i doktorskoj disertaciji [28] navedeno je da u slučaju koda dužine n=1296 implementacija GDBF algoritma obezbeđuje šest puta veći prosečan protok i približno devet puta manju površinu na čipu u odnosu na najefikasniju poznatu implementaciju min-sum algoritma (koji predstavlja podoptimalnu varijantu BP algoritma, pogodnu za implementaciju zasnovanu na fiksnom zarezu).

 ISBN 978-86-7466-930-3

V. ZAKLJUČAK

Dve predložene strategije kombinovanja BP i GDBF-w/m algoritama pogodne su u slučaju kada su u prijemniku dostupna oba tipa dekodera. Dobijeni rezultati pokazuju da se kombinovanjem dva uobičajena pristupa mogu dobiti značajno poboljšane performanse ili performanse uporedive onima koje ima BP dekoder, ali uz smanjenu kompleksnost.

U ovom radu su parametri dekodera određeni empirijski. Sigurni smo da se dodatno poboljšanje performansi može dobiti ako se izvrši optimizacija pojedinih parametara, kao što su momentum vektor ili pragovi za odlučivanje kod GDBF-w/m algoritma, kao i broj iteracija koji odgovara pojedinim rundama dekodovanja. Ovo će biti tema naših budućih istraživanja.

ZAHVALNICA

Rezultati objavljeni u ovom radu delom su dobijeni u okviru saradnje ostvarene kroz i program Saradnje srpske nauke sa dijasporom Fonda za nauku Republike Srbije (br. ugovora 6462951), kao i kroz ERASMUS+ KA2 program saradnje Univerziteta u Beogradu i Univerziteta u Arizoni. Angažovanje Predraga Ivaniša i Srđana Brkića podržano je od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije. Angažovanje Baneta Vasića podržano je od strane NSF u okviru projekata CIF-1855879, CCF-2106189, CCSS-2027844 i CCSS-2052751 i NASA-SURP.

LITERATURA

[1] R. G. Gallager, *Low Density Parity Check Codes*, MIT Press, Cambridge, Mass., 1963.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[3] D. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[4] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, 47(2):619–637, 2001.

[5] ETSI Digital Video Broadcasting (DVB). *Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and other Broadband Satellite Applications; Part 2: DVB-S2 Extensions (DVB-S2X)*, ETSI EN 302307-2 V.1.1.1 (2014-10); ETSI: Sophia Antipolis, France, 2014.

[6] IEEE Standard for Information Technology—*Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE Standard 802.11-2016; IEEE: New York, NY, USA, 2016.

[7] 3rd Generation Partnership Project. *Technical Specification Group Radio Access Network; NR; Multiplexing and Channel Coding (Release 16)*, 3GPP TS 38.212 V16.5.0 (2021-03); 3GPP: Valbonne, France, 2021.

[8] L. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, Vol. 27, no. 5, pp. 533-547, Sep. 1981.

[9] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM 2001),* San Antonio, USA, November 25-29 2001, vol.2, pp. 1036-1036E.

[10] D. Declercq, M. Fossorier, and E. Biglieri, *Channel Coding: Theory, Algorithms, And Applications*. Academic Press Library in Mobile and Wireless Communications, Elsevier, 2014.

[11] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasić, "Finite alphabet iterative decoders, Part I: Decoding beyond belief propagation on the binary symmetric channel," *IEEE Trans. Commun*, vol. 61, no. 10, pp. 4033–4045, Nov. 2013.

[12] D. Declercq, E. Li, B. Vasić, and S. Planjery, "Approaching maximum likelihood decoding of finite length LDPC codes via FAID diversity," in *Proc. IEEE Information Theory Workshop*, Lausanne, Switzerland, Sep. 3–7 2012, pp. 487–491.

[13] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami and I. Takumi, "Gradient descent bit flipping algorithms for decoding LDPC codes," *IEEE Trans. Commun*, vol. 58, no. 6, pp. 1610-1614, June 2010.

[14] O.-A. Rasheed, P. Ivanis, and B. Vasić, "Fault-tolerant probabilistic gradient-descent bit flipping decoders," *IEEE Commun. Letters*, vol. 18, no. 9, pp. 1487 – 1490, Sep. 2014.

