Poređenje performansi unidirekcionog i bidirekcionog stek algoritma

Mladen Kovačević

***Sadržaj* — U ovom radu dati su osnovni principi dekodovanja kodova sa strukturom stabla odnosno njihove podklase - konvolucionih kodova, sa naglaskom na sekvencijalnom dekodovanju. Navedene su neke osobine osnovnog (unidirekcionog) i bidirekcionog stek algoritma za sekvencijalno dekodovanje. Na kraju su pokazani i rezultati njihovog poređenja putem računarskih simulacija.**

***Ključne reči* — bidirekcioni stek algoritam, sekvencijalno dekodovanje, stek algoritam.**

I. UVOD

TEORIJA zaštitnog kodovanja počela je da se razvija 1948. godine nakon Šenonovog rada "Matematička teorija komunikacija" i njegove fundamentalne teoreme o kanalnom kodovanju. Šenon je definisao za svaki kanal veličinu koju je nazvao kapacitet kanala i pokazao da je moguće slati informacije brzinama manjim od kapaciteta sa proizvoljno malom verovatnoćom greške. Ovaj rezultat označio je početak potrage za dobrim kodovima i efikasnim postupcima dekodovanja.

Paralelno se razvijalo više pravaca u rešavanju ovog problema - algebarski, probabilistički, geometrijski. U prvim decenijama veoma intenzivno se razvijala algebarska teorija zaštitnog kodovanja, još od prvog Hemingovog koda pa sve do Rid - Solomonovih (RS) kodova. Sa druge strane, paralelno je napredovala probabilistička teorija (trelis kodovi). Ona je naročito postala aktuelna devedesetih godina prošlog veka, sa pojavom i razvojem turbo kodova zajedno sa MAP iterativnim dekodovanjem.

Trelis kodovi imaju široko područje primene kao što su svemirske i satelitske komunikacije, TCM (Trellis Coded Modulation), mobilni sistemi novije generacije itd. dok se algoritmi za njihovo dekodovanje primenjuju i na razne druge oblasti (dekodovanje blok kodova, kompresija podataka, prepoznavanje oblika).

Postupci za dekodovanje trelis kodova su zbog svega ovoga vrlo važan deo teorije zaštitnog kodovanja. Sa pronalaskom optimalnog algoritma (u smislu najmanje verovatnoće greške) kakav je Viterbijev nije završen razvoj ove oblasti jer je potrebno dekoder optimizovati i po drugim parametrima kakav je njegova kompleksnost (prostorna ili vremenska). Ovaj pristup doveo je do pronalaska velikog broja suboptimalnih postupaka dekodovanja sa znatno smanjenom kompleksnošću i neznatno povećanom verovatnoćom greške.

M. Kovačević, Fakultet Tehničkih Nauka, Trg D. Obradovića 6, 21000 Novi Sad, Srbija; (telefon: 381-64-3740221; e-mail: mladenkovacevic@eunet.yu).

II. DISKRETNI KANALI BEZ MEMORIJE. METRIKA

U daljem tekstu biće pretpostavljeno da je prenos informacija vršen preko diskretnog kanala bez memorije (DMC, Discrete Memoryless Channel). Takođe se smatra da nema povratne sprege u kanalu, tj. da nema uticaja prijemnika na predajnik. Za takav kanal važi:

$$\mathrm{P}[y_n \mid x_0,...,x_{n-1},x_n,y_0,...,y_{n-1}] = \mathrm{P}[y_n \mid x_n], \quad (1)$$

$$\mathrm{P}[y_1,...,\ y_n \mid x_1,...,\ x_n] = \prod\nolimits_{n=1}^{N} \mathrm{P}[y_n \mid x_n]. \quad (2)$$

pa je on potpuno opisan skupom verovatnoća prelaza $\{p[y_j \mid x_l],\ j=1,...,r,\ l=1,...,s\}$ gde su x_l simboli ulaznog a y_j izlaznog kanalnog alfabeta, sa *s* odnosno *r* elemenata, redom.

