

ارائه روشی برای تعیین قوانین بهینه در کدگذار LDPC مبتنی بر آتوماتای سلولی

مریم گرگین*^۱، دکتر محمدخیراندیش^۲

۱-دانشگاه آزاد اسلامی، واحد دزفول، گروه مهندسی کامپیوتر، دزفول، ایران،
mgorgin20@gmail.com

۲-دانشگاه آزاد اسلامی، واحد دزفول، گروه مهندسی کامپیوتر، دزفول، ایران،
mohkh_2004@yahoo.com

چکیده

لزوم ارسال و دریافت صحیح اطلاعات روی کانال‌های ارتباطی نویزی، توسعه کدهای تصحیح خطا را در پی داشته است. در بین این کدها، کدگذاری مبتنی بر ماتریس توازن کم‌چگال (LDPC)، دسته‌ای پرطرفدار در استانداردهای مدرن و آتی مخابرات بی‌سیم می‌باشد که ظرفیتی نزدیک به حد شانون را فراهم می‌نماید. ویژگیهای ساختاری ماتریس توازن در این کدگذاری باعث بهبود عملکرد این کد، از نظر توانایی تصحیح خطا شده است. با وجود پیچیدگی خطی فرآیند کدبرداری به‌عنوان یک نقطه قوت، پیچیدگی فرآیند کدگذاری، انگیزه لازم برای بهبود سرعت کدگذاری را ایجاد می‌نماید. با توجه به ماهیت کدگذاری LDPC، استفاده از ساختار ساده و در عین حال موازی آتوماتای سلولی، به بهبود سرعت این الگوریتم کمک می‌نماید. مسئله اصلی در به کارگیری آتوماتای سلولی، تعیین مجموعه قوانین حاکم بر سلول‌های آن است. در این مقاله، از الگوریتم فراابتکاری رقابت استعماری، جهت تعیین مجموعه قوانین بهینه حاکم بر سلولهای کدگذار LDPC، مبتنی بر آتوماتای سلولی استفاده شده است. این بهینه‌سازی منجر به بهبود مناسبی به ازای طول محدود کلمه کد می‌شود و انتظار می‌رود با افزایش طول کلمه کد، افزایش یابد.

کلمات کلیدی: کدگذاری LDPC، ماتریس توازن، نرخ BER، آتوماتای سلولی، الگوریتم رقابت استعماری

۱-مقدمه

امروزه با پیشرفت علم و تکنولوژی و لزوم دستیابی به دقت بالای سیستم‌های تصمیم‌گیری، صحت داده‌های دریافتی در شبکه‌ها، بیش از پیش اهمیت یافته است. تأثیرپذیری سیگنال‌ها از عوامل مخرب محیطی، موجب وقوع خطا در داده‌های انتقالی می‌شود و به کارگیری روش‌های کنترل خطا را به منظور تشخیص و تصحیح خطاهای ممکن، الزامی می‌نماید. الگوریتم‌هایی که علاوه بر تشخیص وقوع خطا، بدون نیاز به ارسال مجدد داده‌ها، توانایی اصلاح تعدادی از خطاهای رخ داده را داشته باشند، به عنوان الگوریتم‌های برتر کنترل خطا مطرح می‌شوند و امروزه، تنوع و اهمیت زیادی یافته‌اند. بررسی عوامل تأثیرگذار بر عملکرد سیستم‌های کنترل خطا، در کنار شبیه‌سازی و مقایسه عملکرد چهار الگوریتم مطرح و توانمند BCH، Reed Solomon (RS)، LDPC و Convolution نشان داده است که الگوریتم LDPC کارآمدترین الگوریتم در این میان بوده است؛ زیرا در مقایسه با سایر الگوریتم‌های کدگشایی، عملکردی نزدیکتر به حد شانون دارد. با وجود اینکه کدهای توربو به حد شانون نزدیک هستند، اما همچنان پیچیدگی کدگشایی بیشتری نسبت به کدهای LDPC دارند (Almamory, 2012). اگرچه پیچیدگی خطی فرآیند کدبرداری الگوریتم LDPC، به عنوان یک نقطه قوت برای این الگوریتم قابل توجه است، اما پیچیدگی کدگذاری آن بالا و از مرتبه $O(n^2)$ می‌باشد (Qi و همکاران، 2007). این موضوع، انگیزه لازم برای پرداختن به بهبود سرعت کدگذاری را ایجاد می‌نماید.

با توجه به ماهیت روش LDPC، استفاده از روشهای موازی سازی می تواند به بهبود سرعت این الگوریتم کمک نماید. یکی از روش هایی که در راستای بهبود عملکرد کدگذاری مورد توجه قرار گرفته، استفاده از آتوماتای سلولی است که به واسطه سادگی ساختار و عملکرد موازی اجزای خود، در پیاده سازی ساختارهای ساده با رفتار پیچیده، مورد توجه واقع شده است. در برخی از کاربردها، آتوماتای سلولی غیریکنواخت عملکرد بهتری از خود نشان می دهد؛ اما مسئله اصلی در به کارگیری آن، تعیین مجموعه ای از قوانین بهینه حاکم بر سلول های آن است. (Queen, 2016) دو آتوماتای سلولی، به ترتیب با قانون 90 و 240 را برای طراحی کدگذار LDPC به یکدیگر الحاق نمود. با این حال، ماهیت مسئله به عنوان یک مسئله بهینه سازی، انگیزه استفاده از روش های مبتنی بر الگوریتم های بهینه سازی را ایجاد نموده است. در این مقاله، برای حل مسئله مذکور، از میان انواع الگوریتم های بهینه سازی، الگوریتم های فراابتکاری مد نظر قرار گرفته و از میان آنها، الگوریتم رقابت استعماری، به دلیل نقاط قوت خود مورد توجه قرار گرفته است.