[15] H. Cui, J. Lin, Z. Wang, "An Improved Gradient Descent Bit-Flipping Decoder for LDPC Codes," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 66, no. 8, pp. 3188-3200, Aug. 2019.

[16] P. Ivanis, O.-A. Rasheed, and B. Vasić, "MUDRI: A fault-tolerant decoding algorithm," in *Proc. IEEE International Conference on Communications (ICC 2015)*, London, UK, June 8–12 2015, pp. 4291– 4296.

[17] B. Vasić, P. Ivaniš, D. Declercq, and K. LeTrung, "Approaching maximum likelihood performance of LDPC codes by stochastic resonance in noisy iterative decoders," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2016, pp. 1–9.

[18] D. Declercq, C. Winstead, B. Vasic, F. Ghaffari, P. Ivanis, and E. Boutillon, "Noise-Aided Gradient Descent Bit-Flipping Decoders approaching Maximum Likelihood Decoding", in *Proc 9th International Symposium on Turbo Codes & Iterative Information Processing (ISTC 2016)*, Special Session: Noisy Error Correction, Brest, France, 5-9 September 2016, pp. 300-304.

[19] P. Ivaniš, B. Vasić, D. Declercq, "Performance Evaluation of Faulty Iterative Decoders using Absorbing Markov Chains", *in Proc. IEEE International Symposium on Information Theory (ISIT 2016)*, Barcelona, Spain, July 10-15 2016, pp. 1566-1570.

[20] P. Ivaniš and B. Vasić, "Error Errore Eicitur: A Stochastic Resonance Paradigm for Reliable Storage of Information on Unreliable Media," *IEEE Trans. Commun*, vol. 64, no. 9, pp. 3596-3608, Sep. 2016.

[21] V. Savin, "Gradient Descent Bit-Flipping Decoding with Momentum," in *Proc. International Symposium on Topics in Coding (ISTC 2021)*, Montreal, Canada, 30 August - 3 September 2021.

[22] P. Ivaniš, S. Brkić, B. Vasić, "Suspicion Distillation Gradient Descent Bit-Flipping Algorithm," *Entropy*, vol. 24, no. 4, Article No. 558, Apr. 2021.

[23] R. M. Tanner, D. Sridhara, T. Fuja, "A class of group-structured LDPC codes," In *Proc. ISTA*, Ambleside, UK, 2001.

[24] S. Zhang and C. Schlegel, "Controlling the Error Floor in LDPC Decoding," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3566-3575, Sep. 2013.

[25] R. Asvadi, A. H. Banihashemi and M. Ahmadian-Attari, "Lowering the Error Floor of LDPC Codes Using Cyclic Liftings," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213-2224, Apr. 2011.

[26] X.-Y. Hu, E. Eleftheriou and D.-M. Arnold, "Progressive edge-growth Tanner graphs,". in *Proc IEEE Global Telecommunications Conference (GLOBECOM'01)*, November 2001, vol.2, pp. 995-1001.

[27] K. Le, F. Ghafari, D. Declercq and B. Vasić, "Efficient Hardware Implementation of Probabilistic Gradient Descent Bit-Flipping," *IEEE Trans. Circuits and Systems I: Regular* Papers, vol. 64, no. 4, pp. 906-917, Apr. 2016.

[28] L. T. Khoa, *New Direction on Low Complexity Implementation of Probabilistic Gradient Descent Bit-Flipping Decoder*, PhD Thesis, École Nationale Supérieure de l'Électronique et de ses Applications, Université de Cergy Pontoise, Cergy Pontoise, France, 2017.

ABSTRACT

In this paper we propose the approaches that combine two types of iterative decoding algorithms that are usually used for decoding of low density parity check codes (LDPC). One strategy is based on a low-complexity bit-flipping algorithm, and the proposed modification enable significant performance improvement, with no significant increase of the average computing complexity. The other strategy is based on belief propagation decoder, and the resulting decoder has improved error correction capabilities for the codes with short codeword length.

LDPC decoders with re-initializations based on synergy of hard decision and message passing principles

Predrag Ivaniš, Srđan Brkić, Bane Vasić

 ISBN 978-86-7466-930-3