Zadatak dekodera koji minimizuje verovatnoću greške dekodovanja kodne reči jeste da nađe sekvencu koja maksimizuje združenu verovatnoću ulazne i izlazne kanalne sekvence

$$\mathrm{P}[\bar{y}_{L+M},\bar{x}_{L+M}] = \mathrm{P}[\bar{y}_{L+M} \mid \bar{x}_{L+M}] \cdot \mathrm{P}[\bar{x}_{L+M}]. \quad (3)$$

Pošto entropijsko kodovanje i šifrovanje koji dolaze pre zaštitnog kodovanja obično izjednače verovatnoće svih ulaznih kanalnih sekvenci $\mathrm{P}[\bar{x}_{L+M}]$, dovoljno je maksimizovati verovatnoće $\mathrm{P}[\bar{y}_{L+M} \mid \bar{x}_{L+M}]$ ili bilo koju monotono rastuću fukciju tih verovatnoća. Obično se uzima:

$$\begin{aligned} d(\bar{y}_{L+M} \mid \bar{x}_{L+M}) &= A \cdot \left(\log_2 \mathrm{P}[\bar{y}_{L+M} \mid \bar{x}_{L+M}] - f(\bar{y}_{L+M})\right) \\ &= A \cdot \sum\nolimits_{l=0}^{L+M} \left(\log_2 \mathrm{P}[\bar{y}_{L+M}^{(l)} \mid \bar{x}_{L+M}^{(l)}] - f(\bar{y}_{L+M}^{(l)})\right) \end{aligned} \quad (4)$$

U prethodnim jednačinama $\bar{x}_{L+M}$ označava poslatu sekvencu dužine *L+M*, a $\bar{y}_{L+M}$ odgovarajuću primljenu sekvencu. Pojedinačne grane (grana je definisana u sledećem odeljku) ovih sekvenci označene su superskriptima (l), gde je *l* redni broj grane. Konstanta *A* je proizvoljan pozitivan broj.

Tako dobijen izraz naziva se metrika, a dekoder koji svoju odluku bazira na njenoj maksimizaciji naziva se dekoder po maksimalnoj verodostojnosti. Ovaj tip metrike je, međutim, pogodan samo za poređenje sekvenci iste dužine, dok neki algoritmi kao strategiju koriste poređenje sekvenci različite dužine. Metrika koja je izvedena da optimizuje ovakve algoritme naziva se Fanoova metrika i definisana je sa:

$$d_F(\bar{y}_l \mid \bar{x}_l) = A \cdot \sum_{n=1}^{l \cdot N} \left(\log_2 \frac{\mathrm{P}[y_n \mid x_n]}{\mathrm{P}[y_n]} - R \right) \quad (5)$$

gde je R kodna brzina koda, a N broj simbola na grani.

III. KODOVI SA STRUKTUROM STABLA

Kod sa strukturom stabla definisan je kao preslikavanje niza K-dimenzionih simbola q-arnog alfabeta u kodnu reč koja se može predstaviti kao niz grana u stablu. Preslikavanje je takvo da od mogućih q^K grana koje izlaze iz svakog čvora stabla, simbol izvornog alfabeta bira neku prema unapred definisanom pravilu. Tako se dolazi do sledećeg čvora u stablu i tako redom. Svakoj grani pridružen je niz od N ulaznih kanalnih simbola (iz r-arnog alfabeta). Na taj način izvorna sekvenca definiše put u stablu, koji sa druge strane definiše niz kanalnih simbola, odnosno kodnu reč. Kodna brzina ovakvog koda je $R = \frac{K}{N} \cdot \log_2 q$.

A. Trelis kodovi

Kada se ograniči "memorija" koda sa strukturom stabla, tj. kada na izbor određene grane u stablu ne utiču svi do tada pridošli simboli izvora već samo prethodnih M i trenutni, dobija se trelis kod. Trelis koder može se zamisliti kao konačan automat čija se struktura može predstaviti pomoću grafa, trelis dijagrama ili naravno stabla.

B. Konvolucioni kodovi

Konvolucioni kodovi su posebna podklasa trelis kodova kod kojih je preslikavanje sekvence ulaznih simbola u kodnu reč linearno. Najčešće se u praksi koriste konvolucioni koderi koji su vremenski nepromenljivi (FCE - Fixed Convolutional Encoder) i za koje je $q = r$. Takođe se pretpostavlja da simboli ulaznog i izlaznog alfabeta kodera pripadaju nekom konačnom polju. Ovakav koder definisan je sa $K \cdot N$ generatorskih sekvenci odnosno polinoma $g^{(k,n)}$ stepena ne većeg od M. Ako je $\bar{i}_l$ l-ta grana ulazne sekvence u ovakav koder (koja se sastoji od K q-arnih simbola), a $\bar{x}_l$ odgovarajuća grana izlazne sekvence iz kodera (koja se sastoji od N r-arnih simbola; ovde je $r = q$), onda je:

$$x_l^n = \sum_{k=1}^{K} \sum_{m=0}^{M} i_{l-m}^k \cdot g_m^{(k,n)}, \; n = 1,\ldots, N \quad (6)$$

gde se prepoznaje operacija konvolucije. Sa superskriptima n i k su u granama sekvenci označeni redosledi simbola u grani.