سایر بخش های مقاله به صورت ذیل سازمان یافته اند: بخش دوم به معرفی مبانی کدگذاری LDPC، مفهوم آتوماتای سلولی و الگوریتم رقابت استعماری می پردازد. در بخش سوم، مجموعه ای از کارهای انجام شده در حوزه کدگذاری مبتنی بر LDPC مرور می شوند. بخش چهارم به معرفی روش پیشنهادی اختصاص یافته است و نتایج حاصل از ارزیابی روش پیشنهادی، در بخش پنجم ارائه خواهد شد. در نهایت، بخش ششم، با عنوان نتیجه گیری، پایان بخش مقاله خواهد بود.

۲- مبانی تحقیق

۲-۱ کدهای LDPC

اگرچه کدهای LDPC، نخستین بار توسط Gallager در سال 1962 کشف شدند (Gallager, 1962)، در آن زمان، با توجه به قابلیت های محدود پردازنده های الکترونیکی، فراموش شدند. در سال 1996، مجدداً MacKay و Neal به کد Gallager توجه نمودند و امروزه به نام کدهای LDPC، در حوزه کنترل خطا مورد توجه ویژه قرار دارند. از جمله مزایای این روش، ساختار توصیفی ساده، قابلیت بالای موازی سازی در پیاده سازی سخت افزاری و تأمین ظرفیتی نزدیک به حد شانون قابل اشاره هستند. حداقل فاصله همینگ (d_{min}) برای این روش کدگذاری، متناسب با افزایش طول بلوک اطلاعاتی افزایش می یابد و این به منزله افزایش اختلاف بین کلمه کدها و در نتیجه ارتقاء قابلیت تصحیح خطا است. به علاوه، کدگشایی مؤثر توسط الگوریتم های تکراری موازی با تأخیر کم نیز می تواند به عنوان مزیتی دیگر مورد اشاره قرار گیرد (Farheen و همکاران، 2014).

کد LDPC یک کد بلوکی خطی است که مشخصه بارز آن، خلوت بودن ماتریس بررسی توازن (PCM) است؛ یعنی تعداد کمی از 1ها لابلای تعداد زیادی از صفرها قرار دارند. این ماتریس، H ، توسط الگوریتم کدگشایی، برای تعیین بردار خطا و در نتیجه، تصحیح خطا استفاده می شود. به ازای نرخ کدگذاری $r = K / N$ که در آن، K و N به ترتیب، طول داده های خام و کلمه کد را نشان می دهند، ماتریس H دارای $N-K$ سطر و N ستون است.

با در نظر گرفتن قابلیت تصحیح خطا، به عنوان یک معیار ارزیابی عملکرد کدهای کدگذاری کانال، ساختار ماتریس بررسی توازن و روش مورد استفاده برای ساخت آن، نقش مهمی ایفا می کنند. یکی از روش های مورد استفاده برای ایجاد ماتریس بررسی توازن، شامل استفاده از ساختارهای شبه تصادفی است. همانطور که در رابطه (2) نشان داده شده است، ماتریس توازن شبه تصادفی، متشکل از دو بخش است؛ بخش اول، زیر ماتریس قطری است که پهنای قطر آن، 2 است و بخش دوم، زیرماتریسی با ساختار تصادفی است.

$$H_{m \times n} = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{array} \right]_{A_{m \times n}} \quad (1)$$

یکی دیگر از مشکلات مربوط به LDPC، پیچیدگی فرآیند کدگذاری است. همانند کاربردهای دیگر، استفاده از روش‌های پردازش موازی برای غلبه بر این مشکل پیشنهاد شد. به عنوان یک ساختار موازی ساده، اتوماتای سلولی (CA)، حضوری قابل ملاحظه داشته است. دلیل اهمیت توجه به CA، این واقعیت است که عملکرد آن می‌تواند در قالب یک ماتریس مشخصه بیان شود که با ضرب در یک رشته ورودی، رشته خروجی را تولید می‌نماید.

