

# ارائه دو الگوریتم دیکدینگ هیبرید جدید با عملکرد بسیار خوب و پیچیدگی بسیار کم برای دیکدینگ کدهای LDPC

احسان اولیائی ترشیزی<sup>۱</sup>، دانش آموخته کارشناسی ارشد، حسین شریفی<sup>۲</sup>، استادیار

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه شهید بهشتی - تهران - ایران - eh.olyaei@gmail.com

۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه شهید بهشتی - تهران - ایران - sharifi@aryasatel.com

**چکیده:** در این مقاله، دو الگوریتم دیکدینگ هیبرید جدید برای دیکدینگ کدهای LDPC<sup>۱</sup> ارائه شده‌اند. الگوریتم اول تحت عنوان WBF<sup>۲</sup> پیشنهادی، نام‌گذاری شده و مبتنی بر ایده مکمل نمودن چندین بیت در هر تکرار است. استفاده از سازوکاری جدید برای تعیین تعداد و موقعیت بیت‌هایی که باید در هر تکرار مکمل شوند، باعث ایجاد سرعت همگرایی بسیار زیاد و هم‌چنین نیاز به تعداد تکرارهای بسیار کم برای این الگوریتم شده است، به‌گونه‌ای که این الگوریتم تنها با چند تکرار به بهترین جواب می‌رسد. هم‌چنین از دیگر مزایای الگوریتم WBF پیشنهادی می‌توان به اختلاف عملکرد بسیار ناچیز آن با الگوریتم قدرتمند Sum-Product اشاره نمود. نوآوری دیگر این مقاله، ترکیب الگوریتم WBF پیشنهادی با الگوریتم Sum-Product است. نتایج شبیه‌سازی مؤید این هستند که الگوریتم هیبرید پیشنهادی دوم مبتنی بر الگوریتم‌های WBF پیشنهادی و Sum-Product، دارای عملکرد احتمال خطایی دقیقاً مشابه با الگوریتم Sum-Product اما با سرعت همگرایی بسیار سریع‌تر از آن و هم‌چنین پیچیدگی بسیار کم‌تر در مقایسه با آن است.

**واژه‌های کلیدی:** کدهای LDPC، دیکدینگ تکراری هیبرید، عملکرد احتمال خطا، الگوریتم دیکدینگ Parallel WBF

## Two New Hybrid Decoding Algorithm for High-Performance Low-Complexity Decoding of LDPC Codes

E. Olyaei Torshizi<sup>1</sup>, H. Sharifi<sup>2</sup>

1, 2- Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran

**Abstract:** In this paper, two new hybrid algorithms for decoding LDPC codes are proposed. The first hybrid decoding algorithm that we call it Proposed WBF, is based on the idea of flipping variable multi bits in each iteration. Using a high-precision mechanism for determining the number and position of the bits to be flipped in each iteration, causes faster convergence rate and in a few iterations this algorithm reaches the best answer. From another advantages of this algorithm is its high performance with very negligible difference compared to Sum-Product algorithm. The other novelty in this paper is presentation of another new hybrid two stage decoding algorithm that combines the Proposed WBF and Sum-Product algorithms. Simulation results show that the second proposed hybrid decoding algorithm is a new and highly efficient algorithm with a performance equivalent to that of the Sum-Product algorithm and less complexity and fewer iterations required compared to that algorithm.

**Keywords:** LDPC Codes, Hybrid iterative decoding, Error probability performance, Parallel WBF decoding

تاریخ ارسال مقاله: ۱۳۹۲/۰۸/۱۲

تاریخ اصلاح مقاله: ۱۳۹۳/۰۵/۱۱

تاریخ پذیرش مقاله: ۱۳۹۳/۰۵/۲۰

نام نویسنده مسئول: احسان اولیائی ترشیزی

نشانی نویسنده مسئول: ایران - تهران - اوین - دانشگاه شهید بهشتی - دانشکده مهندسی برق و کامپیوتر

## ۱- مقدمه

که در واقع هم مبتنی بر تصمیم‌گیری نرم و هم مبتنی بر تصمیم‌گیری سخت هستند، به‌منظور کاهش فاصله میان عملکرد این دودسته الگوریتم ایجاد شده‌اند. شبیه‌سازی‌ها و بررسی‌های تحلیلی نشان می‌دهند که هنوز هم میان عملکرد الگوریتم‌های هیبرید و الگوریتم‌های مبتنی بر تصمیم‌گیری نرم اختلاف به نسبت زیادی وجود دارد.

در این مقاله به‌منظور ایجاد پلی برای پر نمودن فاصله موجود میان عملکرد الگوریتم‌های هیبرید و الگوریتم‌های مبتنی بر تصمیم‌گیری نرم، ابتدا یک الگوریتم هیبرید جدید با عنوان الگوریتم دیکدینگ WBF پیشنهادی ارائه می‌شود که علاوه بر اینکه می‌تواند در هر تکرار چندین بیت را مکمل نماید، سرعت همگرایی بالاتری نسبت به الگوریتم Modified WBF موجود، دارد. در ادامه با بیان ایده جدید ترکیب‌نمودن آن با الگوریتم‌های قدرتمند Sum-Product و Min-Sum، الگوریتم‌های بسیار کارآمد جدیدی با عملکردی معادل عملکرد الگوریتم‌های Sum-Product و Min-Sum و با پیچیدگی سخت‌افزاری و تعداد تکرارهای به‌مراتب کم‌تر از آن‌ها ارائه می‌دهیم.

سایر بخش‌های این مقاله بدین‌صورت هستند: بخش دوم کدهای LDPC را توصیف، تعریف‌ها و نمادگذاری‌های مورد‌استفاده را بیان می‌کند. بخش سوم الگوریتم‌های ارائه‌شده را تشریح و بخش چهارم نتایج حاصل از شبیه‌سازی را در مقایسه با سایر الگوریتم‌های موجود نشان می‌دهند. در نهایت، بخش پنجم نیز نتایج حاصل را جمع‌بندی و خلاصه مقاله را بیان می‌نماید.

## ۲- توصیف کدهای LDPC

## ۲-۱- معرفی کدهای LDPC و بیان تعریف‌های پایه

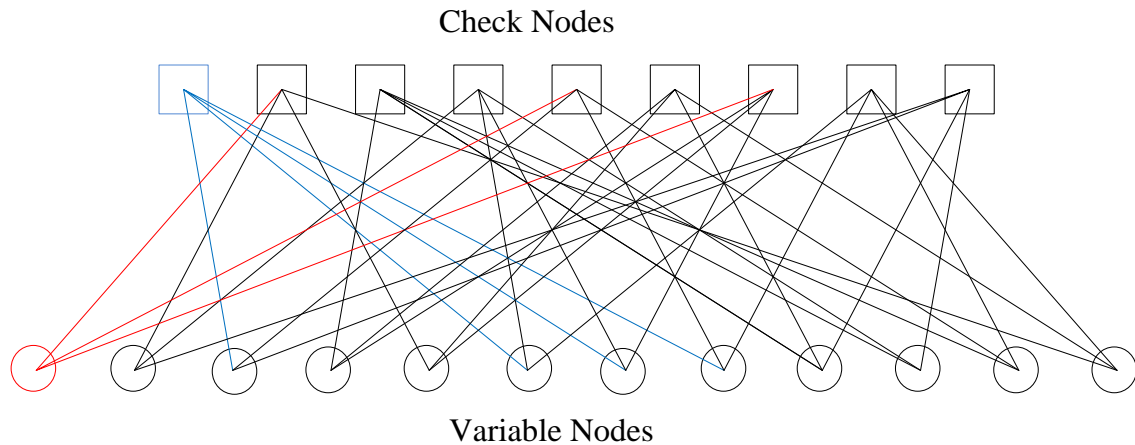
کدهای LDPC از خانواده کدهای بلوک خطی<sup>۴</sup> هستند. این کدها توسط یک ماتریس بیت توازن<sup>۵</sup> H بسیار تنک<sup>۱</sup> توصیف می‌شوند. نام‌گذاری این کدها از ویژگی تنک‌بودن ماتریس بیت توازن آن‌ها می‌آید که تنها شامل تعداد بسیار اندکی "1" در مقایسه با تعداد "0"ها است. یک ماتریس بیت توازن تنک، الگوریتم دیکدینگ را تسهیل می‌نماید و باعث طراحی دیکدرهایی با پیچیدگی کم می‌شود.