Funkcionisanje fiksnog konvolucionog kodera najjedno- stavnije se predočava uz pomoć trelis dijagrama koji predstavlja dijagram stanja sa uračunatom i vremenskom komponentom. Na ovom dijagramu čvor predstavlja jedno od $q^{K \cdot M}$ stanja kodera. U svaki čvor ulazi i iz njega izlazi q^K grana; svakoj grani je pridružen izlaz kodera (niz od N simbola) koji odgovara prelazu između njenih krajnjih čvorova, tj. stanja.

Skoro sve današnje komunikacije odvijaju se prenosom informacija podeljenih u frejmove određene dužine. Tako i konvolucioni koderi mogu da "pakuju" informacije ako se ulazna sekvenca izdeli na delove dužine L grana ($L \cdot K$ q-arnih simbola), i svakoj od njih se doda neka fiksna i proizvoljna sekvenca dužine M grana kako bi se koder naterao da dođe u određeno (i uvek isto) završno stanje. Ovakva sekvenca naziva se rep i obično se uzima da je ispunjena svim nulama. Rep smanjuje kodnu brzinu za faktor $L/(L+M)$, ali ovo smanjenje nije asimptotski značajno.

IV. DEKODOVANJE KODOVA SA STRUKTUROM STABLA

Zadatak dekodera je, kao što je već rečeno, da pronađe putanju koja je najverovatnije bila poslata odnosno putanju sa najvećom krajnjom metrikom. Međutim, efikasnost dekodera je potrebno posmatrati i kroz njegovu kompleksnost, a ne samo verovatnoću greške.

Računska kompleksnost algoritma predstavlja broj aritmetičkih operacija po dekodovanom simbolu ili grani. Obično se, međutim, posmatra broj čvorova koji su "produženi" jer sve takve operacije nad čvorovima zahtevaju približno isti broj elementarnih operacija. Jedna operacija nad čvorom u trelisu ili stablu predstavlja određivanje stanja i metrika svih njegovih naslednika. Dakle, broj svih obrađenih čvorova je mera računske (vremenske) kompleksnosti. Osim vremenske posmatra se i prostorna komleksnost algoritma koja predstavlja količinu resursa koje dati algoritam koristi (memorija, procesori, itd.). Kao univerzalna mera kompleknosti algoritma može se uzeti proizvod vremenske i prostorne kompleksnosti.

Optimalan algoritam, u smislu minimalne verovatnoće greške, jeste Viterbijev koji ispituje sve putanje u trelis dijagramu dobivši na kraju putanju sa najvećom metrikom. Mana Viterbijevog algoritma je, međutim, eksponencijalna zavisnost broja stanja koja treba ispitati, pa samim tim i računske (vremenske) kompleksnosti algoritma, od memorije kodera M što ga čini nepodobnim za korišćenje u slučaju kodova sa velikom memorijom. Za kodove sa velikom memorijom potrebno je koristiti suboptimalne postupke dekodovanja koji se jednim imenom nazivaju - sekvencijalno dekodovanje.

Ideja sekvencijalnog dekodera je u tome da se ne ispituju sve moguće putanje kroz trelis, već da se posmatraju samo one najverovatnije odnosno one sa najvećom metrikom. Sekvencijalni dekoder će u svakom koraku produžiti neke (ili samo jednu) od putanja koje je već ranije ispitao, i to one koje imaju najveće metrike. Na taj način, smanjuje se kompleksnost algoritma dok verovatnoća greške nije značajno povećana (i dalje eksponencijalno opada sa M). Najpoznatiji i verovatno najjednostavniji algoritam iz ove klase jeste stek algoritam.

V. STEK ALGORITAM ZA SEKVENCIJALNO DEKODOVANJE

Stek algoritam prvi je predložio Zigangirov [4] a kasnije nezavisno od njega i Jelinek [4], pa se još naziva i

ZJ algoritam. Osnovna ideja je da se u svakom koraku produži samo najbolji čvor, odnosno najbolja do tada ispitana putanja. Očigledno je onda da će biti potrebno porediti putanje različitih dužina pa se zbog toga koristi Fanoova metrika.