۲- اتوماتای سلولی

اتوماتای سلولی (CA) شبکه‌ای منظم از سلول‌ها است که هر یک دارای روابط محلی با سلولهای همسایه خود بوده و تحت تأثیر آنها تغییر وضعیت می‌دهند. هر سلول، در زمان راه‌اندازی، در یک وضعیت برگرفته از مجموعه‌ای متناهی از وضعیتها قرار گرفته و در هر مرحله زمانی، طبق قاعده‌ای تعریف شده روی همسایگی سلول، وضعیت خود را به‌هنگام می‌سازد. این شبکه می‌تواند ساختاری یک بعدی یا چند بعدی داشته باشد. با توجه به حاکمیت قوانین یکسان یا متفاوت بر سلولهای مختلف، می‌توان CA ها را دارای دو نوع یکنواخت و غیریکنواخت دانست. در یک اتوماتای سلولی یک بعدی با سلولهایی با وضعیت دودویی و شعاع همسایگی r ، 2^r الگوی دودویی مختلف برای همسایگی وجود دارد و به تبع آن، 2^{2^r} قانون مختلف می‌توان بر آنها حاکم نمود.

انتخاب قانون مناسب برای سلولهای یک اتوماتای یکنواخت، شاید مسئله‌ای چالشی محسوب نشود؛ اما در یک CA غیریکنواخت، انتخاب جایگشتی مناسب از قوانین حاکم بر سلولها می‌تواند چالش برانگیز باشد و شاید در مواردی بتوان آن را یک مسئله سخت دانست و پای الگوریتم‌های بهینه‌سازی را به عرصه باز نمود.

۲-۳ الگوریتم بهینه‌سازی

بسته به تنوع کاربردها، تا کنون، الگوریتم‌های بهینه‌سازی مختلفی ارائه شده که هر یک نقطه ضعفها و نقاط قوت خاص خود را داشته‌اند. دسته الگوریتم‌های تکاملی و از میان آنها، الگوریتم‌های فراابتکاری که خود شامل کلونی زنبور عسل مصنوعی (ABC)، شبیه‌سازی تبرید (SAZ)، رقابت استعماری (ICA)، بهینه‌سازی تجمع ذرات (PSO) و ... است، طی سالهای اخیر، در حل مسائل مختلفی مورد توجه قرار گرفته‌اند. از میان این الگوریتم‌ها و به واسطه نقاط قوت ممتاز، الگوریتم رقابت استعماری برای حل مسئله بهینه‌سازی این مقاله مورد توجه واقع شده است. این الگوریتم، در سال ۲۰۰۷، توسط آتش‌پز گرگری و با الهام از پدیده‌ها و روابط سیاسی، نظامی، اقتصادی و اجتماعی بین کشورها، ارائه شده و مبتنی بر جمعیت تصادفی است (Atashpaz، ۲۰۰۷). در دنیای واقعی، برخی کشورها (استعمارگرها) به‌واسطه قدرت بیشتر، به دنبال تسلط بر کشورهای دیگر (مستعمره‌ها) و بهره‌گیری از منابع آنها هستند و در این راه، با هم رقابت و مبارزه می‌نمایند. در عین حال، برخی مستعمره‌ها، از طریق ایجاد تحولاتی در خود، سعی در نزدیک شدن به استعمارگرها داشته، حتی گاهی به رقابت با آنها می‌پردازند و چه بسا، جای آنها را نیز بگیرند. به تدریج، غلبه برخی استعمارگرها بر سایرین و جذب مستعمره‌های آنها، موجب شکست و انحلال استعمارگران مغلوب می‌شود و به این ترتیب، دنیا به سمت داشتن یک یا چند استعمارگر محدود می‌رود که قدرت‌های برتر خواهند بود.

۳- کارهای مرتبط

تحقیقات انجام‌شده در حوزه کدگذاری LDPC باعث پدید آمدن ایده‌های جدید برای کدگذاری LDPC با پیچیدگی کمتر شده‌است؛ زیرا ماتریس بررسی توازن آن خلوت است. برای مثال، با فرض اینکه بیت‌های نامشخصی توسط کانال پاک شده‌اند، کدگذار پیام مبتنی بر گراف (Tanner و همکاران، ۲۰۰۴) از خود کدگشا برای کدگذاری استفاده کرده و بنابراین، فرآیند کدگذاری دقیقاً مانند کدگشایی بود. اگر چه مدل مذکور می‌تواند بر کدهای منظم، نامنظم، تصادفی یا ساختاری اعمال شود، اما همیشه کارساز نیست (Di و همکاران، ۲۰۰۲). (Richardson و همکاران، ۲۰۰۱) روش کدگذاری دیگری طراحی کردند که در آن، ماتریس بررسی توازن (PCM)، از طریق عملیات سطر-ستونی به یک شکل تقریباً پایین مثلثی (ALT) تبدیل می‌شد (به جز عمل جمع سطرها). روش مذکور، خلوتی ماتریس را حفظ می‌کرد و پیچیدگی کدگذاری را به $O(n + g^2)$ رساند (g پارامتر شکاف نامیده می‌شود و در واقع، تعداد سطرهایی از PCM را نشان می‌دهد که تنها از طریق جایگشت‌های