نمایش گرافیکی کدهای LDPC توسط یک گراف دوبخشی به نام گراف تنر<sup>۱۱</sup> صورت می‌پذیرد. در این گراف، مجموعه گره‌ها به دو مجموعه غیرتهی به نام‌های گره‌های متغیر<sup>۱۲</sup> (VNها) و گره‌های واری<sup>۱۳</sup> (CNها) طبقه‌بندی می‌شوند. هر لبه<sup>۱۴</sup> که متناظر با یک درایه "1" در ماتریس بیت توازن است، یک گره از مجموعه VNها را به یک گره از مجموعه CNها متصل می‌کند. VNها و CNها به ترتیب متناظر با ستون‌ها و سطرها<sup>۱۵</sup> ماتریس بیت توازن هستند و به ترتیب نشان‌دهنده یک سمبل بیت در کلمه کد و یک معادله پیریتی کد می‌باشند. ماتریس بیت توازن یک کد LDPC دارای n گره متغیر و m گره واری است.

با توجه به افزایش روزافزون کاربردهای مخابراتی و به دنبال آن نیاز به کدهای کانالی با عملکردهای بهتر در حضور کانال‌های نویزی، کدهای LDPC و مباحث مرتبط با آن‌ها به دلیل دارا بودن عملکرد بسیار عالی و نزدیک به حد شانون، به یکی از مباحث موردعلاقه محققان در تئوری کدینگ تبدیل شده‌اند. مزیت عمده آن‌ها این است که عملکردی نزدیک به ظرفیت شانون را روی مجموعه بزرگی از کانال‌های ارسال و ذخیره داده فراهم می‌آورند و هم‌چنین دارای الگوریتم‌هایی با پیچیدگی خطی برای دیکدینگ می‌باشند. کدهای LDPC برای اولین بار توسط آقای گالیجر<sup>۲</sup> در سال ۱۹۶۳ ابداع گردیدند [۱] و پس‌از آن به مدت حدود سه دهه مورد غفلت واقع شدند. در آن زمان، به پتانسیل خارق‌العاده و باورنکردنی این کدها پی برده نشد. مطالعه کدهای LDPC، در اواسط دهه ۱۹۹۰ با کار مک‌کی، لویی و دیگران [۲] احیا گشت. عملکرد بسیار عالی کدهای LDPC باعث شده تا این کدها همواره در کانون توجه باشند و الگوریتم‌های بسیار زیاد و متنوعی برای آن کدینگ<sup>۳</sup> و بخصوص دیکدینگ<sup>۵</sup> آن‌ها ارائه شود [۳].

الگوریتم‌های دیکدینگ موجود را می‌توان به سه دسته طبقه‌بندی نمود: الگوریتم‌های مبتنی بر تصمیم‌گیری سخت<sup>۶</sup>، الگوریتم‌های مبتنی بر تصمیم‌گیری نرم<sup>۷</sup> و الگوریتم‌های هیبرید. این الگوریتم‌های دیکدینگ عبارت‌اند از: الگوریتم دیکدینگ OSMLG [۴ و ۵]، الگوریتم دیکدینگ Bit Flipping گالیجر [۱]، الگوریتم Weighted MLG [۶]، الگوریتم‌های Weighted Bit Flipping [۷]، IWBF [۸] و Modified WBF [۹]، الگوریتم دیکدینگ A Posteriori Probability (APP) [۱]، الگوریتم تکراری مبتنی بر Belief Propagation که عموماً با عنوان الگوریتم دیکدینگ Sum Product شناخته می‌شود [۱۰-۱۳]، و سایر نسخه‌های الگوریتم BP همانند الگوریتم Min-Sum [۱۴]، و Normalized/Offset-BP [۱۴-۱۶]. الگوریتم‌های OSMLG و Bit-Flipping به الگوریتم‌های مبتنی بر تصمیم‌گیری سخت تعلق دارند. الگوریتم‌های APP، BP و سایر تقریب‌های آن جزء الگوریتم‌های مبتنی بر تصمیم‌گیری نرم و الگوریتم‌های WBF، IWBF و Modified WBF از جمله روش‌های دیکدینگ هیبرید هستند.

هر الگوریتم دیکدینگ تکراری در واقع یک مصالحه میان پیچیدگی و کیفیت عملکرد احتمال خطا برقرار می‌کند. بسیار بدیهی است که الگوریتم‌های مبتنی بر تصمیم‌گیری نرم که از ایده Message Passing برای انجام عملیات دیکدینگ خود استفاده می‌کنند، می‌توانند به عملکرد بهتری در مقایسه با الگوریتم‌های مبتنی بر تصمیم‌گیری سخت دست یابند. عملکرد بهتر این الگوریتم‌ها به دلیل استفاده آن‌ها از دنباله دریافتی از کانال برای انجام عملیات دیکدینگ است که این امر باعث افزایش بسیار زیاد پیچیدگی این الگوریتم‌ها در محاسبات و پیاده‌سازی سخت‌افزاری، می‌شود. عملکرد احتمال خطا الگوریتم‌های مبتنی بر تصمیم‌گیری سخت تقریباً حدود ۳dB پایین‌تر از الگوریتم‌های مبتنی بر تصمیم‌گیری نرم است. الگوریتم‌های دیکدینگ هیبرید



شکل (۱): تنر گراف متناظر با ماتریس بیت توازن H بیان شده در رابطه (۱)

قبل از ارسال، به دنباله دوقطبی  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  نگاشته می شود که در آن  $x_i = (2c_i - 1)$  و  $1 \leq i \leq N$  می باشند. هم چنین  $\mathbf{y} = (y_1, y_2, \dots, y_N)$  را دنباله دریافتی مبتنی بر تصمیم گیری نرم در خروجی فیلتر منطبق خروجی تعریف می کنیم. برای  $1 \leq i \leq N$  است که در آن  $y_i = x_i + n_i$  یک متغیر تصادفی گوسی با میانگین صفر و واریانس  $N_0/2$  است. یک دنباله باینری آغازی مبتنی بر تصمیم گیری سخت دنباله دریافتی،  $\mathbf{z}^{(0)} = (z_1^{(0)}, z_2^{(0)}, \dots, z_N^{(0)})$  به صورت زیر تعیین می گردد:

$$z_i^{(0)} = \begin{cases} 1, & \text{if } y_i \geq 0 \\ 0, & \text{if } y_i < 0 \end{cases} \quad (2)$$

برای هر دنباله مبتنی بر تصمیم گیری سخت  $\mathbf{z}$  که در انتهای هر تکرار ساخته می شود، می توان بردار سیندرم<sup>۱۸</sup> را به صورت  $\mathbf{s} = \mathbf{z} \cdot \mathbf{H}^T$  محاسبه محاسبه نمود. برای هر خروجی کانال  $y_i$  و  $1 \leq i \leq N$ ، نسبت درست نمایی لگاریتمی (LLR) را به صورت زیر تعریف می کنیم:

$$L_i \triangleq \ln \frac{P(c_i=1 | y_i)}{P(c_i=0 | y_i)} = \frac{4y_i}{N_0} \quad (3)$$

$L_i$ ،  $|L_i|$ ، قابلیت اطمینان<sup>۱۹</sup> تصمیم آغازی  $z_i^{(0)}$  نامیده می شود. برای هر بردار باینری  $\mathbf{v} = (v_1, v_2, \dots, v_N)$ ،  $HW(\mathbf{v})$  را به عنوان وزن همینگ<sup>۲۰</sup> بردار  $\mathbf{v}$  تعریف می کنیم.