Kao što mu i samo ime kaže ovaj algoritam sadrži stek u kome su sve ispitane putanje do određenog trenutka sortirane po metrici. U svakom koraku se najbolja od njih zamenjuje njenim naslednicima (kojih ima q^K) sa odgovarajućim metrikama. Koraci ovog algoritma su sledeći (ovi koraci mogu se razlikovati pošto postoji više varijacija ovog algoritma):

1. Inicijalizacija steka ubacivanjem početnog čvora (korena stabla) sa metrikom jednakom nuli.
2. Najbolja putanja iz steka se produžava za jednu granu (izračunavaju se stanja i metrike svih njenih naslednika), briše se, a njeni naslednici sortiraju i ubacuju u stek tako da on ostane sortiran po metrici.
3. Ako je najbolja putanja na dubini $L+M$, njoj odgovarajuća informaciona sekvenca se prosleđuje na izlaz i algoritam je završen. U suprotnom prelazi se na korak 2.

Stek algoritam zasniva se na *principu neodabiranja* [2] koji glasi: Ako neke dve putanje u stablu, definisane informacionim sekvencama $\bar{i}^1_{L+M}$ i $\bar{i}^2_{L+M}$, divergiraju na dubini j i važi:

$$\min_{j<l<L+M} d\left(\bar{x}^1_l, \bar{y}_l\right) > \min_{j<l<L+M} d\left(\bar{x}^2_l, \bar{y}_l\right) \qquad (7)$$

tada $\bar{i}^2_{L+M}$ ne može biti putanja na vrhu steka kada se algoritam završi. Drugim rečima, stek algoritam bira putanju sa najvećim minimumom metrike; od svih putanja u stablu ona koja ima najveći minimum biće njegova konačna odluka. Ovo očigledno ne mora biti i putanja sa najvećom akumulisanom krajnjom metrikom pa sledi, kao što je već rečeno, da ovaj algoritam nije optimalan u smislu verovatnoće greške. Dakle, izlaz stek algoritma neće se, u opštem slučaju, poklapati sa izlazom Viterbijevog dekodera ali ova razlika nije toliko velika da ne bi bila opravdana znatno manjom kompleksnošću sekvencijalnog dekodera kakav je stek algoritam.

Kompleksnost stek algoritma skoro je nezavisna od memorije koda M. Ona je, međutim, slučajna promenljiva pa je i vreme dekodovanja varijabilno što je i jedan od nedostataka ovog algoritma. Može se pokazati [4] da je raspodela ove slučajne promenljive tzv. Paretoova:

$$\mathrm{P}[C \geq \eta] < A \cdot \eta^{-\rho} \qquad (8)$$

gde je A pozitivna konstanta, $\rho \geq 0$ Paretoov eksponent a C predstavlja broj produženih čvorova po dekodovanoj grani. Slučajna promenljiva C dakle opisuje računsku kompleksnost algoritma. To je i najbolje moguće ponašanje raspodele promenljive C, odnosno i donja granica verovatnoće sa leve strane nejednakosti (8) data je Paretoovom raspodelom [4]. Kompleksnost takođe zavisi i od kodne brzine. Naime, pokazuje se da je potrebno ograničiti kodnu brzinu koda na vrednosti manje od granične brzine R_0 (*cutoff rate*) [4], $R_0 \leq R_C$, gde je R_C kapacitet kanala. Za brzine veće od ove značajno raste verovatnoća da broj operacija dekodera prevaziđe neku vrednost η.

VI. Bidirekcioni stek algoritam

Skoro svi osnovni (unidirekcioni) algoritmi imaju i svoje bidirekcione analogone. Kod njih je neophodno da podaci budu izdeljeni u frejmove i da svaka sekvenca sadrži rep, odnosno da ima poznato završno stanje. Može se pokazati da se takve sekvence mogu dekodovati i od kraja pri čemu se tada koristi inverzni kod.

Inverzni kod dobija se inverzijom generatorskih sekvenci, odnosno generatorske sekvence inverznog koda su "slika u ogledalu" generatorskih sekvenci originalnog koda. Svi bidirekcioni algoritmi koriste dve pretrage stabla, sa obe strane. Pritom se pri pretraživanju stabla unapred koristi originalni kod, dok pretraga unazad koristi inverzni kod.