سطری و ستونی به فرم پایین مثلثی در نمی آید (Q_i و همکاران، ۲۰۰۷). چالش های موجود در راه کاهش پیچیدگی فرآیند کدگذاری LDPC، برخی از محققان را بر آن داشت تا از آتوماتای سلولی برای حل مشکل استفاده کنند. در ($Queen$ و همکاران، ۲۰۱۶)، ماتریس PCM تولید شده توسط یک مولد عدد تصادفی مبتنی بر CA غیر یکنواخت، با PCM ایجاد شده توسط CA یکنواخت، تحت قوانین ۳۰، ۹۰ و ۱۵۰ مقایسه شد و نشان داد که PCM تصادفی بهتر از PCM مبتنی بر CA یکنواخت است. سپس آنها برای ایجاد کدگذار LDPC بر اساس CA غیریکنواخت، از ترکیب دو آتوماتای سلولی با قوانین ۹۰ و ۲۴۰ استفاده کردند. (Anbuselvi و همکاران، ۲۰۱۷) برای تصاویر کدگذاری شده که در کانال های AWGN منتقل می شوند، از کدهای LDPC مبتنی بر CA غیریکنواخت استفاده کردند.

۴- روش پیشنهادی

کدگذار مورد نظر در این مقاله، متشکل از دو بخش تصور می شود: یک CA غیریکنواخت با قوانین متفاوت حاکم بر سلول ها و یک CA یکنواخت با سلول هایی تحت قانون ۲۴۰ که رفتار زیر ماتریس همانی را پیاده سازی می کند. در اینجا، مسئله شامل تعیین قوانین برای بخش غیریکنواخت بوده و به عنوان یک مسئله بهینه سازی مورد توجه قرار گرفته است. برای حل مسئله، الگوریتم رقابت استعماری، به واسطه نقاط قوت خود، انتخاب شده و بر این اساس، مراحل کلی روش پیشنهادی به صورت نشان داده شده در شکل (۶-الف) قابل تصور هستند. با توجه به صورت مسئله، انتخاب از بین دو قانون برای هر سلول، به عنوان مبنا قرار گرفته و یافتن ترکیب بهینه این قوانین در یک آتوماتای سلولی M سلولی مد نظر است؛ M طول بلوک اطلاعاتی خام و $N=2M$ طول کلمه کد خواهد بود. در این راستا، یک CA غیریکنواخت با شعاع همسایگی $r=1$ (همسایگی ۳ سلولی) مورد توجه قرار گرفته و دو قانون از بین $2^3 = 256$ قانون ممکن انتخاب می شوند. بر اساس تعریف الگوریتم رقابت استعماری در ($Atashpaz$ ، ۲۰۰۷)، پارامترهای لازم برای حل مسئله عبارتند از تعداد کشورهای جمعیت اولیه ($N_{Country}$)، تعداد امپراطوریهای اولیه (N_{Emp})، نرخ جذب (β)، نرخ انقلاب (μ) و ضریب تأثیر مستعمرها (γ).

اولین مرحله اساسی، تشکیل جمعیت اولیه کشورهاست که هر یک به واسطه مقادیر ویژگیهای خود از دیگری متمایز می شود. بردار ویژگیهای هر کشور، یک جایگشت بیت های صفر و یک در نظر گرفته شده که مقدار صفر برای عنصر i ام بردار ویژگیها، معادل استفاده از قانون اول و مقدار ۱، معادل استفاده از قانون دوم در سلول i ام CA است. به این ترتیب، با در نظر داشتن یک آتوماتای سلولی M سلولی، $2M$ جایگشت مختلف می تواند وجود داشته باشد. از میان این تعداد، $N_{Country}$ جایگشت به صورت تصادفی انتخاب شده و به عنوان جمعیت اولیه کشورها در نظر گرفته می شوند.

دومین مرحله اساسی روش پیشنهادی، شامل ارزش گذاری اولیه کشورهاست. محاسبه ارزش برای هر کشور، نیازمند تشکیل ماتریس کدگذاری بر اساس حاکمیت قوانین متناظر با آن کشور بر یک آتوماتای سلولی است. در این مقاله، ترکیب دو معیار

"میزان خلوتی" ($Sparsity$) و "نرخ خطای بی تی" (BER)، به عنوان معیار ارزش گذاری هر کشور، $Val(Country_i)$

در نظر گرفته شده و طبق رابطه (۳) محاسبه می شود:

$$Val(Country_i) = Sparsity + (1-BER) \quad i=1,2,\dots,N_{Country} \quad (2)$$

$Sparsity$ به صورت نسبت تعداد عناصر صفر در ماتریس تشکیل شده برای هر کشور، به تعداد کل عناصر ماتریس محاسبه می شود و مقادیر بیشتر برای آن به معنای غلبه تعداد صفرها بر ۱ها و لذا خلوت تر بودن ماتریس است. از طرفی، محاسبه نرخ BER برای هر کشور، بر اساس رابطه (۴)، نیازمند کدگذاری بر اساس ماتریس کدگذاری تشکیل شده و سپس کدگشایی بر اساس ماتریس توازن متناظر با آن است:

$$BER = \frac{\text{The number of bits decoded az erroneous}}{\text{Total length of data block}} \times 100\% \quad (3)$$