### ۳- الگوریتم های هیبرید ارائه شده

#### ۳-۱- بیان ایده اصلی

اغلب الگوریتم های موجود برای دیکدینگ کدهای LDPC، مبتنی بر این فرض هستند که الگوریتم تکراری در هر تکرار می تواند تنها یک بیت از دنباله دریافتی را مکمل نماید. از جمله این الگوریتم ها که دارای عملکرد به مراتب خوبی هستند می توان الگوریتم دیکدینگ Standard

یک لبه بین هر گره متغیر و گره واری می تواند به وسیله یک درایه "1" در سطر و ستون متناظر در ماتریس بیت توازن نمایش داده شود. کدهای LDPC بر دو نوع منظم و نامنظم<sup>۱۵</sup> می باشند. در این مقاله تنها کدهای LDPC منظم مورد بررسی قرار می گیرند. در کدهای LDPC منظم، وزن تمام سطرها و تمام ستون های ماتریس بیت توازن یکسان و به ترتیب معادل با  $\omega_c$  و  $\omega_r$  می باشند به عبارت دیگر تعداد درایه های "1" موجود در تمام سطرها یکسان و معادل با  $\omega_r$  و تعداد درایه های "1" موجود در تمام ستون ها یکسان و معادل با  $\omega_c$  است. گراف تنر یک کد LDPC منظم با وزن های سطر و ستون  $\omega_c = 3$  و  $\omega_r = 4$  در شکل (۱) نشان داده شده است، هم چنین ماتریس بیت توازن متناظر با آن به صورت زیر است. به منظور روشن شدن ارتباط میان VN ها با CN ها به وسیله لبه ها، VN اول و لبه های متناظر با آن با رنگ قرمز و CN اول و لبه های متناظر با آن با رنگ آبی در شکل (۱) نشان داده شده است.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (1)$$

#### ۲-۲- نمادگذاری ها

فرض کنید یک کد LDPC باینری (N,K) برای کنترل خطا<sup>۱۶</sup> بر روی یک کانال با ورودی باینری همراه با نویز گوسی جمع شونده<sup>۱۷</sup> (BI-AWGN) با میانگین صفر و چگالی طیف توان  $N_0/2$  مورد استفاده قرار می گیرد و هم چنین سیگنالینگ مورد استفاده، BPSK با انرژی واحد است. یک کلمه کد  $\mathbf{c} = (c_1, c_2, \dots, c_N) \in \{GF(2)\}^N$

که در آن  $\omega_c$  وزن هر ستون ماتریس بیت توازن است. استفاده از این رابطه باعث بهبود عملکرد دیکدینگ و هم‌چنین کاهش پیچیدگی آن نسبت به الگوریتم‌های مشابه همانند الگوریتم Parallel WBF می‌شود. چراکه استفاده از این رابطه برخلاف مکانیزم موجود در الگوریتم Parallel WBF، ما را از نیاز به شبیه‌سازی برای تعیین بیت‌هایی که باید در هر تکرار مکمل گردند، بی‌نیاز می‌گرداند. ایده این بهسازی مبتنی بر این مشاهده است که برای ماتریس‌های کدهای LDPC، معمولاً وزن بردار سیندرم با تعداد خطاها افزایش می‌یابد. در واقع می‌توان چنین بیان نمود که میان تعداد بیت‌هایی که در هر تکرار باید مکمل گردند و وزن همینگ بردار سیندرم همبستگی بسیار بالایی وجود دارد و از آنجاکه هدف الگوریتم تکراری در نهایت رسیدن به بردار سیندرم تمام صفر است بنابراین در هر تکرار، وزن همینگ بردار سیندرم تخمین مناسبی است برای تعداد بیت‌هایی که باید مکمل گردند.

### ۳-۲- الگوریتم هیبرید WBF پیشنهادی

در میان الگوریتم‌هایی که استراتژی آن‌ها مبتنی بر مکمل نمودن چندین بیت در هر تکرار است، الگوریتم Parallel WBF [۲۲] عملکرد بسیار خوبی دارد. به همین دلیل از آن به‌عنوان پایه‌ای برای ایجاد و ارائه الگوریتم جدیدی با عنوان الگوریتم WBF پیشنهادی در این مقاله استفاده شده است. الگوریتم WBF ارائه‌شده نسبت به الگوریتم Parallel WBF دارای دو تغییر بسیار مهم است که باعث بهبود عملکرد آن می‌گردد. نخست آن‌که از یک سازوکار جدید و متفاوت برای تعیین تعداد بیت‌هایی که باید در هر تکرار مکمل گردند، استفاده می‌کند و تغییر مهم دیگر بعدی استفاده از یک مکانیزم تشخیص حلقه جدید برای این الگوریتم است. مکانیزم تشخیص حلقه برای اولین بار در [۲۳] برای الگوریتم‌هایی که تنها یک بیت را در هر تکرار مکمل می‌کنند ارائه شده است، اما سازوکاری که از آن برای تشخیص حلقه در این الگوریتم استفاده شده است، متفاوت با آن و برای الگوریتم‌هایی که چندین بیت را در هر تکرار مکمل می‌کنند طراحی گردیده، به‌گونه‌ای که بسیار وابسته به تعداد بیت‌هایی است که باید در هر تکرار مکمل شوند. روند اجرای الگوریتم WBF اصلاح‌شده پیشنهادی بدین صورت است:

در ابتدای تکرار  $k$ ام، ابتدا بردار سیندرم  $\mathbf{S}^{(k)}$  برای بردار دریافتی تعیین می‌شود. در صورتی که بردار سیندرم مخالف صفر باشد  $(\mathbf{S}^{(k)} \neq \mathbf{0})$ ، تعداد بیت‌هایی که باید در آن تکرار مکمل شوند به‌وسیله رابطه (۴) محاسبه می‌شوند. سپس الگوریتم WBF اصلاح‌شده پیشنهادی برای هر  $n \in [1, N]$ ، قابلیت اطمینان  $E_n(\alpha)$  را به‌صورت زیر محاسبه می‌کند:

$$E_n(\alpha) = \sum_{m \in M(n)} (2s_m - 1) |\omega_{n,m} - \alpha y_n| \quad (5)$$

که در آن  $\omega_{n,m}$  و  $s_m$  عبارت‌اند از:

WBF و دیگر نسخه‌های بهبودیافته آن همانند الگوریتم‌های IWBF و الگوریتم دیکدینگ Modified WBF را نام برد [۱۷]. این الگوریتم‌ها علاوه بر مکمل نمودن تنها یک بیت در هر تکرار، دارای نرخ همگرایی کندی هستند و هم‌چنین برای رسیدن به جواب قابل‌قبول نیاز به تعداد تکرارهای زیادی دارند. هم‌چنین فاصله میان عملکرد آن‌ها با الگوریتم‌های مبتنی بر تصمیم‌گیری نرم هنوز هم به نسبت زیاد است [۱۸ و ۱۹].

در این مقاله با ارائه ایده‌ای جدید مبتنی بر مکمل نمودن چندین بیت در هر تکرار و ایجاد تغییرات بنیادی دیگری در روند این الگوریتم‌های تکراری، الگوریتم هیبرید جدیدی را با عنوان الگوریتم WBF پیشنهادی با عملکرد بسیار بهتر و نزدیک‌تر به عملکرد الگوریتم SPA ارائه داده‌ایم. استفاده از این ایده باعث افزایش بسیار زیاد سرعت همگرایی تمام الگوریتم‌های جدیدی [۲۰-۲۱] می‌شود که عملکرد خوبی دارند اما تنها یک بیت را در هر تکرار مکمل می‌کنند و لذا دارای سرعت همگرایی کندی هستند. البته باید توجه داشت که مکمل نمودن اشتباه حتی یک بیت در یک تکرار باعث تسری خطا به‌صورت زنجیروار در کل الگوریتم شده و در نهایت منجر به واگرایی الگوریتم دیکدینگ تکراری می‌گردد. از این‌رو لزوم استفاده از مکانیزمی با دقت بسیار بالا برای تعیین تعداد و مکان بیت‌هایی که باید در هر تکرار مکمل گردند، به‌خوبی روشن است. ایده مکمل نمودن بیت‌ها به‌صورت موازی در هر تکرار باعث برتری‌های بسیاری نسبت به مکمل نمودن تنها یک بیت در هر تکرار می‌شود که از جمله آن می‌توان به افزایش بسیار سریع‌تر نرخ همگرایی الگوریتم و هم‌چنین کاهش تعداد تکرارها اشاره نمود. ویژگی جالب‌توجه دیگر الگوریتم‌هایی که در هر تکرار بیت‌ها را به‌صورت موازی مکمل می‌نمایند، این است که این الگوریتم‌ها قادر به اجتناب از بروز خطاهایی برای دیکدینگ تکراری می‌شوند که به علت محدودیت تعداد تکرارها برای الگوریتم به‌ویژه در مقادیر SNR پایین رخ می‌دهد، هر چند ممکن است تعداد این خطاها بسیار کم باشد. به‌عنوان مثال اگر دنباله دریافتی برای دیکدینگ یک کد LDPC دارای ۲۰۰۰ بیت و ماکزیمم تعداد تکرارهای الگوریتم دیکدینگ ۱۰۰ تکرار باشد، الگوریتم دیکدینگ تکراری مبتنی بر مکمل نمودن یک بیت در هر تکرار، در بهترین حالت قادر به تشخیص و اصلاح ۱۰۰ بیت خطا از دنباله دریافتی است و عملاً این الگوریتم‌ها برای دنباله‌های دریافتی که تعداد خطاهایشان بیش‌تر از ماکزیمم تعداد تکرارهای الگوریتم دیکدینگ است، ناکارآمد هستند و قادر به اصلاح تمام خطاهای چنین دنباله‌هایی نخواهند بود.