Bidirekcioni stek algoritam predložili su nezavisno Šenk i Radivojac [3] [7] [8] i Kalel i Li [6]. Ovaj algoritam koristi dva steka, F (Forward) i B (Backward). Još se uvode pojmovi tunela, TD (Tentative Decision) registra i kriterijuma za "orezivanje" stabla. Tunel je jedinstvena sekvenca dugačka T grana ($0 \leq T \leq M$) koja povezuje dva stanja u trelisu. TD registar je sekvenca dugačka $L+M$ grana koja povezuje poznato početno i krajnje stanje trelisa i pritom ima najveću akumulisanu metriku od svih putanja te dužine koje su ispitane do datog trenutka. Kriterijumi za orezivanje stabla su pravila prema kojima se određuje da je neka putanja "loša", tj. da najverovatnije neće biti deo dekodovane putanje, pa se u tom slučaju može odmah odstraniti iz odgovarajućeg steka. Koraci algoritma su:

1. Inicijalizacija stekova F i B sa početnim stanjem (korenom stabla) odnosno poznatim krajnjim stanjem, redom, oba sa nultom metrikom. Jedan od stekova se aktivira (npr. stek F).
2. Orezivanje stabla na dubini na kojoj se nalazi najbolji put u aktivnom steku prema nekom od kriterijuma orezivanja. Ako je stek ovime ispražnjen, algoritam je završen i sadržaj TD registra se prosleđuje na izlaz. U suprotnom: ako je ovime odstranjen i najbolji put aktivira se drugi stek, a ako nije, najbolji put iz aktivnog steka se produžava za jednu granu i izbacuje iz steka. Njegovi naslednici (na dubini l) se zatim povezuju tunelom ako je moguće, tj. ako su stanja kompatibilna, sa putevima u drugom steku na dubini $L+M-l-T$. Najbolja od ovih putanja i putanje u TD registru se upisuje u TD registar.
3. Putevi koji su tunelovali se ubacuju u stek vodeći računa o njihovim metrikama (odnosno o sortiranju). Promena aktivnog steka i povratak na korak 2.

Postoje razni kriterijumi za orezivanje stabla. Jedan od njih može biti:

- Ukoliko je metrika posmatranog puta na dubini l manja od minimuma metrike putanje iz TD registra, tada se posmatrani put i svi putevi na dubini l aktivnog steka sa metrikom manjom od posmatranog puta, mogu iz njega odstraniti jer je malo verovatno da budu deo

dekodovane putanje.

Za binarne kanale za koje ima smisla pojam Hemingovog rastojanja može se definisati i kriterijum:

- Ukoliko je zbir Hemingovog rastojanja datog puta na dubini l i najmanjeg Hemingovog rastojanja svih ispitanih puteva iz drugog steka na dubini $L+M-l-T$ ili manjoj, veći od Hemingovog rastojanja TD registra i primljene sekvence, tada se posmatrani put izbacuje jer je malo verovatno da bude deo dekodovane putanje.

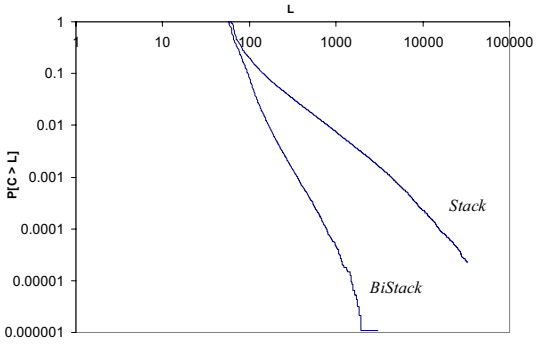
VII. Poređenje

Lako se pokazuje [4] da metrika ispravne putanje u stablu u proseku raste, dok metrike neispravnih putanja imaju tendenciju da opadaju. Stek algoritam će zbog toga, ukoliko nema grešaka u primljenoj sekvenci, samo pratiti ispravnu putanju i stablo se neće razgranati. Vreme dekodovanja je u tom slučaju vrlo malo. Ako se, međutim, pojavi paket grešaka, metrika ispravne putanje će početi da opada i stek algoritam će morati da istraži više putanja u stablu da bi pronašao pravu. Broj istraženih čvorova tada eksponencijalno raste sa dužinom paketa grešaka i na tom "mestu" se dekoder najviše zadržava. Ovo širenje stabla na mestu grešaka je dakle osnovni uzrok varijabilnosti vremena dekodovanja i njegovog povećanja u odnosu na slučaj kada grešaka nema.