به منظور انجام این محاسبه، ماتریس کدگذاری، طبق رابطه (۵)، شامل دو بخش سمت راست GR و سمت چپ GL در نظر گرفته می شود:

$$G = [GL : GR] \quad (4)$$

که در آن، GL یک ماتریس هممانی به ابعاد $M \times M$ است. GR ، ماتریسی $M \times M$ است که برای تشکیل آن، ابتدا برداری M عنصری به نام GR ، به صورت رابطه (۶) شکل می گیرد:

$$GR(1,1 \text{ to } M) = \left[\underbrace{1 \ 0 \ 0 \ \dots \ 0}_M \right] \quad (5)$$

سپس سطرهای جدید i ام ($i=2,3,\dots,M$) بر اساس قاعده زیر تشکیل شده و به GR الحاق می شوند:

- اگر عنصر j ام ($j=1,\dots,M$) بردار قانون صفر باشد، قانون اول برای تعیین درایه j ام در سطر i ام، بر اساس مقادیر سطر $i-1$ ام استفاده می شود. در غیر این صورت، قانون دوم برای این منظور استفاده می شود.

با ترکیب دو ماتریس، GL و GR و تشکیل ماتریس کدگذاری، ماتریس بررسی توازن، H ، با استفاده از رابطه (۷) ساخته می شود:

$$H = [GR^T \ | \ I] \quad (6)$$

پس از محاسبه Sparsity برای ماتریس H ، برای محاسبه نرخ BER، یک مجموعه از فریمهای تصادفی به طول M ساخته شده و با کدگذاری توسط ماتریس G ، به کلمه کدهای $N = 2M$ بیتی تبدیل می شوند. سپس، به منظور ارزیابی عملکرد H از نظر قابلیت کنترل خطا، نویز سفید گاوسی به کلمه کدهای تشکیل شده اعمال می شود و مجموعه ای از فریمهای خطا دار تشکیل می شود. پس از کدگشایی این فریمها، با استفاده از ماتریس H ، تفاوت نظیر به نظیر بین رشته های کدگشایی شده و فریمهای تصادفی اولیه محاسبه شده و نرخ BER با استفاده از رابطه (۴) محاسبه می شود. با قرار دادن مقادیر BER و Sparsity در رابطه (۳)، ارزش هر کشور ارزیابی شده و شایستگی آن مشخص می شود.

در مرحله اساسی بعدی، تشکیل امپراطوریها، کشورها بر اساس ارزشهای خود، به ترتیب نزولی مرتب شده و تعداد مشخصی از برترین کشورها، یعنی N_{Emp} کشور، به عنوان استعمارگر جدا شده و سایر کشورها ($N_{Country} - N_{Emp}$ کشور)، به عنوان مستعمره، متناسب با ارزش استعمارگرها، به صورت تصادفی بین آنها توزیع می شوند؛ هر استعمارگری که ارزش بیشتری داشته باشد، مستعمره های بیشتری جذب می کند. هر استعمارگر، به همراه مستعمره های نسبت داده شده به خود، یک امپراطوری تشکیل می دهد که ارزش آن، تابعی از ارزش استعمارگر و مستعمره های تحت سلطه آن است و بر اساس رابطه (۸) محاسبه می شود:

$$Val(Emp_i) = Val(Imp_i) + \zeta \text{ mean}(Val(Col_{i,j})), \quad i=1,2,\dots,N_{Emp}, \quad j=1,2,\dots,Cols_num_i \quad (7)$$

که در آن، Emp_i ، مبین امپراطوری i ام، Imp_i مبین استعمارگر i ام، $Col_{i,j}$ نشان دهنده مستعمره j ام از امپراطوری i ام و $Cols_num_i$ تعداد مستعمره های امپراطوری i ام است.

با مشخص شدن ارزش امپراطوریها، روالی تکراری آغاز می شود که تا محقق شدن یک شرط خاتمه، ادامه خواهد یافت. در هر تکرار، بر اساس الگوریتم رقابت استعماری، طی فرآیندهای جذب و انقلاب، تغییراتی در مقادیر بردارهای ویژگی کشورها ایجاد می شود که آنها را به بردارهای ویژگی دیگری تبدیل می نماید. سپس، به واسطه تغییرات ایجاد شده در کشورها، ارزش جدید کشورها و امپراطوریها، مجدداً طبق رابطه های (۳) و (۸) محاسبه می شود. ضعیف ترین امپراطور تعیین می شود و یکی از مستعمره های آن که به صورت تصادفی برگزیده می شود از استعمارگر خود جدا شده و به امپراطوری دیگری که به طور تصادفی انتخاب می شود، ملحق می گردد. چنانچه این جابجایی منجر به تنها ماندن یک استعمارگر شود، امپراطوری آن منحل شده و استعمارگر تنها، به عنوان یک مستعمره و به همان شیوه ای که شرح داده شد، به امپراطوری دیگری می پیوندد. به واسطه این تغییرات، ارزش تمام امپراطوریها مجدداً به هنگام می شود.