در الگوریتم‌های هیبرید ارائه‌شده در این مقاله، ما از رابطه جدید زیر برای تعیین تعداد بیت‌هایی که باید در هر تکرار مکمل شوند، استفاده می‌کنیم:

$$P_k = \left\lfloor \frac{HW(\mathbf{S}^{(k)})}{\omega_c} \right\rfloor \quad (4)$$

**تعاریف:**

K : شمارنده تکرارها

$\mathcal{E}_M^{(k)}$  : مجموعه CN های ارضاعنشد در تکرار K ام

$P_k$  : تعداد بیت‌هایی که باید در تکرار K ام مکمل شوند

$B^{(k)}$  : موقعیت بیت‌هایی که باید در تکرار K ام مکمل شوند

**ورودی:** بردار دریافتی از خروجی کانال (  $\mathbf{y}$  )

**خروجی:** بردار کلمه کد صحیح (  $\mathbf{z}^{(k)}$  )

**مقداردهی اولیه:**

در ابتدا  $k=0$  و ماکزیمم تعداد تکرارها را  $k_{\max}$  قرار می‌دهیم. هم‌چنین مجموعه موقعیت بیت‌هایی را که موجب بروز حلقه می‌شوند، در ابتدا تهی قرار می‌دهیم (  $LE_0 = \emptyset$  ).

**مرحله اول:**

محاسبه  $\mathbf{z}^{(0)} = \text{sgn}(\mathbf{y})$  از روی بردار دریافتی از کانال و سپس

$$S^{(0)} = H\omega(\mathbf{z}^{(0)} \cdot \mathbf{H}^T)$$

**مرحله دوم:**

برای هر  $m \in [1, M]$ ، با استفاده از  $\mathbf{z}^{(k)}$  مؤلفه‌های بردار سیندرم را با استفاده از رابطه  $s_m = \sum_n h_{m,n} z_n = \sum_n z_n \pmod{2}$  برای تشکیل

بردار سیندرم  $\mathbf{S}^{(k)}$  محاسبه نمایید.

**مرحله سوم:**

اگر  $\mathbf{S}^{(k)} = \mathbf{0}$  باشد، عملیات دیکدینگ متوقف و خروجی  $\mathbf{z}^{(k)}$  را به‌عنوان کلمه کد دیکدشده در نظر بگیرید؛ در غیر این صورت

$$P_k = \left\lfloor \frac{H\omega(\mathbf{S}^{(k)})}{\omega_c} \right\rfloor$$

ارضاعنشد را برای تشکیل  $\mathcal{E}_M^{(k)} = \{m | s_m = 1, m \in [1, M]\}$  جمع‌آوری نمایید.

**مرحله چهارم:**

اگر  $k < k+1$  و اگر  $k > k_{\max}$  اعلام شکست الگوریتم؛ در غیر این صورت رفتن به مرحله ۵.

**مرحله پنجم:**

برای هر  $n \in [1, N]$ ، با استفاده از  $\omega_{n,m} = \min_{i \in N(m), n} |y_i|$ ، مقادیر قابلیت

$$E_n(\alpha) = \sum_{m \in M(n)} (2s_m - 1) |\omega_{n,m}| - \alpha |y_n|$$

**مرحله ششم:**

برای هر  $m \in [1, M]$ ،  $n^* = \arg \max_{n \in N(m)} E_n(\alpha)$  را تعیین نموده و سپس

گره واری  $m$  ام، یک سیگنال "Flip" به گره متغیر  $n^*$  بنام

$$f(m \rightarrow n) = 0 \text{ برای } n \neq n^* \text{ داریم: } f(m \rightarrow n^*) = 1$$

**مرحله هفتم:**

$$s_m = \sum_n h_{m,n} z_n = \sum_{n \in N(m)} z_n \pmod{2} \quad (6)$$

و

$$\omega_{n,m} = \min_{i \in N(m), n} |y_i| \quad (7)$$

پس از آن الگوریتم برای هر  $n^* \in [1, M]$ ،  $m \in [1, M]$  را به‌عنوان موقعیت مکانی ماکزیمم شدن قابلیت اطمینان  $E_n(\alpha)$  به‌صورت زیر تعیین می‌نماید:

$$n^* = \arg \max_{n \in N(m)} E_n(\alpha) \quad (8)$$

سپس هر CN ارضاعنشد<sup>۱)</sup>، یک سیگنال "flip" را به سمت غیرقابل اطمینان‌ترین VN<sup>۲)</sup> ارسال می‌نماید. پس از آن هر گره متغیر تمام سیگنال‌های "flip" دریافتی را از نقطه‌نظر تعداد باهم جمع نموده و متریک تصمیم‌گیری  $F_n^{(k)}$  را تشکیل می‌دهد.

سپس الگوریتم، مجموعه  $B^{(k)}$  را که شامل موقعیت‌های  $P_k$  تا از بزرگ‌ترین مقادیر  $F_n^{(k)}$  است با استفاده از رابطه (۹) تعیین می‌کند.

$$j_i^{(k)} = \arg \max_{1 \leq n \leq N} F_n^{(k)} \quad (9)$$

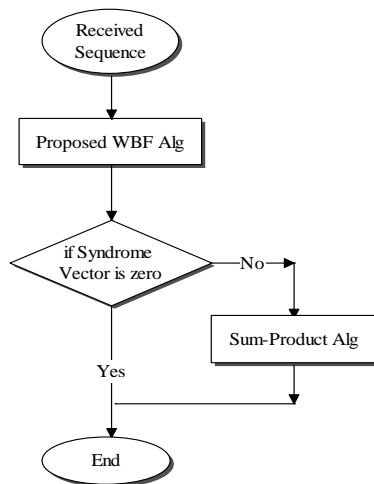
درنهایت نیز الگوریتم تمام موقعیت بیت‌های واقع در مجموعه  $B^{(k)}$  را در دنباله  $\mathbf{z}^{(k-1)}$  مکمل نموده و دنباله جدید  $\mathbf{z}^{(k)}$  را تولید می‌نماید. پارامتر بهینه  $\alpha$  در رابطه  $E_n(\alpha)$  برای هر کد LDPC خاص باید از طریق شبیه‌سازی تعیین گردد. شبیه‌سازی‌ها نشان می‌دهند که نرخ خطا خیلی به مقدار  $\alpha$  حساس نیست و هم‌چنین مقدار بهینه  $\alpha$  به میزان SNR وابسته نیست. جدول (۱)، مقادیر  $\alpha$  بهینه را برای تعدادی از کدهای FG-LDPC عملی، نشان می‌دهد.

**جدول (۱): میزان بهینه  $\alpha$  برای برخی کدهای [A] EG/PG LDPC**

نوع کد	(N,K)	$\omega_r$	$\omega_c$	مقدار بهینه $\alpha$
EG-LDPC	(۳۷ و ۶۳)	۸	۸	۱/۲
PG-LDPC	(۴۵ و ۷۳)	۹	۹	۱/۲
EG-LDPC	(۱۷۵ و ۲۵۵)	۱۶	۱۶	۱/۵
PG-LDPC	(۱۹۱ و ۲۷۳)	۱۷	۱۷	۱/۵
EG-LDPC	(۷۸۱ و ۱۰۲۳)	۳۲	۳۲	۲/۱
PG-LDPC	(۸۱۳ و ۱۰۵۷)	۳۳	۳۳	۲/۱
EG-LDPC	(۳۳۶۷ و ۴۰۹۵)	۶۴	۶۴	۳
PG-LDPC	(۳۴۳۱ و ۴۱۶۱)	۶۵	۶۵	۳

مراحل اجرای الگوریتم دیکدینگ تکراری WBF اصلاح‌شده به‌صورت زیر است:

**الگوریتم دیکدینگ هیبرید WBF پیشنهادی**



شکل (۲): بلوک دیاگرام متناظر با الگوریتم دیکدینگ

الگوریتم هیبرید ترکیبی پیشنهادی میزان زیادی کسب می‌کند. الگوریتم SPA دارد.