Dekoder koji bi dekodovao primljenu sekvencu od repa bi se takođe lako probio do mesta gde se nalaze greške i tu bi se najviše zadržao zbog razgranavanja stabla. Iz ovakvog rezonovanja intuitivno sledi da je bidirekcioni algoritam znatno efikasniji od unidirekcionog jer će algoritam sa obe strane relativno brzo da stigne do grešaka gde će putanje da ispita brže zbog "probijanja" sa obe strane i tunelovanja. Analitička potvrda ove činjenice još nije data, ali je simulacijama potvrđeno [8] da je raspodela broja računskih operacija (slučajna promenljiva C) takođe Paretoova sa približno udvostručenim Paretoovim eksponentom.

A. Numerički rezultati

Poređenje unidirekcionog i bidirekcionog algoritma može se izvršiti i putem računarskih simulacija. Na slici 1. su pokazani rezultati testiranja dve moguće softverske realizacije ovih algoritama na programskom jeziku C++.



Sl. 1. Računska kompleksnost stek agoritma (*Stack*) i bidirekcionog stek algoritma (*BiStack*)

Grafici su dati u logaritamskoj razmeri. Model kanala koji je korišćen je BSC (Binary Symmetric Channel) sa verovatnoćom prelaza $p=0.045$. Uzeta je dužina informacionog frejma od $L=50$ bita, memorija koda je $M=12$, a dužina tunela $T=5$.

Uočava se da je raspodela slučajne promenljive koja predstavlja broj računskih operacija (broj produženih čvorova) po dekodovanoj grani - Paretoova, pri čemu je Paretoov eksponent u slučaju bidirekcionog algoritma približno udvostručen za veliki broj operacija.

VIII. Zaključak

Rezultati koji su dobijeni pokazuju da je bidirekcioni stek algoritam postupak koji značajno smanjuje vremensku kompleksnost stek algoritma i time ubrzava rad prijemnika, što je vrlo važno u sistemima koji zahtevaju obradu podataka u realnom vremenu. Raspodela broja računskih operacija nije, međutim, još uvek izvedena. Može se na osnovu eksperimentalnih krivih samo pretpostaviti da je ona Paretoova. Ostaje da se pokaže da li je ova pretpostavka opravdana.

Literatura

[1] J. B. Anderson, S. Mohan, "Sequential Coding Algorithms: A Survey and Cost Analysis," *IEEE Trans. Comm.*, vol. COM-32, No. 2, pp. 169-176, Feb 1984.

[2] J. L. Masey, Coding and Complexity, CISM courses and lectures No. 216, Springer-Verlag, Wien, 1976.

[3] V. Šenk, "Bistack - A Bidirectional Stack Algorithm for Decoding Trellis Codes" *Proc. of XXXVI Conference on ETAN*, pp. 153-160, Kopaonik, Yugoslavia, 1992.

[4] A. J. Viterbi, J. Omura, Principles of Digital Communication and Coding, McGraw-Hill, Tokyo, 1979.

[5] R. M. Fano, "A heuristic discusion of probabilistic decoding," *IEEE Trans. Inf. Theory*, vol. IT-9, pp. 64-74, April 1963.

[6] S. Kallel, K. Li, "Bidirectional Sequential Decoding," *IEEE Trans. Inf. Theory*, vol. IT-43, No. 4, pp. 1319-1326, July 1997.

[7] V. Šenk, P. Radivojac, "The Bidirectional Stack Algorithm," *Proc.1997 IEEE Int. Symp.Inf. Th.*, ISIT'97, p. 500, Ulm, Germany, July 1997.

[8] V. Šenk, P. Radivojac, "The Bidirectional Stack Algorithm - Simulation Results," Proc. of TELSIKS '95, pp. 349-352, Niš, Yugoslavia, Oct. 1995.

[9] J. L Massey, "Error Bounds for Tree Codes, Trellis Codes and Convolutional Codes with Encoding and Decoding Procedures," CISM Courses and Lectures No. 216, Coding and Complexity, 1975.

Abstract

Main principles of decoding tree codes, especially their subclass - convolutional codes are given, with the emphasis on sequential decoding. Some properties of regular (unidirectional) and bidirectional stack algorithm are mentioned together with their performance comparison through computer simulations.

PERFORMANCE COMPARISON OF UNIDIRECTIONAL AND BIDIRECTIONAL STACK ALGORITHM

Mladen Kovačević