در پایان هر دوره تکامل، شرط خاتمه بررسی می شود. در اینجا، شرط خاتمه اصلی، تجمع تمام امپراطوریها و باقی ماندن تنها یک امپراطور است. اما در صورت ثابت ماندن ارزش بهترین امپراطوری، طی چندین دوره، بنا بر این گذاشته می شود که بهبود جدیدی حاصل نخواهد شد و می توان دوره های تکامل را خاتمه داد و بردار قانون متناظر با بهترین امپراطور را به عنوان

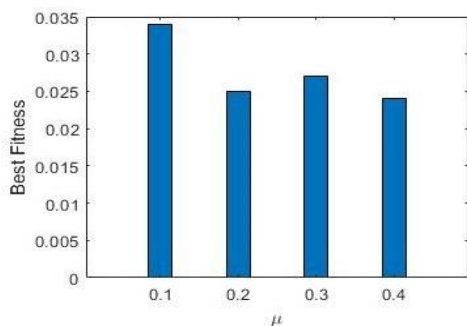
بردار ویژگی بهینه انتخاب نمود. با جایگزینی بیت‌های صفر با اندیس قانون اول و بیت‌های ۱ با اندیس قانون دوم، بردار قانون بهینه حاصل و به عنوان پاسخ مسئله اعلام می‌شود.

۵- ارزیابی نتایج

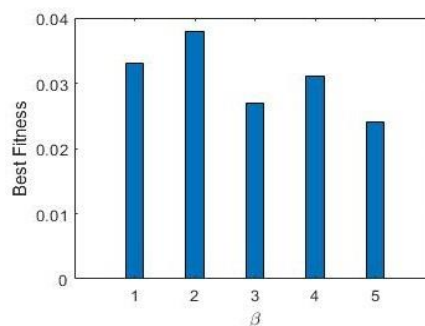
به منظور ارزیابی نرخ BER، هر بار ۱۰ فریم اطلاعاتی به صورت تصادفی تولید می‌شوند. برای کنترل نویز سفید گاوسی اعمال شونده به فریم‌ها، از نسبت سیگنال به نویز بر حسب EbNo، dB، استفاده می‌شود. فریم‌های خطا دار توسط دو الگوریتم decodeLogDomainSimple و decodeBitFlip، کدگشایی شده، مقدار BER برای هر کدام محاسبه شده و میانگین BER حاصل از هر الگوریتم، به ازای تمام فریم‌ها محاسبه می‌شود. نهایتاً حداکثر مقدار BER از بین دو BER بدست آمده، به عنوان خروجی تابع محاسبه BER برگردانده می‌شود. برای ارزیابی مدل پیشنهادی، آزمایش‌های مختلفی جهت تعیین مقادیر پارامترها و نیز آزمایش عملکرد، طراحی و اجرا شده‌اند.

۵-۱- آزمایش اول - تعیین پارامترهای رقابت استعماری

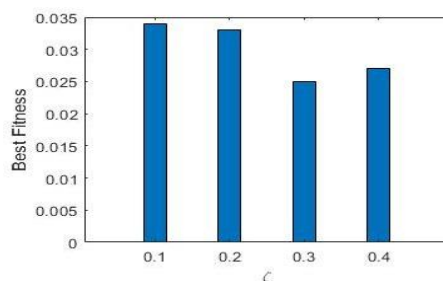
در اولین آزمایش، تعیین مقدار بهینه برای نرخ جذب (β)، نرخ انقلاب (μ) و ضریب تأثیر مستعمره‌ها (ζ) مورد توجه است. به ازای طول بلوک اطلاعاتی خام، $M=8$ ، مقادیر مختلفی برای سه پارامتر مذکور در نظر گرفته شده و تغییرات ارزش بهترین امپراطور، به عنوان معیاری برای سنجش نتایج مورد توجه قرار گرفته است. شکل (۳-الف) این تغییرات را به ازای طول بلوک $M=8$ ، $\zeta=0.1$ ، $\mu=0.1$ و پنج مقدار $\beta = \{1, 2, 3, 4, 5\}$ نشان می‌دهد که بر اساس آن، بهترین نتیجه به ازای $\beta=2$ بدست آمده و لذا در مراحل بعد، $\beta=2$ خواهد بود. در ادامه، به ازای $M=8$ ، $\beta=2$ و $\zeta=0.1$ ، چهار مقدار مختلف $\mu = \{0.1, 0.2, 0.3, 0.4\}$ در نظر گرفته شده و تغییرات ارزش بهترین استعمارگر، در شکل (۳-ب) نشان داده شده‌اند (بهترین ارزش به ازای $\mu=0.1$ بدست آمده است). نتایج آزمایش به ازای $M=8$ ، $\beta=2$ و $\mu=0.1$ و چهار مقدار مختلف $\zeta = \{0.1, 0.2, 0.3, 0.4\}$ در شکل (۳-ج) نشان می‌دهند که بهترین *Best score* متعلق به $\zeta=0.1$ است.