شاید در ابتدای امر این‌طور به نظر برسد که استفاده از ایده ترکیب دو الگوریتم WBF پیشنهادی و SPA (و یا Min-Sum) باعث افزایش حجم محاسبات و در نتیجه افزایش پیچیدگی عملیات دیکدینگ می‌شود، اما در واقعیت این‌گونه نیست چراکه میزان مشارکت الگوریتم دیکدینگ SPA در دیکدینگ هیبرید ترکیبی بسیار کم است. میزان این مشارکت تقریباً معادل با WER دیکدینگ WBF پیشنهادی است. یک خطای کلمه<sup>۲۳</sup>، هنگامی روی می‌دهد که یک پرتی چک ارضا نشود یا بردار دیکدشده، یک کلمه کدی متفاوت از کلمه کد ارسالی باشد (خطای غیرقابل شناسایی). اما از آنجاکه خطای غیرقابل شناسایی به‌ویژه برای کدهایی با طول بلند به‌ندرت روی می‌دهد، بنابراین، نرخ خطای کلمه WER تقریباً معادل با احتمال عدم موفقیت پرتی چک‌ها است. لزوم ارائه و استفاده از الگوریتم WBF پیشنهادی برای مرحله اول دیکدینگ تکراری هیبرید به‌خوبی روشن است چراکه پرواضح است هرچقدر عملکرد مرحله اول دیکدینگ بهتر باشد و از الگوریتم قدرتمندتری برای این منظور استفاده شود، میزان نیاز به استفاده از الگوریتم SPA و یا MSA برای مرحله دوم الگوریتم کم‌تر است و بدین ترتیب پیچیدگی الگوریتم هیبرید ترکیبی کاهش می‌یابد. در نواحی با SNR بالا، WER دیکدینگ WBF پیشنهادی کم است، یعنی الگوریتم SPA به‌ندرت فعالیت می‌کند، بنابراین پیچیدگی محاسباتی الگوریتم هیبرید ارائه‌شده، کاهش می‌یابد. برای حالتی که الگوریتم WBF پیشنهادی می‌تواند خطا را شناسایی و اصلاح نماید، الگوریتم SPA نیز قادر به انجام آن است اما برای حالتی که الگوریتم مرحله اول به شکست می‌انجامد، الگوریتم هیبرید ترکیبی وارد مرحله دوم شده و از فیلتر الگوریتم SPA می‌گذرد، بنابراین می‌توان انتظار داشت که الگوریتم دیکدینگ هیبرید ترکیبی، دقیقاً دارای همان عملکرد الگوریتم SPA است. از آنجاکه پیچیدگی سخت‌افزاری دیکدر WBF پیشنهادی بسیار کم‌تر از دیکدر SPA است، دیکدر هیبرید ترکیبی

برای هر  $n \in [1, N]$ ، سیگنال‌های "Flip" را از همه گره‌های واری ارضاء نشده ممکن برای به دست آوردن متغیر تصمیم‌گیری مکمل نمودن

$$F_n^{(k)} = \sum_{m \in \mathcal{E}_M^{(k)}} f(m \rightarrow n) \text{ جمع کنید.}$$

#### مرحله هشتم:

مجموعه  $B^{(k)} = \{j_1^{(k)}, j_2^{(k)}, \dots, j_{P_k}^{(k)}\} \notin LE_k$  را که شامل موقعیت‌های  $P_k$  تا از بزرگ‌ترین مقادیر  $(F_n^{(k)}, 1 \leq n \leq N)$  است، با استفاده از رابطه

$$j_i^{(k)} = \arg \max_{1 \leq n \leq N} F_n^{(k)} \text{ تعیین نمایید. (بدیهی است که } 1 \leq j_{P_k}^{(k)} < \dots < j_2^{(k)} < j_1^{(k)} \leq N \text{)}$$

#### مرحله نهم:

برای هر  $k$ ، اگر  $B^{(k)} = B^{(k-1)}$  باشد، قرار دهید:  $LE_k = LE_k \cup \{B^{(k)}\}$  و  $P_k \leftarrow P_k - 1$ . سپس برگشتن به مرحله ۸؛ در غیر این صورت رفتن به مرحله ۱۰.

#### مرحله دهم:

تعیین  $z^{(k)}$  با مکمل نمودن موقعیت بیت‌های حاضر در مجموعه  $B^{(k)}$  در دنباله  $z^{(k-1)}$  و سپس رفتن به مرحله ۱.

### ۴- الگوریتم دیکدینگ هیبرید جدید پیشنهادی مبتنی بر ترکیب الگوریتم WBF پیشنهادی با الگوریتم Sum-Product (یا Min-Sum)

نوآوری دیگر این مقاله ارائه یک الگوریتم دیکدینگ تکراری هیبرید دومرحله‌ای است که دو الگوریتم دیکدینگ تکراری را برای کاهش پیچیدگی محاسباتی کدهای (FG-LDPC) باهم ترکیب می‌نماید. دیکدینگ تکراری هیبرید ارائه‌شده می‌تواند شامل یک یا دو مرحله باشد. ترتیب اجرای این دو نوع الگوریتم بدین‌صورت است که ابتدا الگوریتم دیکدینگ WBF پیشنهادی اجرا می‌شود. اگر همه معادلات پرتی چک ارضا شوند، فرایند دیکدینگ با موفقیت به اتمام می‌رسد و خروجی به‌عنوان بردار دیکدشده در نظر گرفته می‌شود. در غیر این صورت، عملیات دیکدینگ به مرحله دوم منتقل می‌شود و با استفاده از الگوریتم قدرتمند Sum-Product ادامه می‌یابد. بلوک دیاگرام متناظر با الگوریتم هیبرید ترکیبی پیشنهادی در شکل (۲) نمایش داده شده است.

تلفات بهره کدینگ تقریباً حدود  $0.5\text{dB}$  در مقایسه با الگوریتم SPA در  $BER = 10^{-7}$  می‌باشند.

## ۵-۲- بررسی متوسط تعداد تکرارهای موردنیاز برای

### الگوریتم WBF پیشنهادی

در این قسمت ماکزیمم تعداد تکرارها به‌منظور دیکدینگ کد PG-LDPC(1057,813) برای الگوریتم‌های WBF استاندارد، Modified WBF و WBF پیشنهادی،  $50$  قرار داده شده است. شکل (۴) میانگین تعداد تکرارها را برحسب  $E_b/N_0$  برای این الگوریتم‌ها باهم مقایسه می‌نماید. با توجه به این شکل، الگوریتم WBF پیشنهادی در  $BER = 10^{-5}$  تنها به حدود ۱ تکرار نیاز دارد. حال آنکه، الگوریتم‌های WBF استاندارد و Modified WBF به بیش‌تر از ۲۰ تکرار نیاز دارند. الگوریتم WBF پیشنهادی ما دارای پیچیدگی تقریباً یکسانی در هر تکرار نسبت به دو الگوریتم دیگر است اما همان‌طور که از شکل نیز پیداست این الگوریتم هم در نواحی با SNR بالا و هم در نواحی با SNR پایین به تعداد تکرار بسیار کم‌تری نسبت به دو الگوریتم دیگر نیاز دارد لذا سرعت همگرایی آن بسیار سریع‌تر است. دلیل آن نیز بسیار واضح است چراکه در الگوریتم دیکدینگ WBF پیشنهادی، حجم عمده عملیات در تکرارهای اول و به‌ویژه در تکرار اول انجام می‌پذیرد و این به دلیل استفاده از رابطه (۴) برای تعیین تعداد بیت‌هایی است که باید در هر تکرار مکمل شوند. روند تغییرات پارامتر  $P^{(k)}$  به‌صورت نزولی است به‌گونه‌ای که به‌صورت تصاعدی کاهش می‌یابد. نتایج شبیه‌سازی نشان می‌دهند که با استفاده از ماکزیمم ۱۰ تکرار می‌توان به بهترین عملکرد برای این الگوریتم دست یافت و افزایش بیش‌تر تعداد تکرارها تأثیر محسوسی بر روی عملکرد آن ندارد.

## ۵-۳- عملکرد احتمال خطای دیکدینگ تکراری هیبرید

### مبتنی بر ترکیب الگوریتم WBF پیشنهادی با الگوریتم‌های SPA و MSA

در دیکدینگ هیبرید ترکیبی، ماکزیمم تعداد تکرارها برای الگوریتم‌های WBF پیشنهادی و SPA به ترتیب ۱۰ و ۱۰۰ تکرار قرار داده شده‌اند. شکل‌های (۵) و (۶) عملکرد خطای کد PG-LDPC(1057,813) را برای الگوریتم‌های دیکدینگ تکراری هیبرید ترکیبی در قیاس با الگوریتم‌های دیکدینگ Standard WBF، Modified WBF، WBF اصلاح‌شده، SPA و MSA نشان می‌دهند.

ارائه‌شده، پیچیدگی سخت‌افزاری قابل‌مقایسه‌ای با دیکدر SPA دارد. به‌منظور مقایسه پیچیدگی میان الگوریتم‌های ارائه‌شده با الگوریتم‌های موجود، در جدول ۲ پیچیدگی محاسباتی در هر تکرار برای الگوریتم‌های مختلف به‌صورت دقیق محاسبه شده است.

جدول (۲): مقایسه پیچیدگی محاسباتی الگوریتم‌های مختلف

الگوریتم	پیچیدگی محاسباتی برای تکرار Kام
WBF استاندارد	$N - 1 + \omega_c \omega_r$
Modified WBF	$N - 1 + \omega_c \omega_r$
WBF پیشنهادی	$2(N - 1) + \omega_c \omega_r + N\omega_c$
SPA	$N(3\omega_c + 1)$
هیبرید پیشنهادی	$3N(\omega_c + 1) + \omega_c(\omega_r + N) - 2$

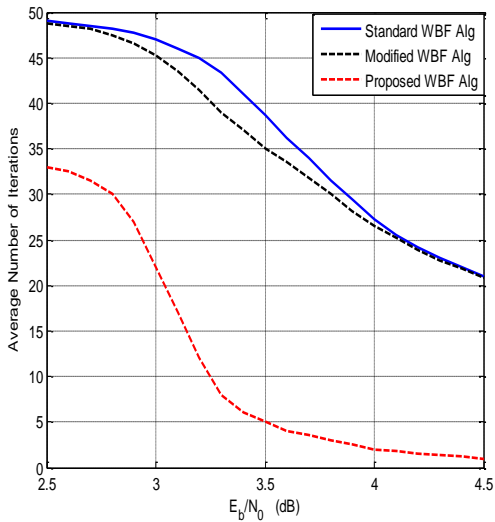
## ۵- نتایج شبیه‌سازی

پارامترهای مهم در ارزیابی یک الگوریتم دیکدینگ عبارت‌اند از: عملکرد احتمال خطای الگوریتم، میزان پیچیدگی، تعداد تکرارهای موردنیاز برای همگرایی الگوریتم و سرعت همگرایی آن. در این مقاله این پارامترهای مهم را برای الگوریتم‌های ارائه‌شده مورد بررسی قرار داده و آن را هم با پارامترهای الگوریتم‌های مبتنی بر تصمیم‌گیری نرم و هم با پارامترهای الگوریتم‌های هیبرید مقایسه نموده‌ایم. برای انجام شبیه‌سازی از یک کد LDPC با هندسه تصویری PG-LDPC(1057,813) استفاده شده است.

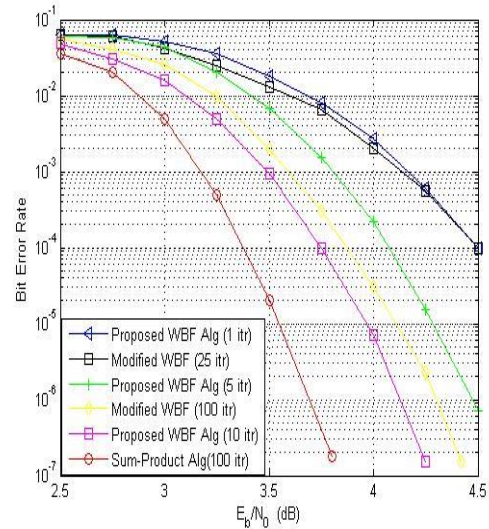
## ۵-۱- عملکرد BER الگوریتم هیبرید WBF پیشنهادی

شکل (۳) عملکرد BER کد PG-LDPC(1057,813) را نشان می‌دهد. فاکتورهای وزن‌دهی بهینه  $\alpha$  برای الگوریتم‌های دیکدینگ Modified WBF، 1.8 [۲۴] و 2.1 برای الگوریتم WBF پیشنهادی، قرار داده شده‌اند. با توجه به این شکل می‌توان دید که الگوریتم هیبرید WBF پیشنهادی بسیار سریع‌تر از الگوریتم Modified WBF همگرا می‌شود. برای مثال، عملکرد الگوریتم WBF پیشنهادی تنها با ۱ تکرار می‌تواند به عملکرد الگوریتم Modified WBF با ۲۵ تکرار دست یابد. الگوریتم WBF پیشنهادی با ۱۰ تکرار عملکرد به‌مراتب بهتری از الگوریتم Modified WBF با ۱۰۰ تکرار همگرا دارد. لذا نتایج شبیه‌سازی مؤید آن هستند که الگوریتم WBF پیشنهادی ارائه‌شده با تعداد تکرارهایی به‌مراتب کم‌تر می‌تواند به عملکرد الگوریتم Modified WBF بعد از همگرایی دست یابد. اما همچنان هر دو الگوریتم Modified WBF با ۱۰۰ تکرار و WBF پیشنهادی با ۱۰ تکرار دارای

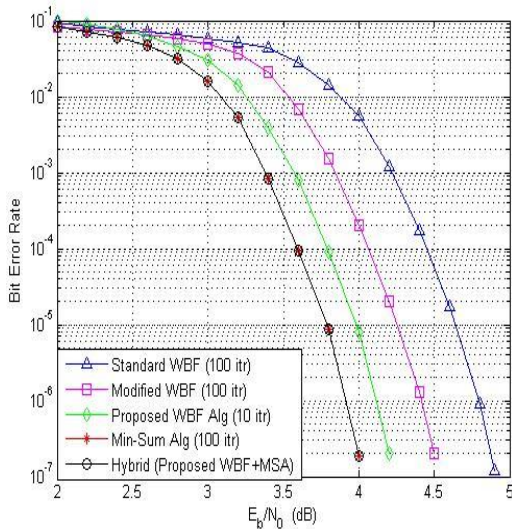




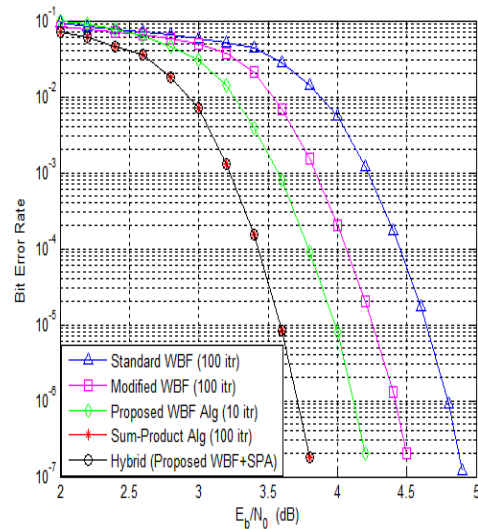
شکل (۴): مقایسه متوسط تعداد تکرارهای موردنیاز برای همگرایی الگوریتم دیکدینگ WBF پیشنهادی با الگوریتم‌های دیکدینگ Standard WBF و Modified WBF برای کد PG-LDPC(1057,813)



شکل (۳): مقایسه عملکرد BER الگوریتم دیکدینگ WBF پیشنهادی با الگوریتم‌های دیکدینگ Standard WBF, Modified WBF و Sum-Product برای کد PG-LDPC(1057,813)



شکل (۶): مقایسه عملکرد BER الگوریتم‌های دیکدینگ WBF پیشنهادی و الگوریتم هیبرید پیشنهادی (Proposed WBF + MSA) با الگوریتم‌های دیکدینگ Standard WBF, Modified WBF و Min-Sum برای کد PG-LDPC(1057,813)



شکل (۵): مقایسه عملکرد BER الگوریتم‌های دیکدینگ WBF پیشنهادی و الگوریتم هیبرید پیشنهادی (Proposed WBF + SPA) با الگوریتم‌های دیکدینگ Standard WBF, Modified WBF و Sum-Product برای کد PG-LDPC(1057,813)

$5 \times 10^{-3}$  است و مفهوم آن این است که از هر ۱۰۰۰ بار تکرار دیکدینگ هیبرید ترکیبی، تنها ۵ بار آن نیازمند الگوریتم SPA یا MSA است. همچنین در  $E_b/N_0 = 4.5 dB$ ، در میان  $10^5$  بار تکرار دیکدینگ هیبرید ترکیبی، تنها یک‌بار نیاز به اجرای الگوریتم SPA یا MSA است و بدین ترتیب با افزایش میزان نسبت سیگنال به نویز، میزان نیاز به اجرای مرحله دوم الگوریتم هیبرید یعنی الگوریتم SPA یا MSA کاهش می‌یابد.