ب) تغییرات ارزش بهترین استعمارگر به ازای $M=8$ ، $\zeta=0.1$ و $\beta=2$ و μ های مختلف



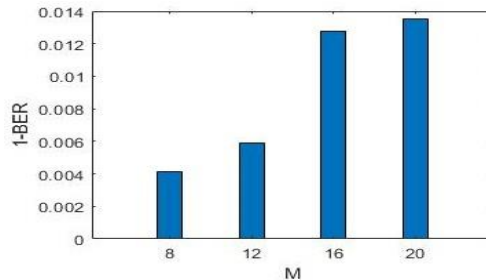
الف) تغییرات ارزش بهترین استعمارگر به ازای $M=8$ ، $\zeta=0.1$ و $\mu=0.1$ و β های مختلف



ج) تغییرات ارزش بهترین استعمارگر به ازای $M=8$ ، $\mu=0.1$ ، $\beta=2$ و ζ های مختلف
شکل (۳): نتایج حاصل از آزمایش اول

۵-۲- آزمایش دوم - ارزیابی تأثیر طول بلوک اطلاعاتی

در این آزمایش، تأثیر افزایش طول بلوک اطلاعاتی بر تغییرات نرخ (BER)، طی دوره‌های تکاملی مورد توجه قرار می‌گیرد. در این راستا، با در نظر گرفتن $\beta=2$ ، $\mu=0.1$ و $\zeta=0.1$ ، چهار مقدار $M=\{8, 12, 16, 20\}$ مورد توجه قرار گرفته و تغییرات BER برای بهترین استعمارگر در قالب شکل (۴) نشان داده شده‌اند.

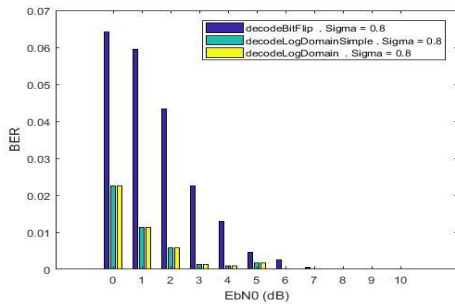


شکل (۴): تغییرات نرخ BER بهترین استعمارگر به ازای $\mu=0.1$ ، $\zeta=0.1$ و $\beta=2$ و مقادیر مختلف M

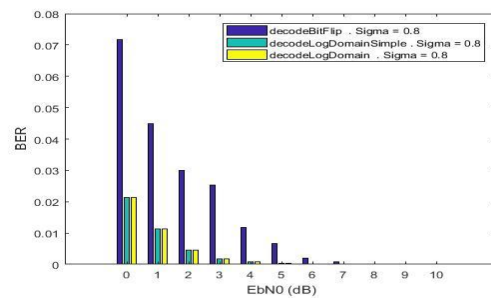
همانطور که ملاحظه می‌شود، با افزایش طول بلوک اطلاعاتی، میزان 1-BER افزایش یافته و این به معنی کاهش خطای کدگشایی می‌باشد

۳-۵ آزمایش سوم - ارزیابی اثر نویز

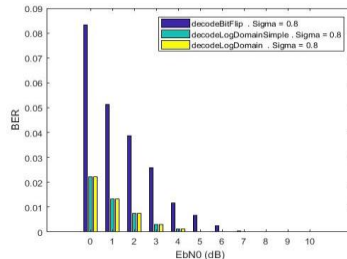
در این آزمایش، ابتدا با در نظر گرفتن $M=20$ ، $\beta=2$ ، $\mu=0.1$ و $\zeta=0.1$ ، بهترین استعمارگر نهایی یافته شده و بردار ویژگی آن، به عنوان بهترین بردار قانون مورد توجه قرار می‌گیرد. سپس، با در نظر گرفتن نسبت‌های سیگنال به نویز مختلف (بر حسب dB) یعنی $E_b/N_0=[1,2,\dots,10]$ ، کدگذاری، افزودن نویز، کدگشایی و ارزیابی مقدار BER به ازای سه الگوریتم کدگشایی decodeBitFlip، decodeLogDomain، decodeLogDomainSimple صورت گرفته و این آزمایش سه بار به ازای جفت قاعده‌های مختلف تکرار شده است. نتایج حاصل از به کارگیری جایگشتهای جفت قاعده $\langle 30, 150 \rangle$ در شکل (۵) نشان می‌دهند که دو الگوریتم کدگشایی DecodeLogDomainSimple و DecodeLogDomain دارای مقادیر BER بهتری هستند و با افزایش E_b/N_0 سریعتر کاهش می‌یابند. همچنین، شکل (۶) و شکل (۷) نتایج را به ازای $\langle 90, 150 \rangle$ و $\langle 30, 90 \rangle$ نشان می‌دهند. به طور مشابه، نتایج اخیر نشان دهنده عملکرد بهتر برای DecodeLogDomain و DecodeBitFlip، در مقایسه با DecodeLogDomainSimple هستند.



شکل (۶): تغییرات نرخ BER، حاصل از سه الگوریتم کدگشایی، با استفاده از جفت قوانین $\langle 90, 150 \rangle$



شکل (۵): تغییرات نرخ BER، حاصل از سه الگوریتم کدگشایی، با استفاده از جفت قوانین $\langle 30, 150 \rangle$



شکل (۷): تغییرات نرخ BER حاصل از سه الگوریتم کدگشایی، با استفاده از جفت قوانین $\langle 30, 90 \rangle$

۴-۵ آزمایش چهارم - مقایسه روش پیشنهادی

در این آزمایش، مقایسه میزان BER حاصل از ماتریس مبتنی بر روش پیشنهادی و ماتریس مبتنی بر (Queen و همکاران، ۲۰۱۶) مورد بررسی قرار گرفته است. جدول (۱) نتایج این مقایسه را به ازای $EbNo = \{0, 5, 10, 15, 20\}$ و $M = \{8, 12, 16, 20\}$ نشان می‌دهد.