شکل (۷) نیز BER و WER الگوریتم WBF اصلاح‌شده را با BER الگوریتم‌های SPA، MSA و هیبریدهای پیشنهادی مبتنی بر ترکیب الگوریتم‌های WBF پیشنهادی با SPA و MSA مقایسه می‌نماید. با توجه به شکل‌های (۵) تا (۷)، مشاهده می‌شود که دیکدینگ تکراری هیبرید ترکیبی می‌تواند به عملکرد تقریباً یکسانی با الگوریتم‌های SPA و MSA دست یابد. در شکل (۷) به‌وضوح روشن است که WER الگوریتم WBF پیشنهادی در  $E_b/N_0 = 4 dB$  معادل با

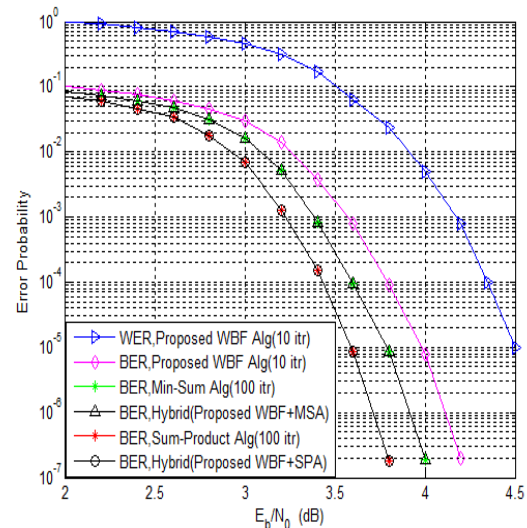


جدول (۳): مقایسه پیچیدگی محاسباتی الگوریتم‌های مختلف برای کد

$\omega_e = \omega_r = 33$  با PG-LDPC(1057,813)

الگوریتم	پیچیدگی محاسباتی تکرار ام K	تعداد تکرارها	پیچیدگی محاسباتی کل الگوریتم
WBFB استاندارد	۲۱۴۵	۱۰۰	۲۱۴۵۰۰
Modified WBFB	۲۱۴۵	۱۰۰	۲۱۴۵۰۰
WBFB پیشنهادی	۳۸۰۸۲	۱۰	۳۸۰۸۲۰
SPA	۱۰۵۷۰۰	۱۰۰	۱۰۵۷۰۰۰۰
هیبرید پیشنهادی	۱۴۳۷۸۲	۱۰	۱۴۳۷۸۲۰

این افزایش پیچیدگی در ازای حدود ۰/۳dB بهبود در عملکرد احتمال خطا حاصل شده است (شکل (۵)). پیچیدگی محاسباتی الگوریتم قدرتمند SPA در مقایسه با الگوریتم WBFB پیشنهادی در هر تکرار حدود ۲/۷ برابر بیشتر و در کل الگوریتم حدود ۲۷ برابر بیشتر است در حالی که اختلاف عملکرد میان این دو الگوریتم تنها حدود ۰/۴dB است. هم‌چنین شایان ذکر است که الگوریتم WBFB پیشنهادی تنها با ۱۰ تکرار به جای ۱۰۰ تکرار به نتیجه مطلوب می‌رسد و بنابراین سرعت همگرایی بسیار سریع‌تری نسبت به الگوریتم SPA دارد. با توجه به نتایج جدول (۳) پیچیدگی محاسباتی الگوریتم هیبرید پیشنهادی دومرحله‌ای برای کل الگوریتم حداقل حدود ۱۴ برابر کم‌تر از الگوریتم SPA است در حالی که این دو الگوریتم با توجه به شکل (۵) عملکرد کاملاً یکسانی دارند. همان‌طور که پیش‌ازین نیز بیان شد، در اجرای الگوریتم هیبرید پیشنهادی لزوماً همیشه نیازی به اجرای مرحله دوم الگوریتم نیست، اما در نتایج مربوط به پیچیدگی الگوریتم هیبرید پیشنهادی در جدول (۳)، محاسبات با فرض اینکه در تمام تکرارها مرحله دوم نیز اجرا می‌شود، انجام شده است. با وجود اینکه پیچیدگی الگوریتم هیبرید در هر تکرار اندکی بیشتر از الگوریتم SPA است، شاید این‌طور تصور شود که الگوریتم SPA دارای پیچیدگی محاسباتی کم‌تری است اما پرواضح است که پیچیدگی الگوریتم هیبرید ترکیبی نسبت به الگوریتم‌های SPA و MSA بسیار کم‌تر است، چراکه اولاً در اجرای الگوریتم دومرحله‌ای تکراری هیبرید، لزوماً همیشه نیازی به انجام مرحله دوم دیکدینگ نیست و تنها اجرای مرحله اول دیکدینگ منجر به تشخیص صحیح کلمه کد ارسالی و ایجاد بردار سیندرم تمام صفر می‌گردد؛ ثانیاً همان‌طور که پیش‌ازین نیز اشاره شد، با افزایش میزان SNR میزان نیاز به اجرای الگوریتم SPA و یا MSA بسیار کاهش می‌یابد. از طرفی با توجه به ماهیت الگوریتم هیبرید مبتنی بر مکمل نمودن بیت‌ها به‌صورت موازی، تعداد تکرارهای لازم برای الگوریتم هیبرید پیشنهادی جهت همگرایی و رسیدن به عملکرد احتمال خطا مطلوب بسیار کم‌تر از الگوریتم SPA است. بنابراین می‌توان این‌طور نتیجه‌گیری نمود که الگوریتم هیبرید ارائه‌شده، عملکرد یکسانی با الگوریتم‌های قدرتمند SPA و MSA دارد در حالی که تعداد تکرارها و میزان پیچیدگی عملکرد آن بسیار کم‌تر از آن‌ها است و از این‌رو الگوریتم پیشنهادی ما بر الگوریتم‌های قدرتمندی همانند SPA



شکل (۷): مقایسه عملکرد BER و WER الگوریتم WBFB پیشنهادی با BER الگوریتم‌های دیکدینگ هیبرید (Proposed WBFB+MSA)، هیبرید (Proposed WBFB+SPA) و Min-sum و Sum-Product برای کد PG-LDPC(1057,813)

با توجه به شکل (۵) بسیار واضح است که عملکرد الگوریتم WBFB اصلاح‌شده تنها حدود ۰/۵dB با الگوریتم قدرتمند SPA اختلاف دارد و در حدود ۰/۳dB عملکرد آن از الگوریتم Modified WBFB بهتر است؛ این بهبود عملکرد در حالی است که الگوریتم WBFB پیشنهادی، به دلیل دارا بودن مکانیزمی که در آن چندین بیت را به‌صورت موازی در هر تکرار مکمل می‌نماید، به تعداد تکرارهای بسیار کم‌تری نیز برای همگرایی نیاز دارد.

عملکرد الگوریتم هیبرید ترکیبی ارائه‌شده با توجه به شکل‌های (۵) تا (۷)، دقیقاً منطبق بر عملکرد الگوریتم‌های SPA و MSA است با این تفاوت که تعداد تکرارهای موردنیاز برای آن بسیار کم‌تر از الگوریتم‌های SPA و MSA است.

#### ۴-۵- پیچیدگی محاسباتی الگوریتم‌های پیشنهادی

به‌منظور روشن شدن هرچه بیشتر کاهش پیچیدگی الگوریتم هیبرید ترکیبی ارائه‌شده نسبت به الگوریتم SPA، مجدداً کد PG-LDPC(1057,813) را با  $\omega_e = \omega_r = 33$  در نظر می‌گیریم. پیچیدگی محاسباتی یا به عبارتی تعداد جمع‌کننده‌های موردنیاز جهت دیکد نمودن این کد، برای الگوریتم‌های دیکدینگ متفاوت در جدول (۳) به‌صورت دقیق محاسبه شده‌اند.

با توجه به نتایج حاصل‌شده از جدول (۳) می‌توان دید که برای کد مذکور الگوریتم‌های WBFB استاندارد و Modified WBFB دارای پیچیدگی یکسانی هستند. تعداد جمع‌کننده‌های موردنیاز برای الگوریتم WBFB پیشنهادی در کل الگوریتم، حدود ۱/۷ برابر بیش‌تر از الگوریتم‌های WBFB استاندارد و Modified WBFB است.

- [2] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645-1646, Aug. 1996.
- [3] T. M. N. Ngatched and M. Bossert, "Two bit-flipping algorithms for low-density parity-check codes," *IEEE Trans. Commun.*, vol. 57, no. 3, pp. 591-596, Mar. 2009.
- [4] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1983.
- [5] J. L. Massey, *Threshold Decoding*, Cambridge, MA: MIT Press, 1963.
- [6] V. D. Kolesnik, "Probability decoding of majority codes," *Prob. Peredachi. Inform.*, vol. 7, pp. 3-12, July 1971.
- [7] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans Inform Theory*, vol. 47, pp. 2711-2736, Nov. 2001.
- [8] Z. Liu and D. A. Pados, "A decoding algorithm for finite geometry LDPC codes," *IEEE Trans. Commun.*, vol. 53, pp. 415-421, Mar. 2005.
- [9] J. Zhang, M. P. C. Fossorier, "A modified weighted bit-flipping decoding of low-density parity-check codes," *IEEE Comm. Lett.*, vol. 8, pp. 165-168, 2004.
- [10] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, IT-45, pp. 399-432, Mar. 1999.
- [11] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular codes," *IEEE Trans. Inform Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.
- [12] F. R. Kschischang, B. J. Frey and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498-519, Feb. 2001.
- [13] R. Lucas, M. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of onestep majority logic decodable codes based on belief propagation," *IEEE Trans. Commun.*, vol. 48, pp. 931-937, June 2000.
- [14] N. Wiberg, "Codes and decoding on general graphs. Ph.D. dissertation", Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.
- [15] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low density parity check codes based on belief propagation," *IEEE Trans. on Comm.*, vol. 47, pp. 673-680, May 1999.
- [16] J. Chen and M. P. C. Fossorier, "Density evolution of two improved BP-based algorithms for LDPC decoding," *IEEE Comm. Lett.*, vol. 50, pp. 406-414, Mar. 2002.
- [17] اولیایی ترشیزی، بررسی عملکرد دیکدینگ کدهای LDPC با استفاده از الگوریتم‌های BP و BF، پایان‌نامه کارشناسی ارشد، دانشگاه شهید بهشتی، تهران، ش ص ۷۵-۴۰، ۱۳۹۱.
- [18] Ma. Kexiang, L. Yongzhao, Z. Hailin, "Fast weighted bit flipping algorithm for higher-speed decoding of Low-Density Parity-Check codes," *china communications*, vol. 10, pp. 114-119, 2013.
- [19] T.C. Chen, "Adaptive-weighted multibit-flipping decoding of lowdensity parity-check codes based on ordered statistics," *IET communications*, vol. 7, pp. 1517-1521, 2013.
- [20] K. Ma, Y. Li and H. Zhang, "Fast weighted bit flipping algorithm for higher-speed decoding of Low-Density Parity-Check codes," *china communications*, vol. 10, pp. 114-119, 2013.
- [21] Y. Yang, J. Li, H. Yu and Y. Xu, "Improved Weighted Bit Flip Voting decoding algorithm for generalized LDPC," *International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1-6, 2012.
- [22] X. Wu, C. Zhao and X. You, "Parallel weighted bit-flipping decoding," *IEEE Commun. Lett.*, vol. 11, pp. 671-673, no. 8, Aug. 2007.
- [23] Z. Liu and D. A. Pados, "A decoding algorithm for finite-geometry LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 415-421, Mar. 2005.
- [24] J. Li and X. Zhang, "Hybrid iterative decoding for low-density Parity-Check Codes based on finite geometries," *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 29-31, Jan. 2008.

MSA برتری دارد، همچنان نتایج شبیه‌سازی نیز مؤید این موضوع هستند.

## ۶- نتیجه

در این مقاله بر پایه الگوریتم Parallel WBF موجود، دو الگوریتم هیبرید بهبودیافته جدید را برای دیکدینگ کدهای LDPC ارائه نمودیم. این الگوریتم‌ها از یک مکانیزم جدید برای تعیین تعداد و موقعیت بیت‌هایی که باید در هر تکرار مکمل شوند، استفاده می‌کنند. در این مقاله یک رابطه جدید برای این منظور ارائه شده است. این معادله مبتنی بر این مشاهده است که یک همبستگی بسیار قوی میان تعداد بیت‌هایی که باید در هر تکرار مکمل شوند و وزن همینگ بردار سیندرم وجود دارد. در مقایسه با دیگر الگوریتم‌های هیبرید قدرتمند همانند الگوریتم‌های Standard WBF و Modified WBF الگوریتم هیبرید پیشنهادی اول تحت عنوان الگوریتم WBF پیشنهادی، عملکرد احتمال خطای بهتر، سرعت همگرایی بسیار بیشتر و همچنین نیاز به تعداد تکرارهای بسیار کمتری را از خود نشان می‌دهد. سپس ما این الگوریتم را با الگوریتم قدرتمند SPA و همچنین Min-Sum برای ارائه یک الگوریتم هیبرید دومرحله‌ای جدید ترکیب نموده‌ایم. الگوریتم ترکیبی حاصل دارای عملکردی دقیقاً همانند عملکرد الگوریتم SPA و یا MSA اما با پیچیدگی به مراتب کمتر نسبت به آن است.

## پیشنهادها

برای مرحله دوم دیکدینگ، می‌توان الگوریتم SPA استاندارد را با تعدادی از نسخه‌های ساده‌شده آن برای دستیابی به یک مصالحه خوب میان پیچیدگی دیکدینگ و عملکرد خطایگزین نمود. در این مقاله ما همچنین از الگوریتم دیکدینگ Min-Sum نیز به‌عنوان جایگزینی برای الگوریتم SPA برای مرحله دوم الگوریتم هیبرید ترکیبی دومرحله‌ای ارائه‌شده، استفاده نمودیم. هم‌چنین می‌توان از نسخه‌های دیگری مانند Normalized BP و یا Offset BP برای بهبود عملکرد آن نیز استفاده نمود.

## سپاسگزاری

این مقاله تحت حمایت مرکز تحقیقات مخابرات ایران با شماره قرارداد ۵۰/۶۹۹۲ قرار دارد. نویسندگان این مقاله از نظرات سازنده و بیان راهکارهای علمی پروفیسور Cong Ling از دپارتمان برق و الکترونیک دانشگاه Imperial College London کمال سپاسگزاری را دارند. هم‌چنین از جناب آقای دکتر محمدعلی طینتی استاد دانشگاه تبریز به دلیل بیان پیشنهادها و ابراز نظرات اصلاحی بی‌نهایت سپاسگزاریم.

## مراجع

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

## زیرنویس‌ها

- 1 Low Density Parity Check Codes
- 2 Weighted Bit Flipping
- 3 Gallager
- 4 Encoding
- 5 Decoding
- 6 Hard Decision
- 7 Soft Decision
- 8 Linear Block Codes
- 9 Parity Check Matrix
- 10 Sparse
- 11 Tanner Graph
- 12 Variable Node
- 13 Check Node
- 14 Edge
- 15 Regular and Irregular
- 16 Error Control
- 17 Additive White Gaussian Noise
- 18 Syndrome Vector
- 19 Reliability Measure
- 20 Hamming Weight
- 21 Unsatisfied Check Node
- 22 Most Unreliable Variable Node
- 23 Word Error