جدول (۱): مقایسه روش پیشنهادی با (Queen و همکاران، ۲۰۱۶) برای تولید کدهای LDPC مبتنی بر اتوماتای سلولی

$EbNo (dB)$					M	الگوریتم	BER
۲۰	۱۵	۱۰	۵	۰			
۰,۳۰۵۷	۰,۲۶۹۷	۰,۲۹۲۸	۰,۲۹۱۵	۰,۲۶۳۸	۸	Queen و همکاران، (۲۰۱۶)	
۰,۲۹۸۸	۰,۳۰۷۰	۰,۳۰۳۵	۰,۳۰۲۸	۰,۲۸۴۵	۱۲		
۰,۲۸۳۳	۰,۲۹۷۸	۰,۲۹۳۸	۰,۲۸۷۸	۰,۲۸۵۳	۱۶		
۰,۲۹۳۰	۰,۲۷۷۳	۰,۲۹۷۵	۰,۲۸۱۵	۰,۲۸۲۰	۲۰		
۰,۱۷۵۵	۰,۱۷۱۰	۰,۱۶۹۵	۰,۱۶۶۰	۰,۱۶۱۰	۸	الگوریتم پیشنهادی	
۰,۱۷۶۰	۰,۱۷۲۵	۰,۱۶۶۵	۰,۱۶۴۰	۰,۱۶۴۵	۱۲		
۰,۱۶۷۵	۰,۱۶۰۵	۰,۱۶۷۵	۰,۱۵۷۵	۰,۱۶۲۰	۱۶		
۰,۱۶۷۰	۰,۱۶۰۰	۰,۱۶۵۰	۰,۱۵۶۵	۰,۱۶۱۰	۲۰		

۶. نتیجه گیری

تلاش برای بهبود فرآیند کدگذاری LDPC موجب شده تا از قابلیت پردازش موازی اتوماتای سلولی استفاده شود. اتوماتای سلولی غیریکنواخت، به دلیل عملکرد بهتر در کاربردهای مختلف، توجه بیشتری را به خود جلب کرده است اما نیازمند تعیین جایگشت بهینه‌ای از قوانین حاکم بر سلولهاست. در این مقاله، تعیین بهترین قوانین برای یک کدگذار LDPC مبتنی بر اتوماتای سلولی غیریکنواخت، به عنوان مسئله اصلی مطرح شده و از الگوریتم رقابت استعماری برای حل آن استفاده شده است. مقایسه این راه حل با راه‌حل‌های پیشین، نشان‌دهنده بهبود به میزان ۵۵,۸٪ برای طول بلوک $M = 20$ و بهبود به میزان ۵۶,۹٪ برای تمام طول بلوک‌های ذکر شده برای محدوده بین ۸ تا ۲۰ است.

مراجع

- [1]. Almaamory, A. N., & Mohammed, H. A. (2012). Performance evaluation and comparison between LDPC and Turbo coded MC-CDMA. Journal of Engineering, 18(8).
- [2]. Anbuselvi, M., & Saravanan, P. (2017). Image coding using Cellular Automata based LDPC codes. International Journal of Computer Science and Network Security, 17(11), 146-150.
- [3]. Di, C., Proietti, D., Telatar, I. E., Richardson, T. J., & Urbanke, R. L. (2002). Finite-length analysis of low-density parity-check codes on the binary erasure channel. IEEE Transactions on Information theory, 48(6), 1570-1579.
- [4]. Farheen, N. S., Hazreena, A., Mathew, J. A., & Ahamed, G. (2014). Performance Evaluation of LDPC Codes Over Various Channels. Advanced Research in Electrical and Electronic Engineering, 1(1), 8-11.
- [5]. Gallager, R. (1962). Low-density parity-check codes. IRE Transactions on information theory, 8(1), 21-28.
- [6]. Lin, S., Costello, D., (2004), "Error Control Coding", Prentice Hall, 2nd edition.

- [7]. Qi, H., & Goertz, N. (2007). Low-Complexity Encoding of LDPC Codes: A New Algorithm and its [8].Performance. Institute for Digital Communications, Joint Research Institute for Signal & Image Processing, School of Engineering and Electronics, The University of Edinburgh.
- [9]. Queen, C. A., Anbuselvi, M., & Salivahanan, S. (2016). Cellular Automata-Based LDPC Decoder. In Artificial Intelligence and Evolutionary Computations in Engineering Systems (pp. 885-894). Springer, New Delhi.
- [10]. Richardson, T. J., & Urbanke, R. L. (2001). Efficient encoding of low-density parity-check codes. *IEEE Transactions on information theory*, 47(2), 638-656.
- [11]. Tanner, R. M., Sridhara, D., Sridharan, A., Fuja, T.E., & Costello, D. J. (2004). LDPC block and convolutional codes based on circulant matrices. *IEEE Transactions on Information Theory*, 50(12), 2966-2984.